# **MethylNet Documentation**

Release 0.1

Joshua Levy

## **CONTENTS:**

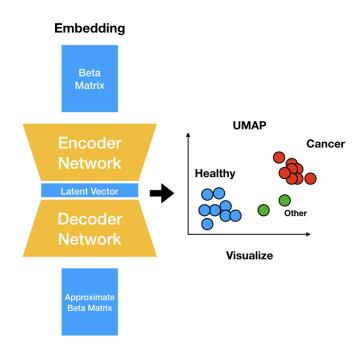
| 1  | datasets.py  | 3  |
|----|--|--|
| 2  | hyperparameter_scans.py  | 7  |
| 3  | torque_jobs.py   | 9  |
| 4  | interpretation_classes.py  | 11   |
| 5  | models.py  | 23   |
| 6  | plotter.py   | 35   |
| 7  | schedulers.py  | 37   |
| 8  | methylnet-embed8.1launch_hyperparameter_scan8.2perform_embedding   | <b>39</b><br>39<br>40  |
| 9  | methylnet-predict9.1 classification_report9.2 launch_hyperparameter_scan9.3 make_new_predictions9.4 make_prediction9.5 regression_report   | 43<br>43<br>44<br>45<br>47   |
| 10 | methylnet-interpret  10.1 bin_regression_shaps.  10.2 extract_methylation_array  10.3 grab_lola_db_cache  10.4 interpret_biology  10.5 list_classes  10.6 list_individuals  10.7 order_results_by_col  10.8 plot_all_lola_outputs  10.9 plot_lola_output  10.10 plot_top_cpgs  10.11 produce_shapley_data  10.12 reduce_top_cpgs  10.13 regenerate_top_cpgs  10.14 shapley_jaccard  10.15 split_hyper_hypo_methylation | 49<br>49<br>50<br>50<br>51<br>52<br>52<br>52<br>53<br>53<br>54<br>55<br>56<br>56 |

|     | 10.16 view_methylation_top_cpgs                                | 57        |
|-----|--|-----------|
| 11  | methylnet-visualize11.1 plot_roc_curve11.2 plot_training_curve |           |
|     | methylnet-torque 12.1 run_torque_job                           | <b>61</b> |
| 13  | Example Usage  | 63        |
| 14  | Indices and tables   | 69        |
| Рy  | thon Module Index  | 71        |
| Inc | dex  | 73        |

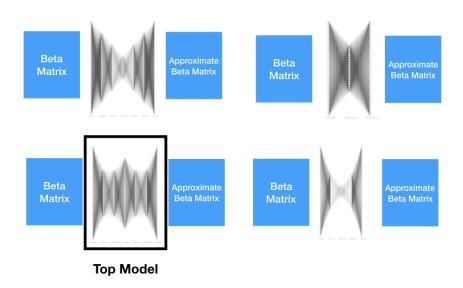
## https://github.com/Christensen-Lab-Dartmouth/MethylNet

See README.md in Github repository for install directions and for example scripts for running the pipeline (not all datasets may be available on GEO at this time).

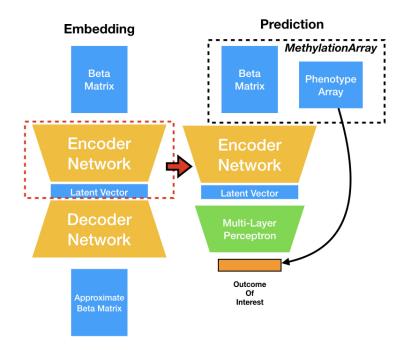
There is both an API and CLI available for use. Examples for CLI usage can be found in ./example\_scripts.

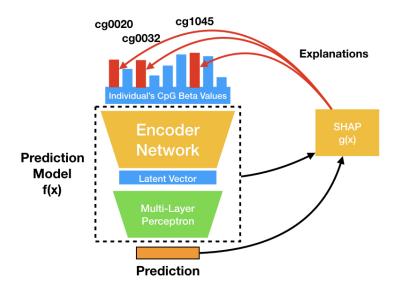


## **Hyper-Parameter Scan**



CONTENTS: 1





2 CONTENTS:

## **CHAPTER**

## **ONE**

## DATASETS.PY

Contains datatypes core to loading MethylationArrays into Torch tensors.

Pytorch Dataset that contains instances of methylation array samples to be loaded.

#### **Parameters**

methylation\_array [MethylationArray] Methylation Array input.

transform [Transformer] Transforms data into torch tensor.

outcome\_col [str] Pheno column(s) to train on.

categorical [bool] Whether predicting categorical/classification.

categorical\_encoder [encoder] Encoder used to binarize categorical column.

## Attributes

samples [np.array] Samples of MethylationArray

features [np.array] List CpGs.

encoder: Encoder used to binarize categorical column.

**new\_shape:** Shape of torch tensors.

length: Number CpGs.

 $methylation\_array$ 

outcome col

transform

## **Methods**

| to_methyl_array(self) | Convert torch dataset back into methylation array,  |
|-----------------------|---|
|                       | useful because turning into torch dataset can cause |
|                       | the original MethylationArray beta matrix to turn   |
|                       | into numpy array, when needs turn back into pandas  |
|                       | dataframe.  |

## to\_methyl\_array(self)

Convert torch dataset back into methylation array, useful because turning into torch dataset can cause

the original MethylationArray beta matrix to turn into numpy array, when needs turn back into pandas dataframe.

#### Returns

## MethylationArray

```
class methylnet.datasets.MethylationPredictionDataSet (methylation\_array, transform, outcome\_col=", categorical=False, categorical=encoder=False)
```

MethylationArray Torch Dataset that contains instances of methylation array samples to be loaded, specifically targetted for prediction.

#### **Parameters**

```
methylation_array [MethylationArray] Methylation Array input.

transform [Transformer] Transforms data into torch tensor.

outcome_col [str] Pheno column(s) to train on.

categorical [bool] Whether predicting categorical/classification.
```

categorical\_encoder [encoder] Encoder used to binarize categorical column.

#### **Attributes**

```
samples [np.array] Samples of MethylationArray features [np.array] List CpGs.
```

**encoder:** Encoder used to binarize categorical column.

**new\_shape:** Shape of torch tensors.

length: Number CpGs.
methylation\_array
outcome\_col

transform

#### **Methods**

| to_methyl_array(self) | Convert torch dataset back into methylation array,  |
|-----------------------|---|
|                       | useful because turning into torch dataset can cause |
|                       | the original MethylationArray beta matrix to turn   |
|                       | into numpy array, when needs turn back into pandas  |
|                       | dataframe.  |

class methylnet.datasets.RawBetaArrayDataSet (beta\_array, transform)

Torch Dataset just for beta matrix in numpy format.

#### **Parameters**

beta\_array [numpy.array] Beta value matrix in numpy form.

transform: Transforms data to torch tensor.

#### **Attributes**

**length:** Number CpGs.

#### beta array

#### transform

**class** methylnet.datasets.**Transformer** (*convolutional=False*, *cpg\_per\_row=30000*, *l=None*) Push methyl array sample to pytorch tensor, possibly turning into image.

#### **Parameters**

```
convolutional [bool] Whether running convolutions on torch dataset.
```

cpg\_per\_row [int] Number of CpGs per row of image if convolutional.

I [tuple] Attributes that describe image.

#### **Attributes**

shape [tuple] Shape of methylation array image for sample

convolutional

```
cpg_per_row
l
```

## **Methods**

generate(self)

Generate function for transform.

```
generate(self)
```

Generate function for transform.

## Returns

#### function

```
methylnet.datasets.cat()
```

Concatenates the given sequence of seq tensors in the given dimension. All tensors must either have the same shape (except in the concatenating dimension) or be empty.

```
torch.cat() can be seen as an inverse operation for torch.split() and torch.chunk().
```

torch.cat() can be best understood via examples.

## **Args:**

**tensors** (**sequence of Tensors**): **any python sequence of tensors of the same type.** Non-empty tensors provided must have the same shape, except in the cat dimension.

dim (int, optional): the dimension over which the tensors are concatenated out (Tensor, optional): the output tensor

## Example:

(continues on next page)

(continued from previous page)

```
methylnet.datasets.get_methylation_dataset(methylation_array, outcome_col, con-
volutional=False, cpg_per_row=1200,
predict=False, categorical=False, categori-
cal encoder=False)
```

Turn methylation array into pytorch dataset.

#### **Parameters**

```
methylation_array [MethylationArray] Input MethylationArray.
outcome_col [str] Pheno column to train on.
convolutional [bool] Whether running CNN on methylation data
cpg_per_row [int] If convolutional, number of cpgs per image row.
predict [bool] Running prediction algorithm vs VAE.
categorical [bool] Whether training on categorical vs continuous variables.
categorical_encoder: Scikit learn encoder.
```

#### Returns

## **Pytorch Dataset**

```
methylnet.datasets.stack()
```

Concatenates sequence of tensors along a new dimension.

All tensors need to be of the same size.

**Arguments:** seq (sequence of Tensors): sequence of tensors to concatenate dim (int): dimension to insert. Has to be between 0 and the number

of dimensions of concatenated tensors (inclusive)

out (Tensor, optional): the output tensor

## HYPERPARAMETER\_SCANS.PY

Run randomized grid search to find ideal model hyperparameters, with possible deployments to batch system for scalability.

```
methylnet.hyperparameter_scans.coarse_scan(hyperparameter_input_csv,
                                                                                              hy-
                                                                                            gener-
                                                        perparameter_output_log,
                                                        ate input,
                                                                       job chunk size,
                                                                                            strat-
                                                        ify_column,
                                                                      reset all,
                                                                                   torque,
                                                                                             gpu,
                                                        gpu_node,
                                                                     nohup,
                                                                               mlp=False,
                                                                                              cus-
                                                        tom_jobs=[], model_complexity_factor=0.9,
                                                                          n_{jobs}=4,
                                                        set\_beta=-1.0,
                                                                                         categori-
                                                        cal=True,
                                                                     add\_softmax = False,
                                                                                             addi-
                                                        tional_command=")
```

Perform randomized hyperparameter grid search

#### **Parameters**

**hyperparameter\_input\_csv** [type] CSV file containing hyperparameter inputs.

**hyperparameter\_output\_log** [type] CSV file containing prior runs.

generate\_input [type] Generate hyperparameter input csv.

**job\_chunk\_size** [type] Number of jobs to be launched at same time.

**stratify\_column** [type] Performing classification?

reset\_all [type] Rerun all jobs previously scanned.

torque [type] Run jobs using torque.

**gpu** [type] What GPU to use, set to -1 to be agnostic to GPU selection.

**gpu\_node** [type] What GPU to use, set to -1 to be agnostic to GPU selection, for torque submission.

nohup [type] Launch jobs using nohup.

**mlp** [type] If running prediction job (classification/regression) after VAE.

**custom\_jobs** [type] Supply custom job parameters to be run.

**model\_complexity\_factor** [type] Degree of neural network model complexity for hyperparameter search. Search for less wide networks with a lower complexity value, bounded between 0 and infinity.

**set\_beta** [type] Don't hyperparameter scan over beta (KL divergence weight), and set it to value.

**n\_jobs** [type] Number of jobs to generate.

categorical [type] Classification task?

add\_softmax [type] Add softmax layer at end of neural network.

methylnet.hyperparameter\_scans.**find\_top\_jobs**(hyperparameter\_input\_csv, hyperparameter\_output\_log, n\_top\_jobs, crossover\_p=0,

val loss column='min val loss')

Finds top performing jobs from hyper parameter scan to rerun and cross-over parameters.

#### **Parameters**

hyperparameter\_input\_csv [str] CSV file containing hyperparameter inputs.

hyperparameter\_output\_log [str] CSV file containing prior runs.

**n\_top\_jobs** [int] Number of top jobs to select

crossover\_p [float] Rate of cross over of reused hyperparameters

val\_loss\_column [str] Loss column used to select top jobs

#### Returns

**list** List of list of new parameters for jobs to run.

methylnet.hyperparameter\_scans.generate\_topology(topology\_grid, probability\_decay\_factor=0.9)

Generates list denoting neural network topology, list of hidden layer sizes.

#### **Parameters**

topology\_grid [list] List of different hidden layer sizes (number neurons) to choose from.

**probability\_decay\_factor** [float] Degree of neural network model complexity for hyperparameter search. Search for less wide networks with a lower complexity value, bounded between 0 and infinity.

## Returns

list List of hidden layer sizes.

## **THREE**

## **TORQUE JOBS.PY**

```
Wraps and runs your commands through torque.
```

Create dictionary to update BASH submission script for torque.

#### **Parameters**

```
command [type] Command to executer through torque.
```

use\_gpu [type] GPUs needed?

additions [type] Additional commands to add (eg. module loads).

queue [type] Queue to place job in.

time [type] How many hours to run job for.

**ngpu** [type] Number of GPU to use.

## Returns

**Dict** Dictionary used to update Torque Script.

Runs torque job after passing commands to setup bash file.

## **Parameters**

```
command [type] Command to executer through torque.
```

use\_gpu [type] GPUs needed?

**additions** [type] Additional commands to add (eg. module loads).

queue [type] Queue to place job in.

**time** [type] How many hours to run job for.

**ngpu** [type] Number of GPU to use.

additional\_options [type] Additional options to pass to Torque scheduler.

#### **Returns**

**job** Custom job name.

```
methylnet.torque_jobs.run_torque_job_(replace_dict, additional_options=")
Run torque job after creating submission script.
```

#### **Parameters**

**replace\_dict** [type] Dictionary used to replace information in bash script to run torque job. **additional\_options** [type] Additional options to pass scheduler.

## Returns

str Custom torque job name.

## INTERPRETATION CLASSES.PY

Contains core classes and functions to extracting explanations for the predictions at single samples, and then interrogates important CpGs for biological plausibility.

**class** methylnet.interpretation\_classes.**BioInterpreter**(*dict\_top\_cpgs*) Interrogate CpGs found to be important from SHAP through enrichment and overlap tests.

#### **Parameters**

dict\_top\_cpgs [type] Dictionary of topCpGs output from ShapleyExplorer object

## **Attributes**

top\_cpgs [type] Dictionary of top cpgs to be interrogated.

#### **Methods**

| get_nearby_cpg_shapleys(self, all_cpgs,         | Query for CpGs that are within a max_gap of the   |
|---|---|
| max_gap)  | topCpGs in the genome, helps find cpgs that could |
|   | be highly correlated and thus also important.     |
| <pre>gometh(self[, collection, allcpgs,])</pre> | Run GO, KEGG, GSEA analyses or return nearby      |
|   | genes.  |
| return_overlap_score(self[, set_cpgs,])         | Perform overlap test, overlapping top CpGs with   |
|   | IDOL CpGs, age related cpgs, etc.                 |
| run_lola(self[, all_cpgs, lola_db, cores,])     | Run LOLA enrichment test.                         |

## get\_nearby\_cpg\_shapleys (self, all\_cpgs, max\_gap)

Query for CpGs that are within a max\_gap of the topCpGs in the genome, helps find cpgs that could be highly correlated and thus also important.

## **Parameters**

**all\_cpgs** [type] List of all cpgs in methylationarray.

max\_gap [type] Max radius to search for nearby CpGs.

#### **Returns**

**Dict** Results for each class/indiv's top CpGs, in the form of nearby CpGs to each of these CpGs sets.

gometh (self, collection='GO', allcpgs=[], length\_output=20, gsea\_analyses=[], gsea\_pickle=")
Run GO, KEGG, GSEA analyses or return nearby genes.

#### **Parameters**

**collection** [type] GO, KEGG, GSEA, GENE overlaps?

**allcpgs** [type] All CpGs defines CpG universe, empty if use all CpGs, smaller group of CpGs may yield more significant results.

**length\_output** [type] How many lines should the output be, maximum number of results to print.

gsea\_analyses [type] GSEA analyses/collections to target.

**gsea\_pickle** [type] Location of gsea pickle containing gene sets, may need to run download help data to acquire.

#### **Returns**

**Dict** Dictionary containing results from each test run on all of the keys in top\_cpgs.

return\_overlap\_score (self, set\_cpgs='IDOL', platform='450k', all\_cpgs=[], output\_csv='output\_bio\_intersect.csv', extract\_library=False)
Perform overlap test, overlapping top CpGs with IDOL CpGs, age related cpgs, etc.

#### **Parameters**

set\_cpgs [type] Set of reference cpgs.

platform [type] 450K or 850K

**all\_cpgs** [type] All CpGs to build a universe.

**output\_csv** [type] Output CSV for results of overlaps between classes, how much doo they share these CpGs overlapped.

extract\_library [type] Can extract a list of the CpGs that ended up being overlapped with.

#### Returns

List Optional output of overlapped library of CpGs.

**run\_lola** (*self*, *all\_cpgs=[]*, *lola\_db=*", *cores=8*, *collections=[]*, *depletion=False*) Run LOLA enrichment test.

#### **Parameters**

all\_cpgs [type] CpG universe for LOLA.

**lola\_db** [type] Location of LOLA database, can be downloaded using methylnet-interpret. Set to extended or core, and collections correspond to these.

cores [type] Number of cores to use.

collections [type] LOLA collections to run, leave empty to run all.

**depletion** [type] Set true to look for depleted regions over enriched.

#### **Returns**

type Description of returned object.

Produces SHAPley explanation scores for individual predictions, approximating a complex model with a basic linear one, one coefficient per CpG per individual.

#### **Parameters**

**prediction\_function** [type] Model or function that makes predictions on the data, can be sklearn .predict method or pytorch forward pass, etc...

cuda [type] Run on GPUs?

## Attributes

**explainer** [type] Type of SHAPley explanation method; kernel (uses LIME model agnostic explainer), deep (uses DeepLift) or Gradient (integrated gradients and backprop for explanations)?

## prediction\_function

cuda

#### **Methods**

| build_explainer(self, train_methyl_array[,     | Builds SHAP explainer using background samples.       |
|--|---|
| ])   |   |
| classifier_assign_scores_to_shap_dat           | a (sestigns the SHAP scores to a SHAPley data object, |
| )  | populating nested dictionaries (containing class-     |
|  | individual information) with SHAPley information      |
|  | and "top CpGs".                                       |
| <pre>feature_select(self, methyl_array,)</pre> | Perform feature selection based on the best overall   |
|  | SHAP scores across all samples.                       |
| from_explainer(method, cuda)                   | Load custom SHAPley explainer                         |
| regressor_assign_scores_to_shap_data           | (sAlfsigns the SHAP scores to a SHAPley data object,  |
| )  | populating nested dictionaries (containing multiout-  |
|  | put and single output regression-individual informa-  |
|  | tion) with SHAPley information and "top CpGs".        |
| return_shapley_predictions(self,[,             | Method in development or may be deprecated.           |
| encoder])                                      |   |
| $return\_shapley\_scores(self,[,])$            | Generate explanations for individual predictions, in  |
|  | the form of SHAPley scores for supplied test data.    |

**build\_explainer** (*self*, *train\_methyl\_array*, *method='kernel'*, *batch\_size=100*) Builds SHAP explainer using background samples.

## **Parameters**

**train\_methyl\_array** [type] Train Methylation Array from which to populate background samples to make estimated explanations.

**method** [type] SHAP explanation method?

**batch\_size** [type] Break up prediction explanation creation into smaller batch sizes for lower memory consumption.

 $\begin{tabular}{ll} {\bf classifier\_assign\_scores\_to\_shap\_data} (self, test\_methyl\_array, n\_top\_features, inter-est\_col='disease', prediction\_classes=None) \\ \end{tabular}$ 

Assigns the SHAP scores to a SHAPley data object, populating nested dictionaries (containing class-individual information) with SHAPley information and "top CpGs".

#### **Parameters**

**test\_methyl\_array** [type] Testing MethylationArray.

**n\_top\_features** [type] Number of top SHAP scores to use for a subdict called "top CpGs"

interest\_col [type] Column in pheno sheet from which class names reside.

prediction\_classes [type] User supplied prediction classes.

#### feature\_select (self, methyl\_array, n\_top\_features)

Perform feature selection based on the best overall SHAP scores across all samples.

#### **Parameters**

methyl\_array [type] MethylationArray to run feature selection on.

**n\_top\_features** [type] Number of CpGs to select.

#### Returns

**MethylationArray** Subsetted by top overall CpGs, overall positive contributions to prediction. May need to update.

## classmethod from\_explainer(method, cuda)

Load custom SHAPley explainer

Assigns the SHAP scores to a SHAPley data object, populating nested dictionaries (containing multioutput and single output regression-individual information) with SHAPley information and "top CpGs".

#### **Parameters**

test\_methyl\_array [type] Testing MethylationArray.

**n\_top\_features** [type] Number of top SHAP scores to use for a subdict called "top CpGs"

**cell\_names** [type] If multi-output regression, create separate regression classes to access for all individuals.

return\_shapley\_predictions (self, test\_methyl\_array, sample\_name, interest\_col, encoder=None)

Method in development or may be deprecated.

```
return_shapley_scores (self, test_methyl_array, n_samples, n_outputs, shap_sample_batch_size=None, top_outputs=None)
```

Generate explanations for individual predictions, in the form of SHAPley scores for supplied test data. One SHAP score per individual prediction per CpG, and more if multiclass/output. For multiclass/multivariate outcomes, scores are generated for each outcome class.

#### **Parameters**

test\_methyl\_array [type] Testing MethylationArray.

**n\_samples** [type] Number of SHAP score samples to produce. More SHAP samples provides convergence to correct score.

**n\_outputs** [type] Number of outcome classes.

**shap\_sample\_batch\_size** [type] If not None, break up SHAP score sampling into batches of this size to be averaged.

**top\_outputs** [type] For deep explainer, limit number of output classes due to memory consumption.

#### Returns

type Description of returned object.

```
class methylnet.interpretation classes.PlotCircos
```

Plot Circos Diagram using ggbio (regular circos software may be better)

## **Attributes**

**generate\_base\_plot** [type] Function to generate base plot

add\_plot\_data [type] Function to add more plot data
generate\_final\_plot [type] Function to plot data using Circos

#### **Methods**

| <pre>plot_cpgs(self, top_cpgs[, output_dir])</pre> | Plot top CpGs location in genome. |
|--|-----------------------------------|

plot\_cpgs (*self*, *top\_cpgs*, *output\_dir='./'*)
Plot top CpGs location in genome.

## **Parameters**

**top\_cpgs** [type] Input dataframe of top CpG name and SHAP scores. Future: Plot SHAP scores in locations?

output\_dir [type] Where to output plots.

class methylnet.interpretation\_classes.ShapleyData

Store SHAP results that stores feature importances of CpGs on varying degrees granularity.

## **Attributes**

top\_cpgs [type] Quick accessible CpGs that have the n highest SHAP scores.

**shapley\_values** [type] Storing all SHAPley values for CpGs for varying degrees granularity. For classification problems, this only includes SHAP scores particular to the actual class.

#### **Methods**

| add_class(self, class_name, shap_df, cpgs,) | Store SHAP scores for particular class to Shap-<br>ley_Data class. |
|---|--|
| add_global_importance(self,)                | Add overall feature importances, globally across all samples.      |
| from_pickle(input_pkl)                      | Load SHAPley data from pickle.                                     |
| to_pickle(self, output_pkl)                 | Export Shapley data to pickle.                                     |

add\_class (self, class\_name, shap\_df, cpgs, n\_top\_cpgs, add\_top\_negative=False)

Store SHAP scores for particular class to Shapley\_Data class. Save feature importances on granular level, explanations for individual and aggregate predictions.

## **Parameters**

class\_name [type] Particular class name to be stored in dictionary structure.

**shap\_df** [type] SHAP data to pull from.

cpgs [type] All CpGs.

**n\_top\_cpgs** [type] Number of top CpGs for saving n top CpGs.

add\_top\_negative [type] Only consider top negative scores.

add\_global\_importance (self, global\_importance\_shaps, cpgs, n\_top\_cpgs)

Add overall feature importances, globally across all samples.

#### **Parameters**

**global\_importance\_shaps** [type] Overall SHAP values to be saved, aggregated across all samples.

cpgs [type] All CpGs.

**n\_top\_cpgs** [type] Number of top CpGs to rank and save as well.

## classmethod from\_pickle(input\_pkl)

Load SHAPley data from pickle.

#### **Parameters**

input\_pkl [type] Input pickle.

to\_pickle (self, output\_pkl)

Export Shapley data to pickle.

#### **Parameters**

output\_pkl [type] Output file to save SHAPley data.

class methylnet.interpretation\_classes.ShapleyDataExplorer(shapley\_data)
 Datatype used to explore saved ShapleyData.

#### **Parameters**

**shapley\_data** [type] ShapleyData instance to be explored.

## **Attributes**

indiv2class [type] Maps individuals to their classes used for quick look-up.

shapley\_data

#### **Methods**

| add_abs_value_classes(self)                   | WIP.  |
|---|---|
| add_bin_continuous_classes(self)              | Bin continuous outcome variable and extract top pos-  |
|   | itive and negative CpGs for each new class.           |
| <pre>extract_class(self, class_name[,])</pre> | Extract the top cpgs from a class                     |
| extract_individual(self, individual)          | Extract the top cpgs from an individual               |
| extract_methylation_array(self,               | Subset MethylationArray Beta values by some top       |
| methyl_arr)                                   | SHAP CpGs from classes or overall.                    |
| jaccard_similarity_top_cpgs(self,             | Calculate Jaccard Similarity matrix between classes   |
| class_names)                                  | and individuals within those classes based on how     |
|   | they share sets of CpGs.                              |
| limit_number_top_cpgs(self, n_top_cpgs)       | Reduce the number of top CpGs.                        |
| list_classes(self)                            | List classes in ShapleyData object.                   |
| list_individuals(self[, return_list])         | List the individuals in the ShapleyData object.       |
| regenerate_individual_shap_values(self,       | Use original SHAP scores to make nested dictionary    |
| )   | of top CpGs based on shapley score, can do this for   |
|   | ABS SHAP or Negative SHAP scores as well.             |
| return_binned_shapley_data(self,[,            | Converts existing shap data based on continuous       |
| ])  | variable predictions into categorical variable.       |
| return_cpg_sets(self)                         | Return top sets of CpGs from classes and individuals, |
|   | and the complementary set.                            |
| return_global_importance_cpgs(self)           | Return overall globally important CpGs.               |
|   | Continued on payt page                                |

Continued on next page

## Table 5 – continued from previous page

|                                      | · · · · · · · · · · · · · · · · · · ·                   |
|--------------------------------------|---|
| return_shapley_data_by_methylation_s | stRetur(.dic)tionary containing two SHAPley datasets,   |
|                                      | each split by low/high levels of methylation.           |
| return_top_cpgs(self[, classes,])    | Given list of classes and individuals, export a dictio- |
|                                      | nary containing data frames of top CpGs and their       |
|                                      | SHAP scores.  |
| view_methylation(self, individual,   | Use MethylationArray to output top SHAP values for      |
| methyl_arr)                          | individual with methylation data appended to output     |
|                                      | data frame.   |

## add\_abs\_value\_classes(self)

WIP.

#### add\_bin\_continuous\_classes(self)

Bin continuous outcome variable and extract top positive and negative CpGs for each new class.

## extract\_class (self, class\_name, class\_intersect=False)

Extract the top cpgs from a class

#### **Parameters**

**class\_name** [type] Class to extract from?

class\_intersect [type] Bool to extract from aggregation of SHAP values from individuals of current class, should have been done already.

#### **Returns**

DataFrame Cpgs and SHAP Values

#### extract\_individual (self, individual)

Extract the top cpgs from an individual

## **Parameters**

**individual** [type] Individual to extract from?

#### Returns

Tuple Class name of individual, DataFrame of Cpgs and SHAP Values

extract\_methylation\_array (self, *methyl\_arr*, classes\_only=True, global\_vals=False,  $n\_extract=1000, class\_name=")$  Subset MethylationArray Beta values by some top SHAP CpGs from classes or overall.

#### **Parameters**

**methyl** arr [type] MethylationArray.

classes only [type] Setting this to False will include the overall top CpGs per class and the top CpGs for individuals in that class.

global\_vals [type] Use top CpGs overall, across the entire dataset.

**n\_extract** [type] Number of top CpGs to subset.

class\_name [type] Which class to subset top CpGs from? Blank to use union of all top CpGs.

#### **Returns**

**MethylationArray** Stores methylation beta and pheno data, reduced here by queried CpGs.

```
jaccard_similarity_top_cpgs (self, class_names, individuals=False, overall=False, cooccur-
rence=False)
```

Calculate Jaccard Similarity matrix between classes and individuals within those classes based on how they share sets of CpGs.

#### **Parameters**

class\_names [type] Classes to include.

individuals [type] Individuals to include.

**overall** [type] Whether to use overall class top CpGs versus aggregate.

**cooccurrence** [type] Output cooccurence instead of jaccard.

#### Returns

pd.DataFrame Similarity matrix.

#### limit\_number\_top\_cpgs (self, n\_top\_cpgs)

Reduce the number of top CpGs.

#### **Parameters**

**n\_top\_cpgs** [type] Number top CpGs to retain.

#### Returns

**ShapleyData** Returns shapley data with fewer top CpGs.

## list\_classes (self)

List classes in ShapleyData object.

#### Returns

List List of classes

## list\_individuals (self, return\_list=False)

List the individuals in the ShapleyData object.

#### **Parameters**

**return\_list** [type] Return list of individual names rather than a dictionary of class:individual key-value pairs.

#### Returns

List or Dictionary Class: Individual dictionary or individuals are elements of list

## regenerate\_individual\_shap\_values (self, n\_top\_cpgs, abs\_val=False, neg\_val=False)

Use original SHAP scores to make nested dictionary of top CpGs based on shapley score, can do this for ABS SHAP or Negative SHAP scores as well.

## **Parameters**

**n\_top\_cpgs** [type] Description of parameter *n\_top\_cpgs*.

**abs\_val** [type] Description of parameter *abs\_val*.

**neg\_val** [type] Description of parameter *neg\_val*.

## Returns

type Description of returned object.

## return\_binned\_shapley\_data(self,

original\_class\_name,

outcome\_col,

add\_top\_negative=False)

Converts existing shap data based on continuous variable predictions into categorical variable.

#### **Parameters**

**original\_class\_name** [type] Regression results were split into ClassName\_pos and ClassName\_neg, what is the ClassName?

outcome\_col [type] Feed in from pheno sheet one the column to bin samples on.

**add\_top\_negative** [type] Looking to include negative SHAPs?

#### Returns

ShapleyData With new class labels, built from regression results.

## return\_cpg\_sets(self)

Return top sets of CpGs from classes and individuals, and the complementary set. Returns dictionary of sets of CpGs and the union of all the other sets minus the current set.

#### Returns

cpg\_sets Dictionary of individuals and classes; CpGs contained in set for particular individual/class

cpg\_exclusion\_sets Dictionary of individuals and classes; CpGs not contained in set for particular individual/class

## return\_global\_importance\_cpgs (self)

Return overall globally important CpGs.

#### Returns

list

## return\_shapley\_data\_by\_methylation\_status(self, methyl\_array)

Return dictionary containing two SHAPley datasets, each split by low/high levels of methylation. Todo: Define this using median methylation value vs 0.5.

#### **Parameters**

methyl\_array [type] MethylationArray instance.

#### **Returns**

**dictionary** Contains shapley data by methylation status.

```
return_top_cpgs (self, classes=[], individuals=[], class_intersect=False, cpg exclusion sets=None, cpg sets=None)
```

Given list of classes and individuals, export a dictionary containing data frames of top CpGs and their SHAP scores.

#### **Parameters**

**classes** [type] Higher level classes to extract CpGs from, list of classes to extract top CpGs from.

individuals [type] Individual samples to extract top CpGs from.

**class\_intersect** [type] Whether the top CpGs should be chosen by aggregating the remaining individual scores.

**cpg\_exclusion\_sets** [type] Dict of sets of CpGs, where these ones contain CpGs not particular to particular class.

**cpg\_sets** [type] Contains top CpGs found for each class.

## Returns

**Dict** Top CpGs accessed and returned for further processing.

data frame.

view\_methylation (self, individual, methyl\_arr)

```
Parameters
                  individual [type] Individual to query from ShapleyData
                  methyl_arr [type] Input MethylationArray
              Returns
                  class name
                  individual
                  top_shap_df Top Shapley DataFrame with top cpgs for individual, methylation data ap-
                    pended.
methylnet.interpretation_classes.cooccurrence_fn(list1, list2)
     Cooccurence of elements between two lists.
          Parameters
              list1
              list2
          Returns
              float Cooccurence between elements in list.
methylnet.interpretation_classes.jaccard_similarity(list1, list2)
     Jaccard score between two lists.
          Parameters
              list1
              list2
          Returns
              float Jaccard similarity between elements in list.
methylnet.interpretation_classes.main_prediction_function(n_workers, batch_size,
                                                                              model, cuda)
     Combine dataloader and prediction function into final prediction function that takes in tensor data, for Kernel
     Explanations.
          Parameters
              n_workers [type] Number of workers.
              batch size [type] Batch size for input data.
              model [type] Model/predidction function.
              cuda [type] Running on GPUs?
          Returns
              function Final prediction function.
methylnet.interpretation_classes.plot_lola_output_(lola_csv, plot_output_dir, descrip-
                                                                    tion_col, cell_types)
     Plots any LOLA output in the form of a forest plot.
          Parameters
```

Use MethylationArray to output top SHAP values for individual with methylation data appended to output

20

```
lola_csv [type] CSV containing lola results.
              plot_output_dir [type] Plot output directory.
              description_col [type] Column that will label the points on the plot.
              cell_types [type] Column containing cell-types, colors plots and are rows to be compared.
methylnet.interpretation_classes.return_dataloader_construct (n_workers,
                                                                                  batch size)
     Decorator to build dataloader compatible with KernelExplainer for SHAP.
          Parameters
              n_workers [type] Number CPU.
              batch_size [type] Batch size to load data into pytorch.
          Returns
              DataLoader Pytorch dataloader.
methylnet.interpretation_classes.return_predict_function(model, cuda)
     Decorator to build the supplied prediction function, important for kernel explanations.
          Parameters
              model [type] Prediction function with predict method.
              cuda [type] Run using cuda.
          Returns
              function Predict function.
methylnet.interpretation_classes.return_shap_values(test_arr,
                                                                                 explainer,
                                                                      n_samples, additional_opts)
     Return SHAP values, sampled a number of times, used in CpGExplainer class.
          Parameters
              test_arr [type] Testing MethylationArray.
              explainer [type] SHAP explainer object.
              method [type] Method of explaining.
              n_samples [type] Number of samples to estimate SHAP scores.
              additional_opts [type] Additional options to be passed into non-kernel/gradient methods.
          Returns
              np.array/list Shapley scores.
methylnet.interpretation_classes.to_tensor(arr)
     Turn np.array into tensor.
```

## **MODELS.PY**

Contains core PyTorch Models for running VAE and VAE-MLP.

Wraps Pytorch VAE module into Scikit-learn like interface for ease of training, validation and testing.

#### **Parameters**

autoencoder\_model [type] Pytorch VAE Model to supply.

**n\_epochs** [type] Number of epochs to train for.

loss\_fn [type] Pytorch loss function for reconstruction error.

optimizer [type] Pytorch Optimizer.

cuda [type] GPU?

kl\_warm\_up [type] Number of epochs until fully utilizing KLLoss, begin saving models here.

beta [type] Weighting for KLLoss.

**scheduler\_opts** [type] Options to feed learning rate scheduler, which modulates learning rate of optimizer.

## Attributes

model [type] Pytorch VAE model.

scheduler [type] Learning rate scheduler object.

vae\_animation\_fname [type] Save VAE embeddings evolving over epochs to this file name. Defunct for now.

**loss\_plt\_fname** [type] Where to save loss curves. This has been superceded by plot\_training\_curves in methylnet-visualize command.

plot\_interval [type] How often to plot data; defunct.

embed\_interval [type] How often to embed; defunct.

validation\_set [type] MethylationArray DataLoader, produced from Pytorch Methylation-Dataset of Validation MethylationArray.

n\_epochs

loss\_fn

optimizer

cuda

#### kl\_warm\_up

beta

## **Methods**

| add_validation_set(self, validation_data) | Add validation data in the form of Validation Dat-  |
|---|---|
|   | aLoader.  |
| fit(self, train_data)                     | Fit VAE model to training data, best model returned |
|   | with lowest validation loss over epochs.            |
| fit_transform(self, train_data)           | Fit VAE model and transform Methylation Array us-   |
|   | ing VAE model.                                      |
| transform(self, train_data)               |   |
|   | Parameters  |

## add\_validation\_set (self, validation\_data)

Add validation data in the form of Validation DataLoader. Adding this will use validation data for early termination / generalization of model to unseen data.

#### **Parameters**

validation\_data [type] Pytorch DataLoader housing validation MethylationDataset.

## fit (self, train\_data)

Fit VAE model to training data, best model returned with lowest validation loss over epochs.

#### **Parameters**

**train\_data** [DataLoader] Training DataLoader that is loading MethylationDataset in batches.

## Returns

self Autoencoder object with updated VAE model.

## fit\_transform(self, train\_data)

Fit VAE model and transform Methylation Array using VAE model.

#### **Parameters**

train\_data [type] Pytorch DataLoader housing training MethylationDataset.

## Returns

np.array Latent Embeddings.

np.array Sample names from MethylationArray

np.array Outcomes from column of methylarray.

## transform(self, train\_data)

## **Parameters**

train\_data [type] Pytorch DataLoader housing training MethylationDataset.

#### **Returns**

np.array Latent Embeddings.

np.array Sample names from MethylationArray

```
np.array Outcomes from column of methylarray.
```

```
class methylnet.models.MLPFinetuneVAE(mlp_model, n_epochs=None, loss_fn=None, opti-
                                                     mizer vae=None, optimizer mlp=None, cuda=True,
                                                     categorical=False,
                                                                            scheduler_opts={},
                                                     put_latent=True, train_decoder=False)
     Wraps VAE_MLP pytorch module into scikit-learn interface with fit, predict and fit_predict methods for ease-
     of-use model training/evaluation.
           Parameters
               mlp_model [type] VAE_MLP model.
               n_epochs [type] Number epochs train for.
               loss_fn [type] Loss function, pytorch, CrossEntropy, BCE, MSE depending on outcome.
               optimizer_vae [type] Optimizer for VAE layers for finetuning original pretrained network.
               optimizer_mlp [type] Optimizer for new appended MLP layers.
               cuda [type] GPU?
               categorical [type] Classification or regression outcome?
               scheduler opts [type] Options for learning rate scheduler, modulates learning rates for VAE
                   and MLP.
               output latent [type] Whether to output latent embeddings during evaluation.
               train_decoder [type] Retrain decoder to adjust for finetuning of VAE?
           Attributes
               model [type] VAE_MLP.
               scheduler_vae [type] Learning rate modulator for VAE optimizer.
               scheduler_mlp [type] Learning rate modulator for MLP optimizer.
               loss_plt_fname [type] File where to plot loss over time; defunct.
               embed_interval [type] How often to return embeddings; defunct.
               validation_set [type] Validation set used for hyperparameter tuning and early stopping criteria
                   for generalization.
               return latent [type] Return embedding during evaluation?
               n epochs
               loss fn
               optimizer_vae
               optimizer mlp
               cuda
               categorical
               output_latent
               train decoder
```

## **Methods**

| add_validation_set(self, validation_data) | Add validation data to reduce overfitting.    |
|---|---|
| fit(self, train_data)                     | Fit MLP to training data to make predictions. |
| <pre>predict(self, test_data)</pre>       | Short summary.                                |

```
add_validation_set (self, validation_data)
```

Add validation data to reduce overfitting.

#### **Parameters**

validation\_data [type] Validation Dataloader MethylationDataset.

fit (self, train\_data)

Fit MLP to training data to make predictions.

#### **Parameters**

train\_data [type] DataLoader with Training MethylationDataset.

#### Returns

**self** MLPFinetuneVAE with updated parameters.

predict (self, test data)

Short summary.

#### **Parameters**

test\_data [type] Test DataLoader MethylationDataset.

#### Returns

np.array Predictions

np.array Ground truth

np.array Latent Embeddings

np.array Sample names.

class methylnet.models.TybaltTitusVAE (n\_input,

n latent,

den\_layer\_encoder\_topology=[100, 100, 100],

hid-

*cuda=False*)

Pytorch NN Module housing VAE with fully connected layers and customizable topology.

## **Parameters**

**n\_input** [type] Number of input CpGs.

**n\_latent** [type] Size of latent embeddings.

**hidden\_layer\_encoder\_topology** [type] List, length of list contains number of hidden layers for encoder, and each element is number of neurons, mirrored for decoder.

cuda [type] GPU?

## Attributes

cuda\_on [type] GPU?

pre\_latent\_topology [type] Hidden layer topology for encoder.

**post\_latent\_topology** [type] Mirrored hidden layer topology for decoder.

encoder\_layers [list] Encoder pytorch layers.

**encoder** [type] Encoder layers wrapped into pytorch module.

**z\_mean** [type] Linear layer from last encoder layer to mean layer.

**z\_var** [type] Linear layer from last encoder layer to var layer.

**z\_develop** [type] Linear layer connecting sampled latent embedding to first layer decoder.

decoder\_layers [type] Decoder layers wrapped into pytorch module.

output\_layer [type] Linear layer connecting last decoder layer to output layer, which is same size as input..

decoder [type] Wraps decoder\_layers and output\_layers into Sequential module.

n\_input

n\_latent

## **Methods**

| call(self, \*input, \*\*kwargs)                        | Call self as a function.                                 |
|--|--|
| add_module(self, name, module)                         | Adds a child module to the current module.               |
| apply(self, fn)  | Applies fn recursively to every submodule (as re-        |
|  | turned by .children()) as well as self.                  |
| buffers(self[, recurse])                               | Returns an iterator over module buffers.                 |
| children(self)   | Returns an iterator over immediate children modules.     |
| cpu(self)  | Moves all model parameters and buffers to the CPU.       |
| cuda(self[, device])                                   | Moves all model parameters and buffers to the GPU.       |
| decode(self, z)  | Decode latent embeddings back into reconstructed         |
|  | input.   |
| double(self)   | Casts all floating point parameters and buffers to       |
|  | double datatype.   |
| encode(self, x)  | Encode input into latent representation.                 |
| eval(self)   | Sets the module in evaluation mode.                      |
| extra_repr(self)                                       | Set the extra representation of the module               |
| float(self)  | Casts all floating point parameters and buffers to float |
|  | datatype.  |
| forward(self, x)                                       | Return reconstructed output, mean and variance of        |
|  | embeddings.  |
| forward_predict(self, x)                               | Forward pass from input to reconstructed input.          |
| <pre>get_latent_z(self, x)</pre>                       | Encode X into reparameterized latent representation.     |
| half(self)   | Casts all floating point parameters and buffers to       |
|  | half datatype.   |
| <pre>load_state_dict(self, state_dict[, strict])</pre> | Copies parameters and buffers from state_dict            |
|  | into this module and its descendants.                    |
| modules(self)  | Returns an iterator over all modules in the network.     |
| <pre>named_buffers(self[, prefix, recurse])</pre>      | Returns an iterator over module buffers, yielding        |
|  | both the name of the buffer as well as the buffer it-    |
|  | self.  |
| named_children(self)                                   | Returns an iterator over immediate children modules,     |
|  | yielding both the name of the module as well as the      |
|  | module itself.   |
|  | Continued on next page                                   |

Continued on next page

Table 3 – continued from previous page

|  | <u> </u>   |
|--|--|
| <pre>named_modules(self[, memo, prefix])</pre>       | Returns an iterator over all modules in the network, |
|  | yielding both the name of the module as well as the  |
|  | module itself.                                       |
| <pre>named_parameters(self[, prefix, recurse])</pre> | Returns an iterator over module parameters, yielding |
|  | both the name of the parameter as well as the param- |
|  | eter itself.   |
| parameters(self[, recurse])                          | Returns an iterator over module parameters.          |
| register_backward_hook(self, hook)                   | Registers a backward hook on the module.             |
| register_buffer(self, name, tensor)                  | Adds a persistent buffer to the module.              |
| register_forward_hook(self, hook)                    | Registers a forward hook on the module.              |
| register_forward_pre_hook(self, hook)                | Registers a forward pre-hook on the module.          |
| register_parameter(self, name, param)                | Adds a parameter to the module.                      |
| sample_z(self, mean, logvar)                         | Sample latent embeddings, reparameterize by adding   |
|  | noise to embedding.                                  |
| state_dict(self[, destination, prefix,])             | Returns a dictionary containing a whole state of the |
|  | module.  |
| to(self, \*args, \*\*kwargs)                         | Moves and/or casts the parameters and buffers.       |
| train(self[, mode])                                  | Sets the module in training mode.                    |
| type(self, dst_type)                                 | Casts all parameters and buffers to dst_type.        |
| zero_grad(self)                                      | Sets gradients of all model parameters to zero.      |
|  |  |

share\_memory

## decode(self, z)

Decode latent embeddings back into reconstructed input.

## **Parameters**

**z** [type] Reparameterized latent embedding.

## Returns

torch.tensor Reconstructed input.

## encode (self, x)

Encode input into latent representation.

#### **Parameters**

**x** [type] Input methylation data.

## Returns

torch.tensor Learned mean vector of embeddings.

torch.tensor Learned variance of learned mean embeddings.

## forward (self, x)

Return reconstructed output, mean and variance of embeddings.

## $forward\_predict(self, x)$

Forward pass from input to reconstructed input.

## $get_latent_z (self, x)$

Encode X into reparameterized latent representation.

#### **Parameters**

**x** [type] Input methylation data.

#### Returns

torch.tensor Latent embeddings.

sample\_z (self, mean, logvar)

Sample latent embeddings, reparameterize by adding noise to embedding.

#### **Parameters**

mean [type] Learned mean vector of embeddings.

logvar [type] Learned variance of learned mean embeddings.

#### Returns

**torch.tensor** Mean + noise, reparameterization trick.

class methylnet.models.VAE\_MLP ( $vae\_model$ ,  $n\_output$ , categorical=False,  $hid-den\_layer\_topology=[100, 100, 100]$ ,  $dropout\_p=0.2$ ,  $add\_softmax=False$ )

VAE\_MLP, pytorch module used to both finetune VAE embeddings and simultaneously train downstream MLP layers for classification/regression tasks.

#### **Parameters**

vae\_model [type] VAE pytorch model for methylation data.

**n\_output** [type] Number of outputs at end of model.

categorical [type] Classification or regression problem?

**hidden\_layer\_topology** [type] Hidden Layer topology, list of size number of hidden layers for MLP and each element contains number of neurons per layer.

**dropout\_p** [type] Apply dropout regularization to reduce overfitting.

add\_softmax [type] Softmax the output before evaluation.

## Attributes

vae [type] Pytorch VAE module.

topology [type] List with hidden layer topology of MLP.

mlp\_layers [type] All MLP layers (# layers and neurons per layer)

output\_layer [type] nn.Linear connecting last MLP layer and output nodes.

mlp [type] nn.Sequential wraps all layers into sequential ordered pytorch module.

output\_z [type] Whether to output latent embeddings.

n\_output

categorical

add\_softmax

dropout\_p

## **Methods**

| call(self, \*input, \*\*kwargs)           | Call self as a function.                   |
|---|--|
| <pre>add_module(self, name, module)</pre> | Adds a child module to the current module. |
|   |  |

Continued on next page

Table 4 – continued from previous page

|   | a from previous page                                     |
|---|--|
| apply(self, fn)   | Applies fn recursively to every submodule (as re-        |
|   | turned by .children()) as well as self.                  |
| buffers(self[, recurse])  | Returns an iterator over module buffers.                 |
| children(self)  | Returns an iterator over immediate children modules.     |
| cpu(self)   | Moves all model parameters and buffers to the CPU.       |
| cuda(self[, device])  | Moves all model parameters and buffers to the GPU.       |
| decode(self, z)   | Run VAE decoder on embeddings.                           |
| double(self)  | Casts all floating point parameters and buffers to       |
|   | double datatype.   |
| eval(self)  | Sets the module in evaluation mode.                      |
| extra_repr(self)  | Set the extra representation of the module               |
| float(self)   | Casts all floating point parameters and buffers to float |
| 11000(0011)   | datatype.  |
| forward(self, x)  | Pass data in to return predictions and embeddings.       |
| forward_embed(self, x)  | Return predictions, latent embeddings and recon-         |
| 101 ward_enwed(scn, x)  | structed input.  |
| forward product(salf v)   | Make predictions, based on output_z, either output       |
| <pre>forward_predict(self, x)</pre>                                 |  |
| 1 1 (1( )   | predictions or output embeddings.                        |
| half(self)  | Casts all floating point parameters and buffers to       |
| 2   | half datatype.   |
| <pre>load_state_dict(self, state_dict[, strict])</pre>              | Copies parameters and buffers from state_dict            |
|   | into this module and its descendants.                    |
| modules(self)   | Returns an iterator over all modules in the network.     |
| <pre>named_buffers(self[, prefix, recurse])</pre>                   | Returns an iterator over module buffers, yielding        |
|   | both the name of the buffer as well as the buffer it-    |
|   | self.  |
| named_children(self)  | Returns an iterator over immediate children modules,     |
|   | yielding both the name of the module as well as the      |
|   | module itself.   |
| <pre>named_modules(self[, memo, prefix])</pre>                      | Returns an iterator over all modules in the network,     |
|   | yielding both the name of the module as well as the      |
|   | module itself.   |
| <pre>named_parameters(self[, prefix, recurse])</pre>                | Returns an iterator over module parameters, yielding     |
|   | both the name of the parameter as well as the param-     |
|   | eter itself.   |
| parameters(self[, recurse])   | Returns an iterator over module parameters.              |
| register_backward_hook(self, hook)                                  | Registers a backward hook on the module.                 |
| register_buffer(self, name, tensor)                                 | Adds a persistent buffer to the module.                  |
| register_forward_hook(self, hook)                                   | Registers a forward hook on the module.                  |
| register_forward_pre_hook(self, hook)                               | Registers a forward pre-hook on the module.              |
| register_parameter(self, name, param)                               | Adds a parameter to the module.                          |
| state_dict(self[, destination, prefix,])                            | Returns a dictionary containing a whole state of the     |
| ( L)  | module.  |
| to(self, \*args, \*\*kwargs)  | Moves and/or casts the parameters and buffers.           |
| toggle_latent_z(self)   | Toggle whether to output latent embeddings during        |
| 209910_1420110_2(6011)  |  |
|   | forward pass   |
| train(self[ model)  | forward pass.  Sets the module in training mode          |
| train(self[, mode])   | Sets the module in training mode.                        |
| <pre>train(self[, mode]) type(self, dst_type) zero_grad(self)</pre> |  |

share\_memory

```
Run VAE decoder on embeddings.
               Parameters
                   z [type] Embeddings.
               Returns
                   torch.tensor Reconstructed Input.
     forward (self, x)
          Pass data in to return predictions and embeddings.
               Parameters
                  x [type] Input data.
               Returns
                  torch.tensor Predictions
                  torch.tensor Embeddings
     forward\_embed(self, x)
          Return predictions, latent embeddings and reconstructed input.
               Parameters
                  x [type] Input data
               Returns
                   torch.tensor Predictions
                   torch.tensor Embeddings
                  torch.tensor Reconstructed input.
     forward_predict (self, x)
          Make predictions, based on output_z, either output predictions or output embeddings.
               Parameters
                   x [type] Input Data.
               Returns
                  torch.tensor Predictions or embeddings.
     toggle_latent_z (self)
          Toggle whether to output latent embeddings during forward pass.
methylnet.models.project_vae(model, loader, cuda=True)
     Return Latent Embeddings of any data supplied to it.
          Parameters
               model [type] VAE Pytorch Model.
               loader [type] Loads data one batch at a time.
               cuda [type] GPU?
          Returns
               np.array Latent Embeddings.
               np.array Sample names from MethylationArray
```

decode(self, z)

```
np.array Outcomes from column of methylarray.
```

methylnet.models.test\_mlp(model, loader, categorical, cuda=True, output\_latent=True) Evaluate MLP on testing set, output predictions.

#### **Parameters**

```
model [type] VAE_MLP model.
loader [type] DataLoader with MethylationDataSet
categorical [type] Categorical or continuous predictions.
cuda [type] GPU?
output latent [type] Output latent embeddings in addition to predictions?
```

#### Returns

```
np.array Predictionsnp.array Ground truthnp.array Latent Embeddingsnp.array Sample names.
```

methylnet.models.train\_decoder\_(model, x, z)

Run if retraining decoder to adjust for adjusted latent embeddings during finetuning of embedding layers for VAE MLP.

#### **Parameters**

```
model [type] VAE_MLP model.x [type] Input methylation data.z [type] Latent Embeddings
```

## Returns

**nn.Module** VAE\_MLP module with updated decoder parameters.

float Reconstruction loss over all batches.

Train Multi-layer perceptron appended to latent embeddings of VAE via transfer learning. Do this for one iteration.

#### **Parameters**

```
model [type] VAE_MLP model.
loader [type] DataLoader with MethylationDataset.
loss_func [type] Loss function (BCE, CrossEntropy, MSE).
optimizer_vae [type] Optimizer for pytorch VAE.
optimizer_mlp [type] Optimizer for outcome MLP layers.
cuda [type] GPU?
categorical [type] Predicting categorical or continuous outcomes.
train_decoder [type] Retrain decoder during training loop to adjust for fine-tuned embeddings.
```

## Returns

```
nn.Module Training VAE_MLP model with updated parameters.
```

float Training loss over all batches

```
methylnet.models.train_vae(model, loader, loss_func, optimizer, cuda=True, epoch=0, kl warm up=0, beta=1.0)
```

Function for parameter update during VAE training for one iteration.

#### **Parameters**

model [type] VAE torch model

loader [type] Data loader, generator that calls batches of data.

loss\_func [type] Loss function for reconstruction error, nn.BCELoss or MSELoss

optimizer [type] SGD or Adam pytorch optimizer.

cuda [type] GPU?

**epoch** [type] Epoch of training, passed in from outer loop.

kl\_warm\_up [type] How many epochs until model is fully utilizes KL Loss.

beta [type] Weight given to KL Loss.

#### Returns

nn.Module Pytorch VAE model

float Total Training Loss across all batches

float Total Training reconstruction loss across all batches

float Total KL Loss across all batches

methylnet.models.vae\_loss(output, input, mean, logvar, loss\_func, epoch, kl\_warm\_up=0, beta=1.0)

Function to calculate VAE Loss, Reconstruction Loss + Beta KLLoss.

#### **Parameters**

output [torch.tensor] Reconstructed output from autoencoder.

input [torch.tensor] Original input data.

**mean** [type] Learned mean tensor for each sample point.

logvar [type] Variation around that mean sample point, learned from reparameterization.

**loss\_func** [type] Loss function for reconstruction loss, MSE or BCE.

**epoch** [type] Epoch of training.

kl warm up [type] Number of epochs until fully utilizing KLLoss, begin saving models here.

beta [type] Weighting for KLLoss.

## Returns

torch.tensor Total loss

torch.tensor Recon loss

torch.tensor KL loss

methylnet.models.val\_decoder\_(model, x, z)

Validation Loss over decoder.

#### **Parameters**

```
model [type] VAE_MLP model.
```

- x [type] Input methylation data.
- z [type] Latent Embeddings

#### Returns

**float** Reconstruction loss over all batches.

methylnet.models.val\_mlp(model, loader, loss\_func, cuda=True, categorical=False, train\_decoder=False)

Find validation loss of VAE MLP over one Epoch.

#### **Parameters**

model [type] VAE\_MLP model.

loader [type] DataLoader with MethylationDataset.

loss\_func [type] Loss function (BCE, CrossEntropy, MSE).

cuda [type] GPU?

categorical [type] Predicting categorical or continuous outcomes.

train\_decoder [type] Retrain decoder during training loop to adjust for fine-tuned embeddings.

#### Returns

nn.Module VAE MLP model.

**float** Validation loss over all batches

methylnet.models.val\_vae(model, loader, loss\_func, optimizer, cuda=True, epoch=0, kl warm up=0, beta=1.0)

Function for validation loss computation during VAE training for one epoch.

#### **Parameters**

model [type] VAE torch model

loader [type] Validation Data loader, generator that calls batches of data.

loss\_func [type] Loss function for reconstruction error, nn.BCELoss or MSELoss

**optimizer** [type] SGD or Adam pytorch optimizer.

cuda [type] GPU?

epoch [type] Epoch of training, passed in from outer loop.

kl warm up [type] How many epochs until model is fully utilizes KL Loss.

beta [type] Weight given to KL Loss.

#### Returns

nn.Module Pytorch VAE model

float Total Validation Loss across all batches

float Total Validation reconstruction loss across all batches

float Total Validation KL Loss across all batches

SIX

## **PLOTTER.PY**

Plotting mechanisms for training that are now defuct.

**class** methylnet.plotter.**Plot**(*title*, *xlab='vae1'*, *ylab='vae2'*, *data=[]*)
Stores plotting information; defunct, superceded, see methylnet-visualize.

## **Methods**

return\_img

class methylnet.plotter.PlotTransformer(data, color\_list)

Plotting Transformer to help with plotting embeddings changing over epochs; defunct.

## **Methods**

transform

class methylnet.plotter.Plotter(plots, animation=True)

Plot embeddings and training curve from Plot objects; defunct, superceded, see methylnet-visualize.

## **Methods**

animate write\_plots

## SEVEN

## SCHEDULERS.PY

Learning rate schedulers that help enable better and more generalizable models.

class methylnet.schedulers.CosineAnnealingWithRestartsLR (optimizer,  $T\_max$ ,  $eta\_min=0$ ,  $last\_epoch=-1$ ,  $T\_mult=1.0$ ,  $al-pha\ decay=1.0$ )

Borrowed from: https://github.com/mpyrozhok/adamwr/blob/master/cyclic\_scheduler.py Needs to be updated to reflect newest changes. From original docstring: Set the learning rate of each parameter group using a cosine annealing schedule, where  $\eta_{max}$  is set to the initial lr and  $T_{cur}$  is the number of epochs since the last restart in SGDR:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + \cos(\frac{T_{cur}}{T_{max}}\pi))$$

When last\_epoch=-1, sets initial lr as lr. It has been proposed in

SGDR: Stochastic Gradient Descent with Warm Restarts. This implements the cosine annealing part of SGDR, the restarts and number of iterations multiplier.

**Args:** optimizer (Optimizer): Wrapped optimizer. T\_max (int): Maximum number of iterations. T\_mult (float): Multiply T\_max by this number after each restart. Default: 1. eta\_min (float): Minimum learning rate. Default: 0. last\_epoch (int): The index of last epoch. Default: -1.

**Attributes** 

step\_n

## **Methods**

| load_state_dict(self, state_dict) | Loads the schedulers state.                   |
|-----------------------------------|---|
| state_dict(self)                  | Returns the state of the scheduler as a dict. |

| cosine  |  |
|---------|--|
| get_lr  |  |
| restart |  |
| step    |  |

Scheduler class that modulates learning rate of torch optimizers over epochs.

## **Parameters**

```
optimizer [type] torch.Optimizer objectopts [type] Options of setting the learning rate scheduler, see default.
```

#### Attributes

schedulers [type] Different types of schedulers to choose from.
scheduler\_step\_fn [type] How scheduler updates learning rate.
initial\_lr [type] Initial set learning rate.

**scheduler\_choice** [type] What scheduler type was chosen.

scheduler [type] Scheduler object chosen that will more directly update optimizer LR.

## **Methods**

| get_1r(self) | Return current learning rate.  |
|--------------|--------------------------------|
| step(self)   | Update optimizer learning rate |

get\_lr (self)

Return current learning rate.

## Returns

float Current learning rate.

step(self)

Update optimizer learning rate

## **EIGHT**

## METHYLNET-EMBED

```
methylnet-embed [OPTIONS] COMMAND [ARGS]...
```

## **Options**

#### --version

Show the version and exit.

## 8.1 launch\_hyperparameter\_scan

Launch randomized grid search of neural network hyperparameters.

```
methylnet-embed launch_hyperparameter_scan [OPTIONS]
```

## **Options**

- -g, --generate\_input

Generate hyperparameter input csv.

-c, --job\_chunk\_size <job\_chunk\_size>
If not series, chunk up and run these number of commands at once..

-sc, --stratify\_column <stratify\_column>
 Column to stratify samples on. [default: disease]

-r, --reset\_all

Run all jobs again.

-t, --torque

Submit jobs on torque.

-gpu, --gpu <gpu>

If torque submit, which gpu to use. [default: -1]

-gn, --gpu\_node <gpu\_node>

If torque submit, which gpu node to use. [default: -1]

## -nh, --nohup

Nohup launch jobs.

-mc, --model\_complexity\_factor <model\_complexity\_factor>

Degree of neural network model complexity for hyperparameter search. Search for less wide and less deep networks with a lower complexity value, bounded between 0 and infinity. [default: 1.0]

-b, --set beta <set beta>

Set beta value, bounded between 0 and infinity. Set to -1 [default: 1.0]

-j, --n\_jobs <n\_jobs>

Number of jobs to generate.

-n, --n\_jobs\_relaunch <n\_jobs\_relaunch>

Relaunch n top jobs from previous run. [default: 0]

-c, --crossover\_p <crossover\_p>

Rate of crossover between hyperparameters. [default: 0.0]

-v, --val\_loss\_column <val\_loss\_column>

Validation loss column.

-a, --additional\_command <additional\_command>

Additional command to input for torque run.

## 8.2 perform\_embedding

Perform variational autoencoding on methylation dataset.

methylnet-embed perform\_embedding [OPTIONS]

## **Options**

-i, --train\_pkl <train\_pkl>

Input database for beta and phenotype data. [default: ./train\_val\_test\_sets/train\_methyl\_array.pkl]

-o, --output\_dir <output\_dir>

Output directory for embeddings. [default: ./embeddings/]

-c, --cuda

Use GPUs.

-n, --n\_latent <n\_latent>

Number of latent dimensions. [default: 64]

-lr, --learning\_rate <learning\_rate>

Learning rate. [default: 0.001]

-wd, --weight\_decay <weight\_decay>

Weight decay of adam optimizer. [default: 0.0001]

-e, --n\_epochs <n\_epochs>

Number of epochs to train over. [default: 50]

-hlt, --hidden\_layer\_encoder\_topology <hidden\_layer\_encoder\_topology>

Topology of hidden layers, comma delimited, leave empty for one layer encoder, eg. 100,100 is example of 5-hidden layer topology. [default: ]

#### -kl, --kl\_warm\_up <kl\_warm\_up>

Number of epochs before introducing kl\_loss. [default: 0]

## -b, --beta <beta>

Weighting of kl divergence. [default: 1.0]

#### -s, --scheduler <scheduler>

Type of learning rate scheduler. [default: null]

#### -d, --decay <decay>

Learning rate scheduler decay for exp selection. [default: 0.5]

## -t, --t\_max <t\_max>

Number of epochs before cosine learning rate restart. [default: 10]

#### -eta, --eta\_min <eta\_min>

Minimum cosine LR. [default: 1e-06]

#### -m, --t\_mult <t\_mult>

Multiply current restart period times this number given number of restarts. [default: 2.0]

#### -bce, --bce loss

Use bce loss instead of MSE.

## -bs, --batch\_size <batch\_size>

Batch size. [default: 50]

#### -vp, --val\_pkl <val\_pkl>

Validation Set Methylation Array Location. [default: ./train\_val\_test\_sets/val\_methyl\_array.pkl]

## -w, --n\_workers <n\_workers>

Number of workers. [default: 9]

## -conv, --convolutional

Use convolutional VAE.

## -hs, --height\_kernel\_sizes <height\_kernel\_sizes>

Heights of convolutional kernels.

## -ws, --width\_kernel\_sizes <width\_kernel\_sizes>

Widths of convolutional kernels.

## -v, --add\_validation\_set

Evaluate validation set.

## -1, --loss\_reduction <loss\_reduction>

Type of reduction on loss function. [default: sum]

#### -hl, --hyperparameter log <hyperparameter log>

CSV file containing prior runs. [default: embeddings/embed\_hyperparameters\_log.csv]

## -sc, --stratify\_column <stratify\_column>

Column to stratify samples on. [default: disease]

## -j, --job\_name < job\_name>

Embedding job name. [default: embed\_job]

NINE

## **METHYLNET-PREDICT**

methylnet-predict [OPTIONS] COMMAND [ARGS]...

## **Options**

#### --version

Show the version and exit.

## 9.1 classification\_report

Generate classification report that gives results from classification tasks.

methylnet-predict classification\_report [OPTIONS]

## **Options**

- -r, --results\_pickle <results\_pickle>
   Results from training, validation, and testing. [default: predictions/results.p]
- -o, --output\_dir <output\_dir>
   Output directory. [default: results/]
- -e, --categorical\_encoder <categorical\_encoder>

One hot encoder if categorical model. If path exists, then return top positive controbutions per samples of that class. Encoded values must be of sample class as interest\_col. [default: ./predictions/one\_hot\_encoder.p]

## 9.2 launch\_hyperparameter\_scan

Run randomized hyperparameter scan of neural network hyperparameters.

methylnet-predict launch\_hyperparameter\_scan [OPTIONS]

## **Options**

## -g, --generate\_input

Generate hyperparameter input csv.

-c, --job\_chunk\_size <job\_chunk\_size>

If not series, chunk up and run these number of commands at once..

-ic, --interest\_cols <interest\_cols>

Column to stratify samples on.

-cat, --categorical

Whether to run categorical analysis or not.

-r, --reset\_all

Run all jobs again.

-t, --torque

Submit jobs on torque.

**-gpu**, **--gpu** <gpu>

If torque submit, which gpu to use. [default: -1]

-gn, --gpu\_node <gpu\_node>

If torque submit, which gpu node to use. [default: -1]

-nh, --nohup

Nohup launch jobs.

-n, --n\_jobs\_relaunch <n\_jobs\_relaunch>

Relaunch n top jobs from previous run. [default: 0]

-c, --crossover\_p <crossover\_p>

Rate of crossover between hyperparameters. [default: 0.0]

-mc, --model\_complexity\_factor <model\_complexity\_factor>

Degree of neural network model complexity for hyperparameter search. Search for less wide networks with a lower complexity value, bounded between 0 and infinity. [default: 1.0]

-j, --n\_jobs <n\_jobs>

Number of jobs to generate.

-v, --val\_loss\_column <val\_loss\_column>

Validation loss column.

-sft, --add\_softmax

Add softmax for predicting probability distributions.

-a, --additional\_command <additional\_command>

Additional command to input for torque run.

## 9.3 make\_new\_predictions

Run prediction model again to further assess outcome. Only evaluate prediction model.

methylnet-predict make\_new\_predictions [OPTIONS]

## **Options**

```
-tp, --test_pkl <test_pkl>
     Test database for beta and phenotype data. [default: ./train_val_test_sets/test_methyl_array.pkl]
-m, --model_pickle <model_pickle>
     Pytorch model containing forward_predict method. [default: ./predictions/output_model.p]
-bs, --batch_size <batch_size>
     Batch size. [default: 50]
-w, --n_workers <n_workers>
     Number of workers. [default: 9]
-ic, --interest_cols <interest_cols>
     Specify columns looking to make predictions on. [default: disease]
-cat, --categorical
     Multi-class prediction. [default: False]
-c, --cuda
     Use GPUs.
-e, --categorical_encoder <categorical_encoder>
     One hot encoder if categorical model. If path exists, then return top positive controbutions per samples of that
     class. Encoded values must be of sample class as interest_col. [default: ./predictions/one_hot_encoder.p]
-o, --output_dir <output_dir>
     Output directory for predictions. [default: ./new_predictions/]
```

## 9.4 make prediction

Train prediction model by fine-tuning VAE and appending/training MLP to make classification/regression predictions on MethylationArrays.

```
methylnet-predict make_prediction [OPTIONS]
```

## -cat, --categorical Multi-class prediction. [default: False] -do, --disease only Only look at disease, or text before subtype\_delimiter. -hlt, --hidden\_layer\_topology <hidden\_layer\_topology> Topology of hidden layers, comma delimited, leave empty for one layer encoder, eg. 100,100 is example of 5-hidden layer topology. [default: ] -lr\_vae, --learning\_rate\_vae <learning\_rate\_vae> Learning rate VAE. [default: 1e-05] -lr\_mlp, --learning\_rate\_mlp <learning\_rate\_mlp> Learning rate MLP. [default: 0.001] -wd, --weight\_decay <weight\_decay> Weight decay of adam optimizer. [default: 0.0001] -dp, --dropout\_p <dropout\_p> Dropout Percentage. [default: 0.2] -e, --n\_epochs <n\_epochs> Number of epochs to train over. [default: 50] -s, --scheduler <scheduler> Type of learning rate scheduler. [default: null] -d, --decay <decay> Learning rate scheduler decay for exp selection. [default: 0.5] -t, --t\_max <t\_max> Number of epochs before cosine learning rate restart. [default: 10] -eta, --eta min <eta min> Minimum cosine LR. [default: 1e-06] -m, --t\_mult <t\_mult> Multiply current restart period times this number given number of restarts. [default: 2.0] -bs, --batch\_size <batch\_size> Batch size. [default: 50] -vp, --val pkl <val pkl> Validation Set Methylation Array Location. [default: ./train\_val\_test\_sets/val\_methyl\_array.pkl] -w, --n\_workers <n\_workers> Number of workers. [default: 9] -v, --add\_validation\_set Evaluate validation set. -1, --loss\_reduction <loss\_reduction> Type of reduction on loss function. [default: sum]

-j, --job\_name < job\_name>

-hl, --hyperparameter\_log <hyperparameter\_log>

Embedding job name. [default: predict\_job]

-sft, --add softmax

Add softmax for predicting probability distributions. Experimental.

CSV file containing prior runs. [default: predictions/predict\_hyperparameters\_log.csv]

## 9.5 regression\_report

Generate regression report that gives concise results from regression tasks.

```
methylnet-predict regression_report [OPTIONS]
```

```
-r, --results_pickle <results_pickle>
   Results from training, validation, and testing. [default: predictions/results.p]
```

```
-o, --output_dir <output_dir>
    Output directory. [default: results/]
```

**TEN** 

## **METHYLNET-INTERPRET**

```
methylnet-interpret [OPTIONS] COMMAND [ARGS]...
```

## **Options**

#### --version

Show the version and exit.

## 10.1 bin\_regression\_shaps

Take aggregate individual scores from regression results, that normally are not categorized, and aggregate across new category created from continuous data.

```
methylnet-interpret bin_regression_shaps [OPTIONS]
```

## **Options**

- -s, --shapley\_data <shapley\_data>
  Pickle containing top CpGs. [default: ./interpretations/shapley\_explanations/shapley\_data.p]
- -t, --test\_pkl <test\_pkl>
   Pickle containing testing set. [default: ./train\_val\_test\_sets/test\_methyl\_array.pkl]
- -c, --col <col>
  Column to turn into bins. [default: age]
- -n, --n\_bins <n\_bins>
   Number of bins. [default: 10]
- -ot, --output\_test\_pkl <output\_test\_pkl>
   Binned shap pickle for further testing. [default: ./train\_val\_test\_sets/test\_methyl\_array\_shap\_binned.pkl]
- -os, --output\_shap\_pkl <output\_shap\_pkl>
   Pickle containing top CpGs, binned phenotype. [default: ./interpretations/shapley\_explanations/shapley\_binned.p]

## 10.2 extract\_methylation\_array

Subset and write methylation array using top SHAP cpgs.

```
methylnet-interpret extract_methylation_array [OPTIONS]
```

## **Options**

- -s, --shapley\_data <shapley\_data>
  Pickle containing top CpGs. [default: ./interpretations/shapley\_explanations/shapley\_data.p]
- -co, --classes\_only
  Only take top CpGs from each class. [default: False]
- -o, --output\_dir <output\_dir>
   Output directory for methylation array. [default: ./interpretations/shapley\_explanations/top\_cpgs\_extracted\_methylarr/]
- -t, --test\_pkl <test\_pkl>
   Pickle containing testing set. [default: ./train\_val\_test\_sets/test\_methyl\_array.pkl]
- -c, --col <col>
   Column to color for output csvs. [default: ]
- -g, --global\_vals
   Only take top CpGs globally. [default: False]
- -n, --n\_extract <n\_extract>
   Number cpgs to extract. [default: 1000]

## 10.3 grab\_lola\_db\_cache

Download core and extended LOLA databases for enrichment tests.

```
methylnet-interpret grab_lola_db_cache [OPTIONS]
```

#### **Options**

```
-o, --output_dir <output_dir>
   Output directory for lola dbs. [default: ./lola_db/]
```

## 10.4 interpret\_biology

Interrogate CpGs with high SHAPley scores for individuals, classes, or overall, for enrichments, genes, GO, KEGG, LOLA, overlap with popular cpg sets, GSEA, find nearby cpgs to top.

```
methylnet-interpret interpret_biology [OPTIONS]
```

- -a, --all\_cpgs\_pickle <all\_cpgs\_pickle>
   List of all cpgs used in shapley analysis. [default: /interpretations/shapley\_explanations/all\_cpgs.p]
- -s, --shapley\_data\_list <shapley\_data\_list>
  Pickle containing top CpGs. [default: ./interpretations/shapley explanations/shapley data.p]

```
-o, --output dir <output dir>
     Output directory for interpretations. [default: ./interpretations/biological explanations/]
-w, --analysis <analysis>
     Choose biological analysis. [default: GO]
-n, --n workers <n workers>
     Number workers. [default: 8]
-1, --lola db <lola db>
     LOLA region db. [default: ./lola_db/core/nm/t1/resources/regions/LOLACore/hg19/]
-i, --individuals <individuals>
     Individuals to evaluate. [default: ]
-c, --classes <classes>
     Classes to evaluate. [default: ]
-m, --max_gap <max_gap>
     Genomic distance to search for nearby CpGs than found top cpgs shapleys. [default: 1000]
-ci, --class_intersect
     Compute class shapleys by intersection of individuals. [default: False]
-lo, --length_output <length_output>
     Number enriched terms to print. [default: 20]
-ss, --set subtraction
     Only consider CpGs relevant to particular class. [default: False]
-int, --intersection
     CpGs common to all classes. [default: False]
-col, --collections <collections>
     Lola collections. [default: ]
-gsea, --gsea_analyses <gsea_analyses>
     Gene set enrichment analysis to choose from, if chosen will override other analysis options. http://software.
     broadinstitute.org/gsea/msigdb/collections.jsp [default: ]
-gsp, --gsea_pickle <gsea_pickle>
     Gene set enrichment analysis to choose from. [default: ./data/gsea collections.p]
-d, --depletion
     Run depletion LOLA analysis instead of enrichment. [default: False]
-ov, --overlap_test
     Run overlap test with IDOL library instead of all other tests. [default: False]
-cgs, --cpg_set <cpg_set>
     Test for clock or IDOL enrichments. [default: IDOL]
-g, --top_global
     Look at global top cpgs, overwrites classes and individuals. [default: False]
-ex, --extract_library
```

## 10.5 list classes

List classes/multioutput regression cell-types that have SHAPley data to be interrogated.

10.5. list classes 51

Extract a library of cpgs to subset future array for inspection or prediction tests. [default: False]

```
methylnet-interpret list_classes [OPTIONS]
```

## **Options**

-s, --shapley\_data <shapley\_data>
Pickle containing top CpGs. [default: ./interpretations/shapley\_explanations/shapley\_data.p]

## 10.6 list\_individuals

List individuals that have ShapleyData. Not all made the cut from the test dataset.

```
methylnet-interpret list_individuals [OPTIONS]
```

## **Options**

-s, --shapley\_data <shapley\_data>
Pickle containing top CpGs. [default: ./interpretations/shapley\_explanations/shapley\_data.p]

## 10.7 order\_results\_by\_col

Order results that produces some CSV by phenotype column, alphabetical ordering to show maybe that results group together. Plot using pymethyl-visualize.

```
methylnet-interpret order_results_by_col [OPTIONS]
```

## **Options**

- -i, --input\_csv <input\_csv>
   Output directory for cpg jaccard\_stats. [default: ./interpretations/shapley\_explanations/top\_cpgs\_jaccard/all\_jaccard.csv]
- -t, --test\_pkl <test\_pkl>
   Pickle containing testing set. [default: ./train\_val\_test\_sets/test\_methyl\_array.pkl]
- -c, --col <col>
   Column to sort on. [default: disease]
- -o, --output\_csv <output\_csv>
  Output directory for cpg jaccard\_stats. [default: ./interpretations/shapley\_explanations/top\_cpgs\_jaccard/all\_jaccard\_sorted.csv]
- -sym, --symmetric
  Is symmetric? [default: False]

## 10.8 plot\_all\_lola\_outputs

Iterate through all LOLA csy results and plot forest plots of all enrichment scores.

```
methylnet-interpret plot_all_lola_outputs [OPTIONS]
```

## **Options**

```
-1, --head_lola_dir <head_lola_dir>
    Location of lola output csvs. [default: interpretations/biological_explanations/]
-o, --plot_output_dir <plot_output_dir>
    Output directory for interpretations. [default: ./interpretations/biological_explanations/lola_plots/]
-d, --description_col <description_col>
    Description column for categorical variables [default: description]
-c, --cell_types <cell_types>
    Cell types. [default: ]
-ov, --overwrite
    Overwrite existing plots. [default: False]
```

## 10.9 plot\_lola\_output

Plot LOLA results via forest plot for one csv file.

```
methylnet-interpret plot_lola_output [OPTIONS]
```

## **Options**

```
-1, --lola_csv <lola_csv>
    Location of lola output csv. [default: ]
-o, --plot_output_dir <plot_output_dir>
    Output directory for interpretations. [default: ./interpretations/biological_explanations/lola_plots/]
-d, --description_col <description_col>
    Description column for categorical variables [default: description]
-c, --cell_types <cell_types>
    Cell types. [default: ]
```

## 10.10 plot\_top\_cpgs

Plot the top cpgs locations in the genome using circos.

```
methylnet-interpret plot_top_cpgs [OPTIONS]
```

```
-s, --shapley_data <shapley_data>
    Pickle containing top CpGs. [default: ./interpretations/shapley_explanations/shapley_data.p]
-o, --output_dir <output_dir>
    Output directory for output plots. [default: ./interpretations/output_plots/]
-i, --individuals <individuals>
    Individuals to evaluate. [default: ]
```

-c, --classes <classes>
 Classes to evaluate. [default: ]

## 10.11 produce\_shapley\_data

Explanations (coefficients for CpGs) for every individual prediction. Produce SHAPley scores for each CpG for each individualized prediction, and then aggregate them across coarser classes. Store CpGs with top SHAPley scores as well for quick access. Store in Shapley data object.

methylnet-interpret produce\_shapley\_data [OPTIONS]

## **Options**

-i, --train pkl <train pkl>

Input database for beta and phenotype data. Use ./predictions/vae\_mlp\_methyl\_arr.pkl or ./embed-dings/vae\_mlp\_methyl\_arr.pkl for vae interpretations. [default: ./train\_val\_test\_sets/train\_methyl\_array.pkl]

-v, --val\_pkl <val\_pkl>

Val database for beta and phenotype data. Use ./predictions/vae\_mlp\_methyl\_arr.pkl or ./embed-dings/vae mlp methyl arr.pkl for vae interpretations. [default: ./train val test sets/val methyl array.pkl]

-t, --test\_pkl <test\_pkl>

Pickle containing testing set. [default: ./train\_val\_test\_sets/test\_methyl\_array.pkl]

-m, --model\_pickle <model\_pickle>

Pytorch model containing forward\_predict method. [default: ./predictions/output\_model.p]

-w, --n workers <n workers>

Number of workers. [default: 9]

-bs, --batch\_size <batch\_size>

Batch size. [default: 512]

-c, --cuda

Use GPUs.

-ns, --n samples <n samples>

Number of samples for SHAP output. [default: 200]

-nf, --n\_top\_features <n\_top\_features>

Top features to select for shap outputs. If feature selection, instead use this number to choose number features that make up top global score. [default: 20]

-o, --output\_dir <output\_dir>

Output directory for interpretations. [default: ./interpretations/shapley\_explanations/]

-mth, --method <method>

Explainer type. [default: kernel]

-ssbs, --shap\_sample\_batch\_size <shap\_sample\_batch\_size>

Break up shapley computations into batches. Set to 0 for only 1 batch. [default: 0]

-r, --n\_random\_representative <n\_random\_representative>

Number of representative samples to choose for background selection. Is this number for regression models, or this number per class for classification problems. [default: 0]

-col, --interest\_col <interest\_col>

Column of interest for sample selection for explainer training. [default: disease]

-rt, --n\_random\_representative\_test <n\_random\_representative\_test>

Number of representative samples to choose for test set. Is this number for regression models, or this number per class for classification problems. [default: 0]

-e, --categorical\_encoder <categorical\_encoder>

One hot encoder if categorical model. If path exists, then return top positive controbutions per samples of that class. Encoded values must be of sample class as interest\_col. [default: ./predictions/one\_hot\_encoder.p]

-cc, --cross\_class

Find importance of features for prediction across classes.

-fs, --feature\_selection

Perform feature selection using top global SHAP scores. [default: False]

-top, --top\_outputs <top\_outputs>

Get shapley values for fewer outputs if feature selection. [default: 0]

-vae, --vae\_interpret

Use model to get shapley values for VAE latent dimensions, only works if proper datasets are set. [default: False]

-cl, --pred\_class <pred\_class>

Prediction class top cpgs. [default: ]

-r, --results\_csv <results\_csv>

Remove all misclassifications. [default: ./predictions/results.csv]

-ind, --individual <individual>

One individual top cpgs. [default: ]

-rc, --residual\_cutoff <residual\_cutoff>

Activate for regression interpretations. Standard deviation in residuals before removing. [default: 0.0]

-cn, --cell\_names <cell\_names>

Multioutput names for multi-output regression. [default: ]

## 10.12 reduce\_top\_cpgs

Reduce set of top cpgs.

```
methylnet-interpret reduce_top_cpgs [OPTIONS]
```

## **Options**

-s, --shapley\_data <shapley\_data>

Pickle containing top CpGs. [default: ./interpretations/shapley\_explanations/shapley\_data,p]

-nf, --n\_top\_features <n\_top\_features>

Top features to select for shap outputs. [default: 500]

-o, --output\_pkl <output\_pkl>

Pickle containing top CpGs, reduced number. [default: ./interpretations/shapley\_explanations/shapley\_reduced\_data.p]

## 10.13 regenerate\_top\_cpgs

Increase size of Top CpGs using the original SHAP scores.

```
methylnet-interpret regenerate_top_cpgs [OPTIONS]
```

## **Options**

- -s, --shapley\_data <shapley\_data>
  Pickle containing top CpGs. [default: ./interpretations/shapley\_explanations/shapley\_data.p]
- -nf, --n\_top\_features <n\_top\_features>
   Top features to select for shap outputs. [default: 500]
- -o, --output\_pkl <output\_pkl>
   Pickle containing top CpGs, reduced number. [default: ./interpretations/shapley\_explanations/shapley\_reduced\_data.p]
- -a, --abs\_val
   Top CpGs found using absolute value.
- -n, --neg\_valReturn top CpGs that are making negative contributions.

## 10.14 shapley\_jaccard

Plot Shapley Jaccard Similarity Matrix to demonstrate sharing of Top CpGs between classes/groupings of individuals.

```
methylnet-interpret shapley_jaccard [OPTIONS]
```

- -s, --shapley\_data <shapley\_data>
  Pickle containing top CpGs. [default: ./interpretations/shapley\_explanations/shapley\_data.p]
- -c, --class\_names <class\_names>
   Class names. [default: ]
- -o, --output\_dir <output\_dir>
   Output directory for cpg jaccard\_stats. [default: ./interpretations/shapley\_explanations/top\_cpgs\_jaccard/]
- -ov, --overall
  Output overall similarity. [default: False]
- -i, --include\_individuals
  Output individuals. [default: False]
- -co, --cooccurrence
  Output cooccurrence instead jaccard. [default: False]
- -opt, --optimize\_n\_cpgs
  Search for number of top CpGs to use. [default: False]

## 10.15 split\_hyper\_hypo\_methylation

Split SHAPleyData object by methylation type (needs to change; but low methylation is 0.5 beta value and below) and output to file.

```
methylnet-interpret split_hyper_hypo_methylation [OPTIONS]
```

## **Options**

- -s, --shapley\_data <shapley\_data>
   Pickle containing top CpGs. [default: ./interpretations/shapley\_explanations/shapley\_data.p]
- -o, --output\_dir <output\_dir>
   Output directory for hypo/hyper shap data. [default: ./interpretations/shapley\_explanations/shapley\_data\_by\_methylation/]
- -t, --test\_pkl <test\_pkl>
   Pickle containing testing set. [default: ./train\_val\_test\_sets/test\_methyl\_array.pkl]

## 10.16 view\_methylation\_top\_cpgs

Write the Top CpGs for each class/individuals to file along with methylation information.

```
methylnet-interpret view_methylation_top_cpgs [OPTIONS]
```

- -s, --shapley\_data <shapley\_data>
  Pickle containing top CpGs. [default: ./interpretations/shapley\_explanations/shapley\_data.p]
- -o, --output\_dir <output\_dir>
   Output directory for cpg methylation shap. [default: ./interpretations/shapley\_explanations/top\_cpgs\_methylation/]
- -t, --test\_pkl <test\_pkl>
   Pickle containing testing set. [default: ./train\_val\_test\_sets/test\_methyl\_array.pkl]

## **ELEVEN**

## METHYLNET-VISUALIZE

```
methylnet-visualize [OPTIONS] COMMAND [ARGS]...
```

## **Options**

#### --version

Show the version and exit.

## 11.1 plot\_roc\_curve

Plot ROC Curves from classification tasks; requires classification\_report be run first.

```
methylnet-visualize plot_roc_curve [OPTIONS]
```

## **Options**

```
-r, --roc_curve_csv <roc_curve_csv>
    Weighted ROC Curve. [default: results/Weighted_ROC.csv]
```

```
-o, --outputfilename <outputfilename>
   Output image. [default: results/roc_curve.png]
```

## 11.2 plot\_training\_curve

Plot training curves as output from either the VAE training or VAE MLP.

```
methylnet-visualize plot_training_curve [OPTIONS]
```

- -t, --training\_curve\_file <training\_curve\_file>
   Training and validation loss and learning curves. [default: predictions/training\_val\_curve.p]
- -o, --outputfilename <outputfilename>
   Output image. [default: results/training\_curve.png]

- -vae, --vae\_train
   Plot VAE Training curves.
- -thr, --threshold <threshold>
   Loss values get rid of greater than this.

## **TWELVE**

## **METHYLNET-TORQUE**

```
methylnet-torque [OPTIONS] COMMAND [ARGS]...
```

## **Options**

#### --version

Show the version and exit.

## 12.1 run\_torque\_job

## Run torque job.

```
methylnet-torque run_torque_job [OPTIONS]
```

## **Options**

- -c, --command <command>
   Command to execute through torque. [default: ]
- -gpu, --use\_gpu

Specify whether to use GPUs. [default: False]

-a, --additions <additions>

Additional commands to add, one liner for now. [default: ]

-q, --queue <queue>

Queue for torque submission, gpuq also a good one if using GPUs. [default: default]

-t, --time <time>

Walltime in hours for job. [default: 1]

-n, --ngpu <ngpu>

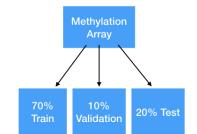
Number of gpus to request. [default: 0]

## **THIRTEEN**

## **EXAMPLE USAGE**

# **Pipeline**

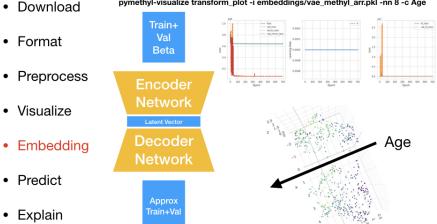
- Download
- Format
- Preprocess
- Visualize
- Embedding
- Predict
- Explain



pymethyl-utils train\_test\_val\_split -tp .8 -vp .125

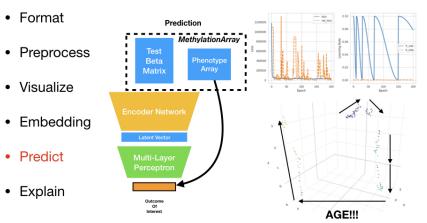
# **Pipeline**

python embedding.py launch\_hyperparameter\_scan -sc Age -t -mc 0.84 -b 1. -g -j 20 python embedding.py launch\_hyperparameter\_scan -sc Age -t -g -n 1 -b 1. pymethyl-visualize transform\_plot -i embeddings/vae\_methyl\_arr.pkl -nn 8 -c Age



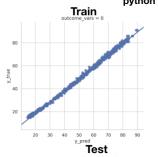
# **Pipeline**

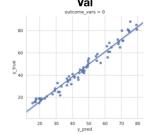
python predictions.py launch\_hyperparameter\_scan -ic Age -t -mc 0.84 -g -j 200
 Download python predictions.py launch\_hyperparameter\_scan -ic Age -t -g -n 1 pymethyl-visualize transform\_plot -i predictions/vae\_mlp\_methyl\_arr.pkl -nn 8 -c Age



# Pipeline python predictions.py regression\_report Val

- Download
- Format
- Preprocess
- Visualize
- Embedding
- Predict
- Explain



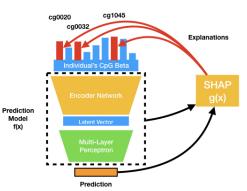


R2 about 96%!!! Mean Absolute Error 3.5 Years

# **Pipeline**

python model\_interpretability.py produce\_shapley\_data -mth gradient -ssbs 30 -ns 300 -bs 100 -rc 4. -r 0 -rt 0 -cn Age -nf 4000 -c

- Download
- Format
- Preprocess
- Visualize
- Embedding
- Predict
- Explain



# **Pipeline**

python model\_interpretability.py split\_hyper\_hypo\_methylation

python model\_interpretability.py bin\_regression\_shaps -c Age

python model\_interpretability.py shapley\_jaccard -c all -s ./interpretations/ shapley\_explanations/shapley\_binned.p -o ./interpretations/ shapley\_explanations/top\_cpgs\_jaccard/ -ov

pymethyl-visualize plot\_heatmap -m similarity -fs .7 -l ./interpretations/ shapley\_explanations/top\_cpgs\_jaccard/all\_jaccard.sorted.csv -o ./ interpretations/shapley\_explanationtylop\_cpgs\_jaccard/all\_jaccard.png -

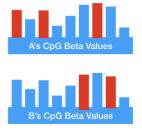
Visualize

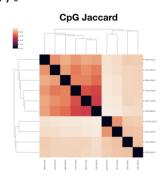
Preprocess

Download

Format

- Embedding
- Predict
- Explain



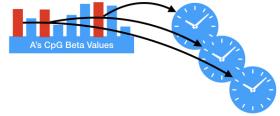


## **Pipeline**

- Download
- Format
- Preprocess
- Visualize
- Embedding
- Predict
- Explain

python model\_interpretability.py interpret\_biology -ov -c all -s interpretations/shapley\_explanations/shapley\_binned.p -cgs clock

(54.0,62.0] top cpgs overlap with 9.63% of clock cpgs (70.0,78.0] top cpgs overlap with 11.33% of clock cpgs (62.0,70.0] top cpgs overlap with 12.46% of clock cpgs (22.0,30.0] top cpgs overlap with 0.57% of clock cpgs (78.0,86.0] top cpgs overlap with 10.2% of clock cpgs (38.0,46.0] top cpgs overlap with 1.98% of clock cpgs (13.92,22.0] top cpgs overlap with 0.85% of clock cpgs (30.0,38.0) top cpgs overlap with 1.42% of clock cpgs (46.0,54.0] top cpgs overlap with 7.93% of clock cpgs (86.0,94.0] top cpgs overlap with 8.22% of clock cpgs



# **Pipeline**

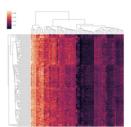
## **Horvath CpGs**

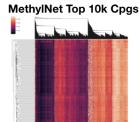
- Download
- Format
   pymethyl-utils subset\_array -i train\_val\_test\_sets/
  test\_methyl\_array.pkl -c\_/interpretations/biological\_explanations/
  cpg\_library.pkl

pymethyl-utils pkl\_to\_csv -i subset/methyl\_array.pkl -o subset/

- Preprocess pymethyl-visualize plot\_heatmap -fs .7 -i ./subset/beta.csv -o ./
   subset/beta.png -c &
- Visualize
- Embedding
- Predict
- Explain

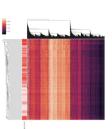
python model\_interpretability.py extract\_methylation\_array -s interpretations/shapley\_explanations/ shapley\_binned.p -c

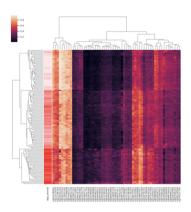




# **Pipeline**

- Download
- Format
- Preprocess
- Visualize
- Embedding
- Predict
- Explain





(13.92,22.0) top cpgs overlap with 0.0% of hannum cpgs (22.0,30.0) top cpgs overlap with 0.0% of hannum cpgs (30.0,38.0) top cpgs overlap with 1.45% of hannum cpgs (38.0,46.0) top cpgs overlap with 4.35% of hannum cpgs (38.0,46.0) top cpgs overlap with 4.35% of hannum cpgs (62.0,76.0) top cpgs overlap with 78.26% of hannum cpgs (62.0,76.0) top cpgs overlap with 78.26% of hannum cpgs (70.0,78.0) top cpgs overlap with 82.61% of hannum cpgs (78.0,86.0) top cpgs overlap with 82.61% of hannum cpgs (78.0,86.0) top cpgs overlap with 82.61% of hannum cpgs (36.0,94.0) top cpgs overlap with 82.62% of hannum cpgs

#### **CHAPTER**

# **FOURTEEN**

## **INDICES AND TABLES**

- genindex
- modindex
- search

### **PYTHON MODULE INDEX**

#### m

methylnet.datasets,2
methylnet.hyperparameter\_scans,6
methylnet.interpretation\_classes,10
methylnet.models,21
methylnet.plotter,34
methylnet.schedulers,35
methylnet.torque\_jobs,8

72 Python Module Index

## **INDEX**

| Symbols   | command line option, 45  |
|---|--|
| -version  | methylnet-predict-make_prediction                                    |
| methylnet-embed command line  | command line option, 46  |
| option, 39  | c, -cell_types <cell_types></cell_types>                             |
| methylnet-interpret command line                                    | <pre>methylnet-interpret-plot_all_lola_outputs</pre>                 |
| option, 49  | command line option, 53  |
| methylnet-predict command line                                      | <pre>methylnet-interpret-plot_lola_output</pre>                      |
| option,43   | command line option, 53  |
| methylnet-torque command line                                       | c, -class_names <class_names></class_names>                          |
| option, 61  | methylnet-interpret-shapley_jaccard                                  |
| methylnet-visualize command line                                    | command line option, 56  |
| option, 59  | c, -classes <classes></classes>                                      |
| -a, -abs_val  | methylnet-interpret_biology  |
| methylnet-interpret-regenerate_top_cpg                              | command line option, 51  |
| command line option, 56   | methylnet-interpret-plot_top_cpgs                                    |
| -a, -additional_command   | command line option, 53  |
| <additional_command></additional_command>                           | c, -col <col/>   |
| <pre>methylnet-embed-launch_hyperparameter_</pre>                   | scmethylnet-interpret-bin_regression_shaps                           |
| command line ontion 40  | command line option, 49  |
| <pre>methylnet-predict-launch_hyperparamete</pre>                   | r_methylnet-interpret-extract_methylation_array                      |
| command line option,44  | command line option, 30  |
| -a, -additions <additions></additions>                              | methylnet-interpret-order_results_by_col                             |
| <pre>methylnet-torque-run_torque_job</pre>                          | command line option, 52  |
| Command Time Operon, or   | c, -command <command/>   |
| <pre>-a, -all_cpgs_pickle <all_cpgs_pickle></all_cpgs_pickle></pre> | <pre>methylnet-torque-run_torque_job   command line option, 61</pre> |
| methylnet-interpret-interpret_biology                               | c, -crossover_p <crossover_p></crossover_p>                          |
| Command Time operon, 50   | methylnet-embed-launch_hyperparameter_scan                           |
| -b, -beta <beta></beta>   | command line option, 40  |
| methylnet-embed-perform_embedding                                   | methylnet-predict-launch_hyperparameter_scan                         |
| command line option, 41   | command line option, 44  |
| -b, -set_beta <set_beta></set_beta>                                 |  |
| methylnet-embed-launch_hyperparameter_                              | Scan methylnet-embed-perform_embedding                               |
| command line option, 40   | command line option, 40  |
| -bce, -bce_loss   | methylnet-interpret-produce_shapley_data                             |
| methylnet-embed-perform_embedding                                   | command line option, 54  |
| command line option, 41   | methylnet-predict-make_new_predictions                               |
| -bs, -batch_size <batch_size></batch_size>                          | command line option, 45  |
| methylnet-embed-perform_embedding                                   | methylnet-predict-make_prediction                                    |
| command line option, 41   | 1 7 1 45   |
| methylnet-interpret-produce_shapley_da<br>command line option,54    | c, -job_chunk_size <job_chunk_size></job_chunk_size>                 |
| methylnet-predict-make_new_predictions                              |  |
| mechyrner-breater-make_new_breaterions                              | _ 11 - 11 - 1 - 1 - 1 - 1  |

```
command line option, 39
                                            methylnet-predict-make_prediction
   methylnet-predict-launch_hyperparameter_scamcommand line option, 46
      command line option, 44
                                        -dp, -dropout_p <dropout_p>
-cat, -categorical
                                            methylnet-predict-make_prediction
   methylnet-predict-launch_hyperparameter_scamcommand line option, 46
      command line option, 44
                                        -e, -categorical encoder
   methylnet-predict-make_new_predictions
                                               <categorical encoder>
      command line option, 45
                                            methylnet-interpret-produce_shapley_data
   methylnet-predict-make_prediction
                                               command line option, 55
      command line option, 45
                                            methylnet-predict-classification_report
-cc, -cross_class
                                               command line option, 43
   methylnet-interpret-produce_shapley_datamethylnet-predict-make_new_predictions
      command line option, 55
                                               command line option, 45
                                        -e, -n_epochs <n_epochs>
-cqs, -cpq_set <cpq_set>
   methylnet-interpret-interpret_biology methylnet-embed-perform_embedding
      command line option, 51
                                                command line option, 40
-ci, -class_intersect
                                            methylnet-predict-make_prediction
   methylnet-interpret-interpret_biology
                                               command line option, 46
      command line option, 51
                                        -eta, -eta_min <eta_min>
-cl, -pred_class <pred_class>
                                            methylnet-embed-perform embedding
   methylnet-interpret-produce_shapley_data
                                               command line option, 41
      command line option, 55
                                            methylnet-predict-make_prediction
-cn, -cell_names <cell_names>
                                               command line option, 46
   methylnet-interpret-produce_shapley_datx, -extract_library
                                            methylnet-interpret-interpret_biology
      command line option, 55
-co, -classes_only
                                               command line option, 51
   methylnet-interpret-extract_methylatiofs_arrfeqature_selection
      command line option, 50
                                            methylnet-interpret-produce_shapley_data
                                               command line option, 55
-co, -cooccurrence
   methylnet-interpret-shapley_jaccard -g, -generate_input
      command line option, 56
                                            methylnet-embed-launch_hyperparameter_scan
-col, -collections <collections>
                                               command line option, 39
   methylnet-interpret-interpret_biology
                                           methylnet-predict-launch_hyperparameter_scan
      command line option, 51
                                               command line option, 44
-col, -interest col <interest col>
                                        -q, -qlobal_vals
   methylnet-interpret-produce_shapley_datamethylnet-interpret-extract_methylation_array
      command line option, 54
                                               command line option, 50
-conv, -convolutional
                                         -g, -top_global
   methylnet-embed-perform_embedding
                                            methylnet-interpret-interpret_biology
      command line option, 41
                                               command line option, 51
-d, -decay <decay>
                                         -qn, -qpu node <qpu node>
   methylnet-embed-perform_embedding
                                            methylnet-embed-launch_hyperparameter_scan
      command line option, 41
                                                command line option, 39
   methylnet-predict-make_prediction
                                            methylnet-predict-launch_hyperparameter_scan
      command line option, 46
                                               command line option, 44
-d, -depletion
                                         -qpu, -qpu <qpu>
   methylnet-interpret_biology methylnet-embed-launch_hyperparameter_scan
      command line option, 51
                                               command line option, 39
-d, -description_col <description_col>
                                          methylnet-predict-launch_hyperparameter_scan
   methylnet-interpret-plot_all_lola_outputs command line option, 44
      command line option, 53
                                         -gpu, -use_gpu
   methylnet-interpret-plot_lola_output
                                           methylnet-torque-run_torque_job
      command line option, 53
                                               command line option, 61
-do, -disease_only
                                        -qsea, -qsea analyses <qsea analyses>
```

```
command line option, 45
   methylnet-interpret-interpret_biology
       command line option, 51
                                         -ic, -interest_cols <interest_cols>
                                            methylnet-predict-launch_hyperparameter_scan
-qsp, -qsea pickle <qsea pickle>
   methylnet-interpret-interpret_biology
                                                command line option, 44
      command line option, 51
                                            methylnet-predict-make_new_predictions
-hcsv, -hyperparameter input csv
                                                command line option, 45
       <hyperparameter input csv>
                                            methylnet-predict-make prediction
   methylnet-embed-launch_hyperparameter_scan command line option, 45
       command line option, 39
                                         -ind, -individual <individual>
   methylnet-predict-launch_hyperparameter_methylnet-interpret-produce_shapley_data
       command line option, 43
                                                command line option, 55
                                         -int, -intersection
-hl, -hyperparameter_log
       <hyperparameter_log>
                                            methylnet-interpret-interpret_biology
                                                command line option, 51
   methylnet-embed-perform_embedding
       command line option, 41
                                         -j, -job_name <job_name>
   methylnet-predict-make_prediction
                                            methylnet-embed-perform_embedding
       command line option, 46
                                                command line option, 41
                                            methylnet-predict-make prediction
-hl, -hyperparameter output log
       <hyperparameter_output_log>
                                                command line option, 46
   methylnet-embed-launch_hyperparameter_js_cam_jobs <n_jobs>
       command line option, 39
                                            methylnet-embed-launch_hyperparameter_scan
   methylnet-predict-launch_hyperparameter_scamcommand line option, 40
       command line option, 43
                                            methylnet-predict-launch_hyperparameter_scan
-hlt, -hidden layer encoder topology
                                                command line option, 44
       <hidden_layer_encoder_topology>
                                         -kl, -kl_warm_up <kl_warm_up>
   methylnet-embed-perform_embedding
                                            methylnet-embed-perform embedding
       command line option, 40
                                                command line option, 40
-hlt, -hidden_layer_topology
                                         -l, -head_lola_dir <head_lola_dir>
       <hidden_layer_topology>
                                            methylnet-interpret-plot_all_lola_outputs
   methylnet-predict-make_prediction
                                                command line option, 53
       command line option, 46
                                         -1, -lola_csv <lola_csv>
-hs, -height_kernel_sizes
                                            methylnet-interpret-plot_lola_output
       <height_kernel_sizes>
                                                command line option, 53
   methylnet-embed-perform_embedding
                                         -1, -lola_db <lola_db>
       command line option, 41
                                            methylnet-interpret-interpret_biology
-i, -include_individuals
                                                command line option, 51
   methylnet-interpret-shapley_jaccard -1, -loss_reduction <loss_reduction>
       command line option, 56
                                            methylnet-embed-perform_embedding
-i, -individuals <individuals>
                                                command line option, 41
   methylnet-interpret-interpret_biology
                                            methylnet-predict-make_prediction
       command line option, 51
                                                command line option, 46
   methylnet-interpret-plot_top_cpgs
                                         -lo, -length output <length output>
       command line option, 53
                                            methylnet-interpret-interpret biology
   methylnet-interpret-view_methylation_top_cpgsmmand line option, 51
                                         -lr, -learning_rate <learning_rate>
       command line option, 57
-i, -input_csv <input_csv>
                                            methylnet-embed-perform_embedding
   methylnet-interpret-order_results_by_col
                                               command line option, 40
       command line option, 52
                                         -lr_mlp, -learning_rate_mlp
-i, -train_pkl <train_pkl>
                                                <learning_rate_mlp>
   methylnet-embed-perform_embedding
                                            methylnet-predict-make_prediction
       command line option, 40
                                                command line option, 46
   methylnet-interpret-produce_shapley_dalta_vae, -learning_rate_vae
       command line option, 54
                                                <learning_rate_vae>
   methylnet-predict-make prediction
                                            methylnet-predict-make prediction
```

```
command line option, 46
                                            methylnet-embed-launch_hyperparameter_scan
-m, -max_gap <max_gap>
                                                command line option, 39
                                            methylnet-predict-launch_hyperparameter_scan
   methylnet-interpret-interpret_biology
      command line option, 51
                                                command line option, 44
                                         -ns, -n_samples <n_samples>
-m, -model_pickle <model_pickle>
   methylnet-interpret-produce_shapley_datamethylnet-interpret-produce_shapley_data
      command line option, 54
                                               command line option, 54
   methylnet-predict-make_new_predictionso, -output_csv <output_csv>
      command line option, 45
                                            methylnet-interpret-order_results_by_col
                                                command line option, 52
-m, -t_mult <t_mult>
   methylnet-embed-perform_embedding
                                         -o, -output_dir <output_dir>
      command line option, 41
                                            methylnet-embed-perform_embedding
   methylnet-predict-make_prediction
                                                command line option, 40
      command line option, 46
                                            methylnet-interpret-extract_methylation_array
-mc, -model_complexity_factor
                                               command line option, 50
      <model_complexity_factor>
                                            methylnet-interpret-grab_lola_db_cache
   methylnet-embed-launch_hyperparameter_scan command line option, 50
      command line option, 40
                                            methylnet-interpret-interpret_biology
   methylnet-predict-launch_hyperparameter_scamcommand line option, 50
      command line option, 44
                                            methylnet-interpret-plot_top_cpgs
-mth, -method <method>
                                                command line option, 53
   methylnet-interpret-produce_shapley_datamethylnet-interpret-produce_shapley_data
      command line option, 54
                                                command line option, 54
-n, -n bins <n bins>
                                            methylnet-interpret-shapley_jaccard
                                                command line option, 56
   methylnet-interpret-bin_regression_shaps
                                            methylnet-interpret-split_hyper_hypo_methylatio
      command line option, 49
-n, -n_extract <n_extract>
                                                command line option, 57
   methylnet-interpret-extract_methylation_methylnet-interpret-view_methylation_top_cpgs
      command line option, 50
                                               command line option, 57
-n, -n_jobs_relaunch <n_jobs_relaunch>
                                            methylnet-predict-classification_report
   methylnet-embed-launch_hyperparameter_scan command line option, 43
      command line option, 40
                                            methylnet-predict-make_new_predictions
   methylnet-predict-launch_hyperparameter_scamcommand line option, 45
      command line option, 44
                                           methylnet-predict-make_prediction
-n, -n_latent <n_latent>
                                                command line option, 45
   methylnet-embed-perform_embedding
                                            methylnet-predict-regression_report
      command line option, 40
                                               command line option, 47
-n, -n_workers <n_workers>
                                         -o, -output_pkl <output_pkl>
   methylnet-interpret-interpret_biology
                                           methylnet-interpret-reduce_top_cpgs
                                                command line option, 55
      command line option, 51
                                            methylnet-interpret-regenerate_top_cpgs
-n, -neg val
   methylnet-interpret-regenerate_top_cpgs
                                               command line option, 56
      command line option, 56
                                         -o, -outputfilename <outputfilename>
                                            methylnet-visualize-plot_roc_curve
-n, -ngpu <ngpu>
   methylnet-torque-run_torque_job
                                                command line option, 59
      command line option, 61
                                            methylnet-visualize-plot_training_curve
                                               command line option, 59
-nf, -n_top_features <n_top_features>
   methylnet-interpret-produce_shapley_data -plot_output_dir <plot_output_dir>
      command line option, 54
                                            methylnet-interpret-plot_all_lola_outputs
   methylnet-interpret-reduce_top_cpgs
                                                command line option, 53
      command line option, 55
                                            methylnet-interpret-plot_lola_output
   methylnet-interpret-regenerate_top_cpgs
                                             command line option, 53
      command line option, 56
                                         -opt, -optimize_n_cpgs
                                            methylnet-interpret-shapley_jaccard
-nh, -nohup
```

```
-s, -shapley_data <shapley_data>
       command line option, 56
-os, -output_shap_pkl
                                            methylnet-interpret-bin_regression_shaps
                                                command line option, 49
       <output shap pkl>
   methylnet-interpret-bin_regression_shapsmethylnet-interpret-extract_methylation_array
       command line option, 49
                                                command line option, 50
-ot, -output test pkl
                                            methylnet-interpret-list classes
       <output test pkl>
                                                command line option, 52
   methylnet-interpret-bin_regression_shapsmethylnet-interpret-list_individuals
       command line option, 49
                                                command line option, 52
                                            methylnet-interpret-plot_top_cpgs
-ov, -overall
   methylnet-interpret-shapley_jaccard
                                                command line option, 53
      command line option, 56
                                            methylnet-interpret-reduce_top_cpgs
-ov, -overlap_test
                                                command line option, 55
                                            methylnet-interpret-regenerate_top_cpgs
   methylnet-interpret-interpret_biology
       command line option, 51
                                                command line option, 56
-ov, -overwrite
                                            methylnet-interpret-shapley_jaccard
   methylnet-interpret-plot_all_lola_outputs
                                                command line option, 56
       command line option, 53
                                            methylnet-interpret-split_hyper_hypo_methylatio
-q, -queue <queue>
                                                command line option, 57
   methylnet-torque-run torque job
                                            methylnet-interpret-view_methylation_top_cpgs
      command line option, 61
                                                command line option, 57
-r, -n_random_representative
                                         -s, -shapley_data_list
       <n_random_representative>
                                                <shapley_data_list>
   methylnet-interpret-produce_shapley_datamethylnet-interpret-interpret_biology
       command line option, 54
                                                command line option, 50
-r, -reset_all
                                         -sc, -stratify_column
   methylnet-embed-launch_hyperparameter_scan <stratify_column>
       command line option, 39
                                            methylnet-embed-launch_hyperparameter_scan
   methylnet-predict-launch_hyperparameter_scamcommand line option, 39
                                            methylnet-embed-perform_embedding
      command line option, 44
-r, -results_csv <results_csv>
                                                command line option, 41
   methylnet-interpret-produce_shapley_datat, -add_softmax
       command line option, 55
                                            methylnet-predict-launch_hyperparameter_scan
-r, -results_pickle <results_pickle>
                                                command line option, 44
   methylnet-predict-classification_report methylnet-predict-make_prediction
       command line option, 43
                                                command line option, 46
   methylnet-predict-regression_report -ss, -set_subtraction
       command line option, 47
                                            methylnet-interpret-interpret_biology
-r, -roc_curve_csv <roc_curve_csv>
                                                command line option, 51
   methylnet-visualize-plot_roc_curve
                                         -ssbs, -shap_sample_batch_size
       command line option, 59
                                                <shap sample batch size>
-rc, -residual_cutoff
                                            methylnet-interpret-produce_shapley_data
                                                command line option, 54
       <residual cutoff>
   methylnet-interpret-produce_shapley_dastym, -symmetric
                                            methylnet-interpret-order_results_by_col
       command line option, 55
                                                command line option, 52
-rt, -n_random_representative_test
       <n_random_representative_test>
                                         -t, -t_max <t_max>
   methylnet-interpret-produce_shapley_datamethylnet-embed-perform_embedding
      command line option, 54
                                                command line option, 41
-s, -scheduler <scheduler>
                                            methylnet-predict-make_prediction
   methylnet-embed-perform_embedding
                                                command line option, 46
      command line option, 41
                                         -t, -test_pkl <test_pkl>
   methylnet-predict-make_prediction
                                            methylnet-interpret-bin_regression_shaps
       command line option, 46
                                                command line option, 49
```

```
methylnet-interpret-extract_methylation_arraxyommand line option, 59
       command line option, 50
                                           -vp, -val_pkl <val_pkl>
   methylnet-interpret-order_results_by_colmethylnet-embed-perform_embedding
       command line option, 52
                                                  command line option, 41
   methylnet-interpret-produce_shapley_datamethylnet-predict-make_prediction
       command line option, 54
                                                  command line option, 46
   methylnet-interpret-split_hyper_hypo_methydnatlivosis <analysis>
       command line option, 57
                                               methylnet-interpret-interpret biology
   methylnet-interpret-view_methylation_top_cpgcommand line option,51
       command line option, 57
                                           -w, -n_workers <n_workers>
-t, -time <time>
                                              methylnet-embed-perform_embedding
   methylnet-torque-run_torque_job
                                                  command line option, 41
       command line option, 61
                                              methylnet-interpret-produce_shapley_data
-t, -torque
                                                  command line option, 54
   methylnet-embed-launch_hyperparameter_scmethylnet-predict-make_new_predictions
       command line option, 39
                                                  command line option, 45
   methylnet-predict-launch_hyperparameter_methylnet-predict-make_prediction
       command line option, 44
                                                  command line option, 46
                                           -wd, -weight_decay <weight_decay>
-t, -training_curve_file
       <training curve file>
                                               methylnet-embed-perform embedding
   methylnet-visualize-plot_training_curve
                                                  command line option, 40
       command line option, 59
                                              methylnet-predict-make prediction
-thr, -threshold <threshold>
                                                  command line option, 46
   methylnet-visualize-plot_training_curwes, -width_kernel_sizes
       command line option, 60
                                                  <width kernel sizes>
-top, -top_outputs <top_outputs>
                                              methylnet-embed-perform_embedding
   methylnet-interpret-produce_shapley_data
                                                  command line option, 41
       command line option, 55
-tp, -test_pkl <test_pkl>
   methylnet-predict-make_new_predictionsdd_abs_value_classes()
                                                                               (methyl-
       command line option, 45
                                                  net.interpretation_classes.ShapleyDataExplorer
   methylnet-predict-make_prediction
                                                  method), 17
       command line option, 45
                                           add_bin_continuous_classes()
-v, -add_validation_set
                                                  net.interpretation_classes.ShapleyDataExplorer
   methylnet-embed-perform embedding
                                                  method), 17
       command line option, 41
                                           add_class()
                                                                               (methyl-
   methylnet-predict-make prediction
                                                  net.interpretation_classes.ShapleyData
       command line option, 46
                                                  method), 15
-v, -val_loss_column <val_loss_column>
                                           add_global_importance()
                                                                               (methyl-
   methylnet-embed-launch_hyperparameter_scan net.interpretation classes.ShapleyData
       command line option, 40
                                                  method), 15
   methylnet-predict-launch_hyperparametadd_scalhidation set()
                                                                               (methyl-
       command line option, 44
                                                  net.models.AutoEncoder method), 24
-v, -val_pkl <val_pkl>
                                           add validation set()
                                                                               (methyl-
   methylnet-interpret-produce_shapley_data
                                                  net.models.MLPFinetuneVAE method), 26
       command line option, 54
                                           assemble replace dict() (in module methyl-
-vae, -input_vae_pkl <input_vae_pkl>
                                                  net.torque_jobs), 9
   methylnet-predict-make_prediction
                                           assemble_run_torque() (in module methyl-
       command line option, 45
                                                  net.torque_jobs), 9
-vae, -vae_interpret
                                           AutoEncoder (class in methylnet.models), 23
   methylnet-interpret-produce_shapley_data
       command line option, 55
-vae, -vae_train
                                           BioInterpreter
                                                               (class
                                                                               methyl-
                                                                         in
   methylnet-visualize-plot training curve
                                                  net.interpretation_classes), 11
```

| build_explainer() (mether in the interpretation_classes.CpGExplainer   | method), 31  |
|--|--|
| method), 13  | from_explainer() (methyl-net.interpretation_classes.CpGExplainer                     |
|  | class method), 14  |
| cat () (in module methylnet.datasets), 5   | from_pickle() (methyl-   |
| <pre>classifier_assign_scores_to_shap_dat           (methylnet.interpretation_classes.CpGExpla           method), 13</pre> | T T 16   |
| coarse_scan() (in module meth  | $_{nvl_{-}}$ G   |
| net.hyperparameter_scans), 7   | generate() (methylnet.datasets.Transformer method),                                  |
| cooccurrence_fn() (in module meth  |  |
| $net.interpretation\_classes), 20$   | <pre>generate_topology() (in module methyl-</pre>                                    |
| CosineAnnealingWithRestartsLR (class   | in net.hyperparameter_scans), 8  |
| methylnet.schedulers), 37  | get_latent_z() (methylnet.models.TybaltTitusVAE                                      |
| CpGExplainer (class in meth  | nyl- method), 28 get_lr() (methylnet.schedulers.Scheduler method), 38                |
| net.interpretation_classes), 12  | get_methylation_dataset() (in module methyl-   |
| D  | net.datasets), 6   |
| decode() (methylnet.models.TybaltTitusVAE metho  | nd), get_nearby_cpg_shapleys() (methyl-<br>net.interpretation_classes.BioInterpreter |
| decode() (methylnet.models.VAE_MLP method), 31   | method), 11  |
| E  | <pre>gometh() (methylnet.interpretation_classes.BioInterpreter</pre>                 |
| encode() (methylnet.models.TybaltTitusVAE metho  | od), J   |
| 28   | jaccard_similarity() (in module methyl-  |
| extract_class() (meth  | 191-   |
| net.interpretation_classes.ShapleyDataExplomethod), 17   | jaccard_similarity_top_cpgs() (methyl-   |
| extract_individual() (meth   | not interpretation along as Charley Data Family and                                  |
| net.interpretation_classes.ShapleyDataExplo<br>method), 17   | * ath a d 17   |
| extract_methylation_array() (meth  | ıyl-   |
| net.interpretation_classes.ShapleyDataExpl   | orer limit_number_top_cpgs() (methyl-  |
| method), 17  | net.interpretation_classes.ShapleyDataExplorer<br>method), 18                        |
| F  | list_classes() (methyl-  |
| feature_select() (meth   | net.interpretation_classes.ShapleyDataExplorer<br>method), 18                        |
| net.interpretation_classes.CpGExplainer  | list_individuals() (methyl-  |
| method), 13  | nat interpretation classes Shapley Data Explorer                                     |
| find_top_jobs() (in module meth<br>net.hyperparameter_scans), 8  | method), 18  |
| fit () (methylnet.models.AutoEncoder method), 24   | 26 M   |
| <pre>fit() (methylnet.models.MLPFinetuneVAE method); fit_transform() (methylnet.models.AutoEnco</pre>                      | main manalintina filmatica () (in manalint   |
| method), 24  | methylnet.interpretation_classes), 20  |
| forward() (methylnet.models.TybaltTitusV   | ME MethylationDataSet (class in methylnet.datasets),                                 |
| method), 28  | 3  |
| forward() (methylnet.models.VAE_MLP method),   | an other land data a stal 1  |
| forward_embed() (methylnet.models.VAE_M  | LP methylnet.datasets), 4 methylnet-embed command line option                        |
| method), 31  | trongian 20  |
| forward_predict() (method) 28  | <pre>methylnet-embed-launch_hyperparameter_scar</pre>                                |
| net.models.TybaltTitusVAE method), 28  | command line option  |

```
-a, -additional_command
                                                 <stratify_column>,41
       <additional_command>,40
                                             -t, -t_max <t_max>,41
   -b, -set beta <set beta>,40
                                             -v, -add validation set, 41
   -c, -crossover_p <crossover_p>,40
                                             -vp, -val_pkl < val_pkl > 41
   -c, -job_chunk_size
                                             -w, -n workers <n workers>,41
       <job chunk size>, 39
                                             -wd, -weight decay <weight decay>,
   -g, -generate_input, 39
   -gn, -gpu_node <gpu_node>,39
                                             -ws, -width_kernel_sizes
   -gpu, -gpu <gpu>, 39
                                                 <width kernel sizes>,41
   -hcsv, -hyperparameter_input_csv
                                          methylnet-interpret command line
       <hyperparameter_input_csv>, 39
                                                 option
   -hl, -hyperparameter_output_log
                                             -version, 49
       <hyperparameter_output_log>, 39
                                          methylnet-interpret-bin_regression_shaps
   -j, -n_{jobs} < n_{jobs} >, 40
                                                 command line option
   -mc, -model_complexity_factor
                                             -c, -col < col >, 49
       <model_complexity_factor>,40
                                             -n, -n_bins < n_bins>, 49
   -n, -n_jobs_relaunch
                                             -os, -output_shap_pkl
       <n_jobs_relaunch>,40
                                                 <output shap pkl>,49
   -nh, -nohup, 39
                                             -ot, -output_test_pkl
   -r, -reset_all, 39
                                                 <output_test_pkl>, 49
   -sc, -stratify_column
                                             -s, -shapley_data <shapley_data>,49
       <stratify_column>, 39
                                             -t, -test_pkl <test_pkl>,49
   -t, -torque, 39
                                          methylnet-interpret-extract_methylation_array
   -v, -val_loss_column
                                                 command line option
       <val_loss_column>, 40
                                             -c, -col < col > 50
methylnet-embed-perform_embedding
                                             -co, -classes_only, 50
      command line option
                                             -g, -global_vals, 50
   -b, -beta <beta>,41
                                             -n, -n_extract <n_extract>,50
   -bce, -bce_loss, 41
                                             -o, -output_dir <output_dir>,50
   -bs, -batch_size <batch_size>,41
                                             -s, -shapley_data <shapley_data>,50
   -c, -cuda, 40
                                             -t, -test_pkl < test_pkl > ,50
   -conv, -convolutional, 41
                                          methylnet-interpret-grab_lola_db_cache
   -d, -decay <decay>,41
                                                 command line option
   -e, -n_epochs < n_epochs>, 40
                                             -o, -output_dir <output_dir>,50
   -eta, -eta_min <eta_min>,41
                                          methylnet-interpret-interpret_biology
   -hl, -hyperparameter_log
                                                command line option
       <hyperparameter log>,41
                                             -a, -all_cpgs_pickle
   -hlt, -hidden_layer_encoder_topology
                                                 <all_cpgs_pickle>,50
       <hidden_layer_encoder_topology>,
                                             -c, -classes <classes>,51
                                             -cgs, -cpg_set <cpg_set>,51
   -hs, -height kernel sizes
                                             -ci, -class intersect, 51
       <height_kernel_sizes>,41
                                             -col, -collections <collections>,51
   -i, -train_pkl <train_pkl>,40
                                             -d, -depletion, 51
   -j, -job_name <job_name>,41
                                             -ex, -extract_library, 51
   -kl, -kl_warm_up < kl_warm_up >, 40
                                             -g, -top_global, 51
   -1, -loss_reduction
                                             -qsea, -qsea_analyses
       <loss_reduction>,41
                                                 <gsea_analyses>,51
   -lr, -learning_rate
                                             -qsp, -qsea_pickle <qsea_pickle>,51
       <learning_rate>, 40
                                             -i, -individuals <individuals>,51
   -m, -t_mult <t_mult>, 41
                                             -int, -intersection, 51
   -n, -n_latent < n_latent>, 40
                                             -1, -lola_db <lola_db>,51
   -o, -output dir <output dir>,40
                                             -lo, -length output
   -s, -scheduler <scheduler>,41
                                                 <length_output>,51
   -sc, -stratify_column
                                             -m, -max_gap < max_gap > ,51
```

| -n, -n_workers <n_workers>,51</n_workers>              | -e, -categorical_encoder   |
|--|--|
| -o, -output_dir <output_dir>,50</output_dir>           | <pre><categorical_encoder>, 55</categorical_encoder></pre>                   |
| -ov, -overlap_test,51                                  | -fs, -feature_selection,55   |
| -s, -shapley_data_list                                 | -i, -train_pkl <train_pkl>,54</train_pkl>                                    |
| <pre><shapley_data_list>, 50</shapley_data_list></pre> | -ind, -individual <individual>,55</individual>                               |
| -ss, -set_subtraction, 51                              | -m, -model_pickle <model_pickle>, 54</model_pickle>                          |
| -w, -analysis <analysis>,51</analysis>                 | -mth, -method <method>, 54</method>  |
| methylnet-interpret-list_classes                       | -nf, -n_top_features   |
| command line option                                    | <n_top_features>,54</n_top_features>   |
| -s, -shapley_data <shapley_data>,52</shapley_data>     | -ns, -n_samples <n_samples>,54</n_samples>                                   |
| methylnet-interpret-list_individuals                   | -o, -output_dir <output_dir>,54</output_dir>                                 |
| command line option                                    | -r, -n_random_representative   |
| -s, -shapley_data <shapley_data>,52</shapley_data>     | <pre><n_random_representative>, 54</n_random_representative></pre>           |
| methylnet-interpret-order_results_by_col               | -r, -results_csv <results_csv>,55</results_csv>                              |
| command line option                                    | -rc, -residual_cutoff  |
| -c, -col <col/> ,52                                    | <pre><residual_cutoff>,55</residual_cutoff></pre>                            |
| -i, -input_csv <input_csv>, 52</input_csv>             |  |
|  | -rt, -n_random_representative_test   |
| -o, -output_csv <output_csv>,52</output_csv>           | <pre><n_random_representative_test>, 54</n_random_representative_test></pre> |
| -sym, -symmetric, 52                                   |  |
| -t, -test_pkl <test_pkl>,52</test_pkl>                 | -ssbs, -shap_sample_batch_size   |
| methylnet-interpret-plot_all_lola_outputs              | <pre><shap_sample_batch_size>,54</shap_sample_batch_size></pre>              |
| command line option                                    | -t, -test_pkl <test_pkl>,54</test_pkl>                                       |
| -c, -cell_types <cell_types>,53</cell_types>           | -top, -top_outputs <top_outputs>,55</top_outputs>                            |
| -d, -description_col                                   | -v, -val_pkl <val_pkl>,54</val_pkl>  |
| <pre><description_col>,53</description_col></pre>      | -vae, -vae_interpret,55  |
| -l, -head_lola_dir <head_lola_dir>,</head_lola_dir>    | -w, -n_workers <n_workers>,54</n_workers>                                    |
|  | thylnet-interpret-reduce_top_cpgs  |
| -o, -plot_output_dir                                   | command line option  |
| <pre><plot_output_dir>, 53</plot_output_dir></pre>     | -nf, -n_top_features   |
| -ov, -overwrite, 53                                    | <n_top_features>,55</n_top_features>   |
| methylnet-interpret-plot_lola_output                   | -o, -output_pkl <output_pkl>,55</output_pkl>                                 |
| command line option                                    | -s, -shapley_data <shapley_data>,55</shapley_data>                           |
|  | thylnet-interpret-regenerate_top_cpgs  |
| -d, -description_col                                   | command line option  |
| <description_col>,53</description_col>                 | -a, -abs_val,56  |
| -1, -lola_csv <lola_csv>,53</lola_csv>                 | -n, -neg_val,56  |
| -o, -plot_output_dir                                   | -nf, -n_top_features   |
| <plot_output_dir>,53</plot_output_dir>                 | <n_top_features>,56</n_top_features>   |
| methylnet-interpret-plot_top_cpgs                      | -o, -output_pkl <output_pkl>,56</output_pkl>                                 |
| command line option                                    | -s, -shapley_data <shapley_data>,56</shapley_data>                           |
|  | thylnet-interpret-shapley_jaccard  |
| -i, -individuals <individuals>,53</individuals>        | command line option  |
| -o, -output_dir <output_dir>,53</output_dir>           | -c, -class_names <class_names>,56</class_names>                              |
| -s, -shapley_data <shapley_data>,53</shapley_data>     | -co, -cooccurrence,56  |
| methylnet-interpret-produce_shapley_data               | -i, -include_individuals,56  |
| command line option                                    | -o, -output_dir <output_dir>,56</output_dir>                                 |
| -bs, -batch_size <batch_size>,54</batch_size>          | -opt, -optimize_n_cpgs,56  |
| -c, -cuda, 54  | -ov, -overall,56   |
| -cc, -cross_class,55                                   | -s, -shapley_data <shapley_data>,56</shapley_data>                           |
| -cl, -pred_class <pred_class>,55 me</pred_class>       | thylnet-interpret-split_hyper_hypo_methylation                               |
| -cn, -cell_names <cell_names>,55</cell_names>          | command line option  |
| -col, -interest_col <interest_col>,</interest_col>     | -o, -output_dir <output_dir>,57</output_dir>                                 |
| 54   | -s, -shapley_data <shapley_data>,57</shapley_data>                           |
|  | -t, -test_pkl <test_pkl>,57</test_pkl>                                       |
|  |  |

```
methylnet-interpret-view_methylation_top_cpgstp, -test_pkl <test_pkl>, 45
      command line option
                                             -w, -n_workers <n_workers>,45
   -i, -individuals <individuals>, 57
                                         methylnet-predict-make prediction
   -o, -output_dir <output_dir>,57
                                                command line option
   -s, -shapley_data <shapley_data>,57
                                             -bs, -batch_size <batch_size>,46
   -t, -test pkl <test pkl>,57
                                             -c, -cuda, 45
methylnet-predict command line option
                                             -cat, -categorical, 45
   -version, 43
                                             -d, -decay <decay>,46
methylnet-predict-classification_report
                                             -do, -disease_only,46
      command line option
                                             -dp, -dropout_p <dropout_p>, 46
   -e, -categorical_encoder
                                             -e, -n_epochs < n_epochs>, 46
      <categorical_encoder>, 43
                                             -eta_min <eta_min>,46
   -o, -output_dir <output_dir>, 43
                                             -hl, -hyperparameter_log
                                                 <hyperparameter_log>,46
   -r, -results_pickle
      <results_pickle>,43
                                             -hlt, -hidden_layer_topology
methylnet-predict-launch_hyperparameter_scan
                                                 <hidden_layer_topology>,46
      command line option
                                             -i, -train_pkl <train_pkl>,45
                                             -ic, -interest_cols
   -a, -additional command
      <additional_command>,44
                                                 <interest_cols>, 45
                                             -j, -job_name <job_name>,46
   -c, -crossover_p <crossover_p>,44
   -c, -job_chunk_size
                                             -1, -loss_reduction
      <job_chunk_size>,44
                                                 <loss reduction>,46
   -cat, -categorical, 44
                                             -lr_mlp, -learning_rate_mlp
                                                 <learning_rate_mlp>, 46
   -g, -generate_input,44
   -gn, -gpu_node <gpu_node>,44
                                             -lr_vae, -learning_rate_vae
   -gpu, -gpu <gpu>,44
                                                <learning_rate_vae>, 46
   -hcsv, -hyperparameter_input_csv
                                             -m, -t_mult <t_mult>, 46
      <hyperparameter_input_csv>, 43
                                             -o, -output_dir <output_dir>, 45
                                             -s, -scheduler <scheduler>,46
   -hl, -hyperparameter_output_log
      <hyperparameter_output_log>, 43
                                             -sft, -add_softmax, 46
   -ic, -interest_cols
                                             -t, -t_max < t_max>, 46
       <interest_cols>,44
                                             -tp, -test_pkl <test_pkl>,45
   -j, -n_{jobs} < n_{jobs} >, 44
                                             -v, -add_validation_set, 46
   -mc, -model_complexity_factor
                                             -vae, -input_vae_pkl
       <model_complexity_factor>,44
                                                <input_vae_pkl>, 45
   -n, -n_jobs_relaunch
                                             -vp, -val_pkl < val_pkl >, 46
      <n_jobs_relaunch>,44
                                             -w, -n workers <n workers>,46
   -nh, -nohup, 44
                                             -wd, -weight_decay <weight_decay>,
   -r, -reset_all, 44
                                                46
   -sft, -add_softmax, 44
                                         methylnet-predict-regression_report
   -t, -torque, 44
                                                 command line option
   -v, -val_loss_column
                                             -o, -output_dir <output_dir>, 47
      <val_loss_column>, 44
                                             -r, -results_pickle
methylnet-predict-make_new_predictions
                                                 <results_pickle>,47
      command line option
                                         methylnet-torque command line option
   -bs, -batch_size <batch_size>, 45
                                             -version, 61
   -c, -cuda, 45
                                         methylnet-torque-run_torque_job
   -cat, -categorical, 45
                                                command line option
   -e, -categorical_encoder
                                             -a, -additions <additions>,61
      <categorical_encoder>,45
                                             -c, -command <command>,61
   -ic, -interest_cols
                                             -gpu, -use_gpu, 61
      <interest_cols>, 45
                                             -n, -ngpu < ngpu > 61
   -m, -model_pickle <model_pickle>, 45
                                             -q, -queue <queue>,61
   -o, -output_dir <output_dir>,45
                                             -t, -time < time > , 61
```

| methylnet-visualize command line                          | return_cpg_sets() (methyl-                                  |
|---|---|
| option  | $net.interpretation\_classes. Shapley Data Explorer$        |
| -version, 59  | method), 19   |
| methylnet-visualize-plot_roc_curve                        | return_dataloader_construct() (in module                    |
| command line option                                       | $methylnet.interpretation\_classes), 21$                    |
| -o, -outputfilename                                       | return_global_importance_cpgs() (methyl-                    |
| <pre><outputfilename>, 59</outputfilename></pre>          | $net. interpretation\_classes. Shapley Data Explorer$       |
| -r, -roc_curve_csv <roc_curve_csv>,</roc_curve_csv>       | method), 19   |
| 59  | return_overlap_score() (methyl-                             |
| <pre>methylnet-visualize-plot_training_curve</pre>        | net.interpretation_classes.BioInterpreter                   |
| command line option                                       | method), 12   |
| -o, -outputfilename                                       | return_predict_function() (in module methyl-                |
| <pre><outputfilename>, 59</outputfilename></pre>          | net.interpretation_classes), 21                             |
| -t, -training_curve_file                                  | return_shap_values() (in module methyl-                     |
| <training_curve_file>,59</training_curve_file>            | net.interpretation_classes), 21                             |
| -thr, -threshold <threshold>,<math>60</math></threshold>  | <pre>return_shapley_data_by_methylation_status()</pre>      |
| -vae, -vae_train,59                                       | $(methylnet.interpretation\_classes. Shapley Data Explorer$ |
| methylnet.datasets(module),2                              | method), 19   |
| <pre>methylnet.hyperparameter_scans (module), 6</pre>     | return_shapley_predictions() (methyl-                       |
| <pre>methylnet.interpretation_classes (mod-</pre>         | $net.interpretation\_classes.CpGExplainer$                  |
| ule), 10  | method), 14   |
| methylnet.models(module), 21                              | return_shapley_scores() (methyl-                            |
| methylnet.plotter(module), 34                             | net.interpretation_classes.CpGExplainer                     |
| methylnet.schedulers (module), 35                         | method), 14   |
| methylnet.torque_jobs(module), 8                          | return_top_cpgs() (methyl-                                  |
| MLPFinetuneVAE (class in methylnet.models), 25            | net.interpretation_classes.ShapleyDataExplorer              |
| ·   | method), 19   |
| P   | run_lola() (methyl-   |
| Plot (class in methylnet.plotter), 35                     | net.interpretation_classes.BioInterpreter                   |
| plot_cpgs() (methyl-                                      | method), 12   |
| net.interpretation_classes.PlotCircos method),            | run_torque_job_() (in module methyl-                        |
| 15  | net.torque_jobs), 9   |
| <pre>plot_lola_output_() (in module methyl-</pre>         | 0   |
| net.interpretation_classes), 20                           | S   |
| PlotCircos (class in methyl-                              | <pre>sample_z() (methylnet.models.TybaltTitusVAE</pre>      |
| net.interpretation_classes), 14                           | method), 29   |
| Plotter (class in methylnet.plotter), 35                  | Scheduler (class in methylnet.schedulers), 37               |
| PlotTransformer (class in methylnet.plotter), 35          | ShapleyData (class in methyl-                               |
| <pre>predict() (methylnet.models.MLPFinetuneVAE</pre>     | net.interpretation_classes), 15                             |
| method), 26   | ShapleyDataExplorer (class in methyl-                       |
| <pre>project_vae() (in module methylnet.models), 31</pre> | net.interpretation_classes), 16                             |
| _   | stack() (in module methylnet.datasets), 6                   |
| R   | step () (methylnet.schedulers.Scheduler method), 38         |
| RawBetaArrayDataSet (class in methyl-                     | _   |
| net.datasets), 4  | T   |
| regenerate_individual_shap_values()                       | test_mlp() (in module methylnet.models), 32                 |
| (methylnet.interpretation_classes.ShapleyDataEx           |   |
| method), 18   | net.datasets.MethylationDataSet method),                    |
| regressor_assign_scores_to_shap_data()                    | 3   |
| (methylnet.interpretation_classes.CpGExplainer            |   |
| method), 14   | net.interpretation_classes.ShapleyData                      |
| return_binned_shapley_data() (methyl-                     | method), 16   |
| net.interpretation_classes.ShapleyDataExplorer            |   |
| method) 18  | net interpretation classes) 21                              |

```
toggle_latent_z() (methylnet.models.VAE_MLP
        method), 31
train_decoder_() (in module methylnet.models), 32
train_mlp() (in module methylnet.models), 32
train_vae() (in module methylnet.models), 33
transform()
                      (methylnet.models. AutoEncoder
        method), 24
Transformer (class in methylnet.datasets), 5
TybaltTitusVAE (class in methylnet.models), 26
V
vae_loss() (in module methylnet.models), 33
VAE_MLP (class in methylnet.models), 29
val_decoder_() (in module methylnet.models), 33
val_mlp() (in module methylnet.models), 34
val_vae() (in module methylnet.models), 34
view_methylation()
                                          (methyl-
        net.interpretation\_classes.ShapleyDataExplorer
        method), 19
```