phenotrex Documentation

Release 0.4.0

Lukas Lüftinger

Mar 12, 2020

CONTENTS:

1 phenotrex **1**

2 Installation **3**
2.1 Stable release 3
2.2 From sources 3

3 Usage **5**

4 phenotrex **7**
4.1 phenotrex package 7

5 Credits **13**
5.1 Development Lead 13
5.2 Contributors 13

6 History **15**

7 Indices and tables **17**

Python Module Index **19**

Index **21**

CHAPTER

ONE

PHENOTREX

Microbial Phenotype Prediction, re-implemented with Python 3.7 and scikit-learn

- Supported platforms: Linux, MacOS, Windows
- Free software: MIT license

CHAPTER

TWO

INSTALLATION

2.1 Stable release

To install pheno-trex, run this command in your terminal:

```
$ pip install phenotrex
```

This is the preferred method to install pheno-trex, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for pheno-trex can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/univieCUBE/PICA2
```

Or download the tarball:

```
$ curl  -OL https://github.com/univieCUBE/PICA2/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER

THREE

USAGE

To use pheno-trex in a project:

```
from phenotrex.io import ...  #  file I/O
from phenotrex.ml import ...  # classifiers and training/CV functionality
from phenotrex.util import ...  # plotting and util functions
```

CHAPTER

FOUR

PHENOTREX

4.1 phenotrex package

4.1.1 Subpackages

phenotrex.cli package

Submodules

phenotrex.cli.cccv module

phenotrex.cli.clf_opt module

phenotrex.cli.compute_genotype module

phenotrex.cli.cv module

phenotrex.cli.generic_func module

phenotrex.cli.generic_opt module

phenotrex.cli.get_weights module

phenotrex.cli.main module

phenotrex.cli.plot module

phenotrex.cli.predict module

phenotrex.cli.train module

Module contents

phenotrex.io package

Submodules

phenotrex.io.flat module

phenotrex.io.serialization module

Module contents

phenotrex.ml package

Subpackages

phenotrex.ml.clf package

Submodules

phenotrex.ml.clf.svm module

phenotrex.ml.clf.xgbm module

Module contents

Submodules

phenotrex.ml.cccv module

phenotrex.ml.feature_select module

phenotrex.ml.trex_classifier module

phenotrex.ml.vectorizer module

Module contents

phenotrex.structure package

Submodules

phenotrex.structure.records module

class phenotrex.structure.records.**GenotypeRecord**(*identifier: str, features: List[str]*)

Bases: object

Genomic features of a sample referenced by *identifier*.

features: List[str] = None

identifier: str = None

class phenotrex.structure.records.**GroupRecord**(*identifier: str, group_name: Optional[str], group_id: Optional[int]*)

Bases: object

Group label of sample *identifier*. Notes —— Useful for leave-one-group-out cross-validation (LOGO-CV), for example, to take taxonomy into account.

group_id: Optional[int] = None

group_name: Optional[str] = None

identifier: str = None

class phenotrex.structure.records.**PhenotypeRecord**(*identifier: str, trait_name: str, trait_sign: int*)

Bases: object

Ground truth labels of sample *identifier*, indicating presence/absence of trait *trait_name*:

- 0 if trait is absent
- 1 if trait is present

identifier: str = None

trait_name: str = None

trait_sign: int = None

class phenotrex.structure.records.**TrainingRecord**(*identifier: str, group_name: Optional[str], group_id: Optional[int], trait_name: str, trait_sign: int, features: List[str]*)

Bases: *phenotrex.structure.records.GenotypeRecord*, *phenotrex.structure.records.PhenotypeRecord*, *phenotrex.structure.records.GroupRecord*

Sample containing Genotype-, Phenotype- and GroupRecords, suitable as machine learning input for a single observation.

features = None

identifier = None

Module contents

phenotrex.transforms package

Submodules

phenotrex.transforms.annotation module

phenotrex.transforms.resampling module

class phenotrex.transforms.resampling.**TrainingRecordResampler**(*random_state: float = None, verb: bool = False*)

Bases: object

Instantiates an object which can generate versions of a TrainingRecord resampled to defined completeness and contamination levels. Requires prior fitting with full List[TrainingRecord] to get sources of contamination for both classes.

Parameters

- **random_state** – Randomness seed to use while resampling
- **verb** – Toggle verbosity

fit(*records: List[phenotrex.structure.records.TrainingRecord]*)

Fit TrainingRecordResampler on full TrainingRecord list to determine set of positive and negative features for contamination resampling.

Parameters **records** – the full List[TrainingRecord] on which ml training will commence.

Returns True if fitting was performed, else False.

get_resampled(*record: phenotrex.structure.records.TrainingRecord*, *comple: float = 1*, *conta: float = 0*) → phenotrex.structure.records.TrainingRecord

Resample a TrainingRecord to defined completeness and contamination levels. Comple=1, Conta=1 will double set size.

Parameters

- **comple** – completeness of returned TrainingRecord features. Range: 0 - 1
- **conta** – contamination of returned TrainingRecord features. Range: 0 - 1
- **record** – the input TrainingRecord

Returns a resampled TrainingRecord.

Module contents

phenotrex.transforms.**fastas_to_grs**(**args*, ***kwargs*)

phenotrex.util package

Submodules

phenotrex.util.helpers module

phenotrex.util.helpers.**fail_missing_dependency**(**args*, ***kwargs*)

phenotrex.util.helpers.**get_groups**(*records: List[phenotrex.structure.records.TrainingRecord]*) → numpy.ndarray

Get groups from list of TrainingRecords

Parameters **records** –

Returns list for groups

phenotrex.util.helpers.**get_x_y_tn**(*records: List[phenotrex.structure.records.TrainingRecord]*) → Tuple[numpy.ndarray, numpy.ndarray, str]

Get separate X and y from list of TrainingRecord. Also infer trait name from first TrainingRecord.

Parameters **records** – a List[TrainingRecord]

Returns separate lists of features and targets, and the trait name

phenotrex.util.logging module

`phenotrex.util.logging.get_logger(initname, verb=False)`

This function provides a logger to all scripts used in this project.

Parameters

- **initname** – The name of the logger to show up in log.
- **verb** – Toggle verbosity

Returns the finished Logger object.

phenotrex.util.plotting module

`phenotrex.util.plotting.compleconta_plot(cccv_results: Union[Dict[float, Dict[float, Dict[str, float]]], List[Dict[float, Dict[float, Dict[str, float]]]]], conditions: List[str] = (), each_n: List[int] = None, title: str = '', fontsize: int = 16, figsize=(10, 7), plot_comple: bool = True, plot_conta: bool = True, colors: List = None, save_path: Union[str, pathlib.Path] = None, **kwargs)`

Plots Compleconta CV result for one or multiple models. For perfect completeness and variable contamination as well as perfect contamination and variable completeness, the resulting mean balanced accuracy over folds is plotted.

Parameters

- **cccv_results** – a ComplecontaCV result, or list thereof
- **conditions** – A list of condition names associated cccv_results
- **each_n** – A list of sample counts in datasets associated with cccv_results
- **title** – The plot title
- **fontsize** – The fontsize of the plot
- **figsize** – The figure size (tuple of width, height)
- **plot_comple** – Whether to plot completeness
- **plot_conta** – Whether to plot contamination
- **colors** –
- **save_path** – The save path of the plot; if None, display it with plt.show()
- **kwargs** – any further keyword arguments passed to plt.plot()

Returns None

phenotrex.util.taxonomy module

Module contents

4.1.2 Module contents

Top-level package for phenotrex.

CHAPTER

FIVE

CREDITS

5.1 Development Lead

- Lukas Lüftinger <lukas.lueftinger@outlook.com>

5.2 Contributors

- Patrick Hyden <hydenp89@univie.ac.at>
- Roman Feldbauer <roman.feldbauer@univie.ac.at>

CHAPTER

SIX

HISTORY

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

phenotrex, 12
phenotrex.cli, 7
phenotrex.ml.clf, 8
phenotrex.structure, 9
phenotrex.structure.records, 8
phenotrex.transforms, 10
phenotrex.transforms.resampling, 9
phenotrex.util, 12
phenotrex.util.helpers, 10
phenotrex.util.logging, 11
phenotrex.util.plotting, 11

INDEX

C

compleconta_plot() (*in module phenotrex.util.plotting*), 11

F

fail_missing_dependency() (*in module phenotrex.util.helpers*), 10
fastas_to_grs() (*in module phenotrex.transforms*), 10
features (*phenotrex.structure.records.GenotypeRecord attribute*), 8
features (*phenotrex.structure.records.TrainingRecord attribute*), 9
fit() (*phenotrex.transforms.resampling.TrainingRecordResampler method*), 10

G

GenotypeRecord (*class in phenotrex.structure.records*), 8
get_groups() (*in module phenotrex.util.helpers*), 10
get_logger() (*in module phenotrex.util.logging*), 11
get_resampled() (*phenotrex.transforms.resampling.TrainingRecordResampler method*), 10
get_x_y_tn() (*in module phenotrex.util.helpers*), 10
group_id (*phenotrex.structure.records.GroupRecord attribute*), 9
group_name (*phenotrex.structure.records.GroupRecord attribute*), 9
GroupRecord (*class in phenotrex.structure.records*), 8

I

identifier (*phenotrex.structure.records.GenotypeRecord attribute*), 8
identifier (*phenotrex.structure.records.GroupRecord attribute*), 9
identifier (*phenotrex.structure.records.PhenotypeRecord attribute*), 9
identifier (*phenotrex.structure.records.TrainingRecord attribute*), 9

P

phenotrex (*module*), 12
phenotrex.cli (*module*), 7
phenotrex.ml.clf (*module*), 8
phenotrex.structure (*module*), 9
phenotrex.structure.records (*module*), 8
phenotrex.transforms (*module*), 10
phenotrex.transforms.resampling (*module*), 9
phenotrex.util (*module*), 12
phenotrex.util.helpers (*module*), 10
phenotrex.util.logging (*module*), 11
phenotrex.util.plotting (*module*), 11
PhenotypeRecord (*class in phenotrex.structure.records*), 9

T

TrainingRecord (*class in phenotrex.structure.records*), 9
TrainingRecordResampler (*class in phenotrex.transforms.resampling*), 9
trait_name (*phenotrex.structure.records.PhenotypeRecord attribute*), 9
trait_sign (*phenotrex.structure.records.PhenotypeRecord attribute*), 9