A comprehensive comparison of residue-level methylation levels with the regression-based gene-level methylation estimations by ReGear

Jinpu Cai[1], Yuyang Xu[1], Wen Zhang[2], Shiying Ding[1], Yuewei Sun[1], Jingyi Lyu[3], Meiyu Duan[2], Shuai Liu[2], Lan Huang[2], and Fengfeng Zhou[2,#].

1 Health Informatics Lab, College of Software, and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130012, China.

2 Health Informatics Lab, College of Computer Science and Technology, and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130012, China.

3 Health Informatics Lab, College of Life Sciences, Jilin University, Changchun, Jilin 130012, China.

# Correspondence may be addressed to Fengfeng Zhou: FengfengZhou@gmail.com or ffzhou@jlu.edu.cn . Lab web site: http://www.healthinformaticslab.org/ . Phone: +86-431-8516-6024. Fax: +86-431-8516-6024.

Introduction

This software is based on methylated data types. Perform simple preprocessing on the methylation data and delete the methylation feature from the null values in the methylation data set because the number of null values is not very large. Other methods can also be used for null data preprocessing. The original data set is divided into the training set and the test set (this article chooses 2/3 samples as the training set and 1/3 samples as the test set) and then performs genetic level regression on the methylation features. The regression model trained by the training set forms gene-level features on the test set, and then performs a comprehensive comparison of classification performance. This software has provided a platform for genes and residue. The file is saved in json format file.

The software has a total of three parts, which are the three parts, i.e., train, predict and test. The latest version of this software may be found at:

http://www.healthinformaticslab.org/supp/resources.php

train: Given a training set of methylated data, through three regression methods of Linear, L1 , L2 and five different formulas x , x^2 , x^3 , log10(x) , sqrt(x) respectively to the data set regression, the training The model is saved in the pickle file.

predict: Given a divided test set, calculate the gene-level data matrix through the pickle file trained in Train.

test: Given a data matrix (gene level or residue level), and a class label file, get the results of t-test and other feature selection algorithms and classifiers.

This software introduces it as a third-party package and encapsulates the three parts into different functions. The user can choose the regression method to calculate the loci at the gene level. The software also provides a test sample, which can be deleted if not needed.

train

train package function: run ()

The parameter list of the run function is as follows:

train. run (code, train_data, train_label, platform, model_type, data_type)

code: dataSet ID .

Example: "GSE66695".

train_data: train data (dataframe)

The row index is the methylated ID name, and the column index is the sample name.

Example:

ID_REF	sample1	sample2
Feature1	0.564	0.147
Feature2	0.147	0.697

train_label: train label (dataframe).

There is only one column of column labels, the row index is the sample, and the column is the positive and negative sample 1 or 0.

Example:

	label
Sample1	0
Sample2	1

platform: the file corresponding to the gene methylation site already provided by the software. Directly assign "platform.json".

model_type: There are three ways of regression, "LinearRegression", "L1", "L2".
Example: "LinearRegression"

data_type: The formula of data, there are five forms, "origin_data"(x), "radical_data"(x^2), "cube_data"(x^3), "log_data"(log10(x)), "square_data"(sqrt(x)).
Example: "origin_data"

The test train code:

```
import ReGear.train as retrain
retrain.run("GSE66695",train_data,train_label, "platform.json", "LinearRegression", "origin_data")
```

predict

predict (code, test_data, platform, pickle_file, model_type, data_type)

code: dataSet ID
Example: "GSE66695"

test_data: test_data (dataframe)
The row index is the methylated ID name, and the column index is the sample name
Example:

ID_REF	sample1	sample2
Feature1	0.564	0.147
Feature2	0.147	0.697

platform: the file corresponding to the gene methylation site already provided by the software. Directly assign "platform.json"

model_type: There are three ways of regression, "LinearRegression", "L1", "L2".
Example: "LinearRegression"

data_type: The formula of data, there are five forms, "origin_data"(x), "radical_data"(x^2), "cube_data"(x^3), "log_data"(log10(x)), "square_data"(sqrt(x)).
Example: "origin_data"

The test predict code:

```
import ReGear.predict as pre
pre.predict("GSE66695",test_data,platform,pickle_file, "LinearRegression", "origin_data")
```

test

select_feature (code, data, label, gene = True)

code: dataSet ID

Example: "GSE66695"

data: test data (dataframe)

Example:

ID_REF	sample1	sample2
Feature1	0.564	0.147
Feature2	0.147	0.697

label: test data label (dataframe)

Example:

	label
Sample1	0
Sample2	1

gene = True: whether it is at the gene level, if not, change to gene = false

Example:

gene=true

the test test code

```
import ReGear.test as test
test.select_feature("GSE66695", data, label, gene=True)
```