Rheumatoid arthritis and schizophrenia epigenetic biomarkers detected by a recursively feature refining algorithm, TriVote

Cheng Xu[1,*], Jiamei Liu[1,*], Weifeng Yang[1,*], Yayun Shu[1,*] Zhipeng Wei[2], Weiwei Zheng[2], Xin Feng[2], and Fengfeng Zhou[1,2,#]

1 Software College, Jilin University, Changchun, Jilin, China, 130012.

2 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.

* These authors contributed equally to this work.

# Corresponding author: Fengfeng Zhou, e-mail: FengfengZhou@gmail.com or ffzhou@jlu.edu.cn . Web site: http://www.healthinformaticslab.org/.

1. Installation

TriVote is tested under Python interpreter version 2.7, and requires these python packages to run, i.e. scikit-learn , scipy and numpy. TriVote may be installed by the following steps.

1. Decompress TriVote-v1.rar
2. Enter folder TriVote-v1
3. Run the command line:
   python setup.py install

Windows users may enter the command line interface by run the command “cmd.exe”. Linux and Mac users may enter the above command in the shell environment.

TriVote may run directly without installation.

2. Data Format suggested

TriVote needs a data matrix with the integrated class labels. The data matrix is in the TAB-delimited text file. The first line should be the list of sample class labels, and each column is the data of one sample. The first column should be the feature names. The released TriVote package provides an example data matrix file, abbreviated from the dataset GSE42861.

3. Main functions

3.1. Module TriVote.py, function *TransName* (feaName , feaNum)

This function returns the names of the features with their indexes.

Parameters	**Descriptions of parameters**
feaName	Array-like, shape(n_features,1), record the names of the features selected
feaNum	Array-like, shape(n_featuresSelected,1), record the indexes of the features selected

Returns	**Description of returns**
name	Array-like, shape(n_featuresSelected,1), record the names of the features selected

3.2. Module TriVote.py, function *TransData* (feaNum, X_predict, preprocessingFlag=1)

This function is used to select the features of the data to be predicted according to the results of training process.

Parameters	**Descriptions of parameters**
feaNum	Array-like, shape(n_featuresSelected,1) record the indexes of the features selected
X_predict	Array-like, shape (n_samples, n_features), the raw data to be predicted
preprocessingFlag	Integer, if it equals 0, not perform preprocessing , else perform; optional (default 1)

Returns	**Descriptions of returns**
X_new	Array-like, shape(n-samples, n_featuresSelected), record the data after the selection process of Trivote

3.3. Module TriVote.py, function TriVoteFit(m, l, featureNum, preprocessingFlag=1, circleNum=20, classifier=SVC(), expectAccuracy=0.9, KofK_Fold=10, train_size=0.9, showEachResult=0, RandomSeed=0)

This function is the main interface of TriVote. It trains the model and returns the indexes of the features selected and the classifier trained which can be used to predict new samples.

Paramters	**Descriptions of Parameters**
m	Array-like, shape(n_samples, n_features), record the data matrix of training data
l	Array-like, shape(n_samples,1), record the labels of the training data
featureNum	Integer, record the numbers of features selected the user expect to have or the max number of the features selected that the user allow.
preprocessingFlag	Integer, if it equals 0, not perform preprocessing , else perform; optional (default 1)
circleNum	Integer, the number of the circles to vote for the features to be selected. optional (default 20)
classifier	Scikit-learn object,, the classifer to estimate the performance of the features selected. optional (default SVM classifier)
expectAccuracy	Float, the accuracy the user expect the classifier using the features selected can perform.optional (default 0.9)
KofK_Fold	Integer, the k of the k-fold cross-validation to test the performance(default 10)
train_size	Float, the rate of train data to train the classifier
showEachResult	Integer, if it equals 0, not show each result of the process (including the ones do not achieve the users'expectation), else show; optional (default 0)
RandomSeed	Integer, RandomState , (default 0)

Returns	**Descriptions of returns**
clf	Scikit-learn object, the classifer to estimate the performance of the features selected. optional (default SVM classifier)
feanum	Array-like, shape(n_featuresSelected,1) record the indexes of the features selected

3.4. Module load.py, function *file2matrix*(filename, PosName, NeName)

This function is used to load the data matrix from files follow the recommended format.

Paramters	**Descriptions of Parameters**
filename	String, file name of the data matrix
PosName	String, name of the positive class label
NeName	String, name of the negative class label

Returns	**Descriptions of returns**
m	Array-like, shape(n_samples, n_features), record the data matrix of training data
l	Array-like, shape(n_samples,1), record the labels of the training data
feaName	Array-like, shape(n_features,1), record the names of the features selected

4. Example project

An example project is provided with the demonstration python code and data file.

- Demonstration code file: runTriVote.py
- Example data matrix file: abbrGSE42861.txt

The user may run the demonstration code by typing the following command in the command line interface:

```
> python runTriVote.py
```

The code will automatically analyze the example data matrix file abbrGSE42861.txt. And the selected feature list can be found in the text file featureSelected.txt. The user may change the file name in the demonstration code to analyze a new dataset.

The format of a data matrix file may refer to the file abbrGSE42861.txt.

5. Extra transcriptomic datasets

17 transcriptomic datasets used in this study and the following reference were also provided as TEXT files.

R Ge, M Zhou, Y Luo, Q Meng, G Mai, D Ma, G Wang, **Fengfeng Zhou**. “McTwo: a two-step feature selection algorithm based on maximal information coefficient”. BMC bioinformatics 17 (1), 142