Improved prediction of breast cancer outcome by identifying heterogeneous biomarkers

User Manual
(version 1.0)

March 4, 2018

Index

1. Installation

This library ‘CPR’ requires:

Python 3,
Numpy,(≥1.6.1)
Scipy, (≥0.9)
Scikit-learn (≥0.18)

To download or update those libraries, use ‘pip’ in command line.

```
pip install numpy scipy scikit-leanr
```

Download the CPR module from https://github.com/mathcom/CPR.
If successfully downloaded, user can find the following 4 files:

CPR.py
ex_EXPRESSION.txt
ex_NETWORK.txt
ex_CLINICAL.txt

2. Quick start

1) Run

```
$ python CPR.py [-h] [-m NUMCLUSTERS] [-d DAMPINGFACTOR]
                [-n NUMBIOMARKERS] [-c CONDITIONHUBGENE]
                EXPRESSION_FILE NETWORK_FILE CLINICAL_FILE
                RESULT_FILE
```

Option	Name	Description
	EXPRESSION_FILE *(positional argument)*	Tab-delimited file for gene expression profiles as below: PATIENT TCGA-AR-A24H TCGA-AR-A24L TCGA-AR-A24M A1CF -0.436158 -0.276784 -0.309453 A2M 1.90128 2.72735 4.03939 A4GALT -0.408337 -0.247608 -0.260444
	NETWORK_FILE *(positional argument)*	Tab-delimited file for gene interaction network as below: GENE1 GENE2 RPL37A RPS27A MRPL1 MRPS36 RFC3 SPRTN
	CLINICAL_FILE *(positional argument)*	Tab-delimited file for patient’s clinical data as below. LABEL=**1**:*poor* prognosis and **0**:*good* prognosis: PATIENT LABEL TCGA-AR-A24H 0 TCGA-AR-A24L 0 TCGA-AR-A2LH 1
	RESULT_FILE *(positional argument)*	The results of CPR are saved with the following names: 1) *RESULT_FILE_biomarker.txt* 2) *RESULT_FILE_subnetwork.txt* 3) *RESULT_FILE_accuracy.txt*

Option	Name	Description
-h	*Help message* *(optional argument)*	Show this help message and exit
-m	*Number of sample clusters* *(optional argument)*	A parameter of *K-means clustering* algorithm. This parameter decides number of sample clusters to handle the heterogeneity of patients. The default value makes the number of clusters be determined by using the *silhouette score*. If a specific number is given, the K-means is carried out with the number. (default=0)
-d	*Damping factor* *(optional argument)*	A parameter of *PageRank* algorithm. This parameter decides an influence of network information on prediction. The value must be between 0 and 1. (default=0.7)
-n	*Number of markers* *(optional argument)*	This parameter decides number of biomarkers to use in prediction. (default=70)
-c	*Condition of hub-gene* *(optional argument)*	This parameter is used to identify a hub-gene. When *c* is a given parameter and *x* is the total of genes, we define top *cx* genes with high degree as hub-genes. The value must be between 0 and 1. (default=0.02)

2) Example

$ **python CPR.py** ex_EXPRESSION.txt ex_NETWORK.txt ex_CLINICAL.txt ex_RESULT.txt

(1) Log in command line

```
>>> 0. Arguments
Namespace(CLINICAL_FILE='ex_CLINICAL.txt',      EXPRESSION_FILE='ex_EXPRESSION.txt',
NETWORK_FILE='ex_NETWORK.txt',    RESULT_FILE='ex_RESULT',    conditionHubgene=0.02,
dampingFactor=0.7, numBiomarkers=70, numClusters=0)
>>> 1. Load data
>>> 2. Preprocess data
   n_samples: 189
   n_genes  : 8819
   n_edges  : 150168
>>> 3. Conduct CPR
   K-means clustering
   -> n_clusters: 3
      In cluster[0], n_samples:76, n_goods:44, n_poors:32
      In cluster[1], n_samples:107, n_goods:52, n_poors:55
      In cluster[2], n_samples:6, n_goods:3, n_poors:3
   Modified PageRank
>>> 4. 10-fold Cross validation
   10% complete!
   20% complete!
   30% complete!
   40% complete!
   50% complete!
   60% complete!
   70% complete!
   80% complete!
   90% complete!
   100% complete!
   AUC-ROC : 0.710
   Accuracy: 0.640
```

(2) ex_RESULT_biomarker.txt

- PRscore is the mean of scores computed in each sample cluster.

```
GeneSymbol    PRscore
PIK3R2        1.684137
HSPA8         1.645991
CALM1         1.555840
POLR2L        1.516215
HDAC2         1.481628
H3F3A         1.480993
GSK3B         1.422521
EGFR          1.395188
POLR2C        1.383287
```

(3) ex_RESULT_subnetwork.txt

- A resulting subnetwork is a subnetwork of the given whole network.
- Each edge has at least one biomarker gene.

```
source        target
MAPK13        NFATC3
CREBBP        EP400
GMNN          PSMC2
RNPS1         RPSA
EGFR          TLR2
```

(4) ex_RESULT_accuracy.txt

```
AUC-ROC : 0.710
Accuracy: 0.640
```

3. Description of class CPR

A user can import and utilize CPR.py in python. CPR.py provides one class and its three functions.

1) class CPR

Clustering and PageRank-based gene selection method. For more detail, please refer to "Improved prediction for breast cancer outcome by identifying heterogeneous biomarkers".

Parameters	**dampingFactor**: float, optional (default=0.7) A parameter of *PageRank* algorithm. This parameter decides an influence of network information on prediction Range = 0.0 ~ 1.0 **n_biomarkers**: int, optional (default=70) User can control the number of prognostic biomarkers. **n_clusters**: int, optional (default=0) A parameter of *K-means clustering* algorithm.

	This parameter decides number of sample clusters to handle the heterogeneity of patients. The default value makes the number of clusters be determined by using the *silhouette score*. If a number is given, the K-means is carried out with the number. **c_hubgene**: float, optional (default=0.02) This parameter is used to identify a hub-gene. When *c* is a given parameter and *x* is the total of genes, we define top *cx* genes with high degree as hub-genes. **logshow**: bool, optional (default=False) If *logshow* is *False*, any log in class CPR is not shown in command line.

2) method CPR.fit()

Build a CPR gene selection model.

Parameters	**expr**: numpy.array(shape=[n_samples,n_genes], dtype=numpy.float32) A gene expression dataset without any header. The order of sample must be equal to one of *labels*. The order of gene must be equal to one of *genes*. **labels**: numpy.array(shape=[n_samples], dtype=numpy.int32) A value of vector represents a label of sample. (0: *good* prognosis and 1: *poor* prognosis) The order of label must be equal to one of sample in *expr*. **genes**: list A list of genes in expression data. The order of gene must be equal to one of gene in *expr*. **edges**: list A list of edges, and all edges have *tuple* type. **random_state**: int or None, optional (default=1) This parameter is used in scikit-learn functions. If *int*, the results are always same. If *None*, a result can be different each time.

3) method CPR.get_biomarkers()

A list of biomarkers identified by built model is provided as list type.

Returns	**biomarkers**: list An element of the result has *tuple* type and two values. (0:gene symbol and 1:PRscore) The list is sorted by the descending order of PRscore.

4) method CPR.get_subnetwork()

A list of edges containing at least one biomarker is provided as list type.

Returns	**edges**: list An element of the result has *tuple* type. Each tuple consists of two gene symbols.

4. Contact

Bug reporting, questions or any suggestions are highly appreciated.

Jonghwan Choi (mathcom@inu.ac.kr)

Jaegyoon Ahn (jgahn@inu.ac.kr)

5. Reference

Choi, Jonghwan, et al. "Improved prediction of breast cancer outcome by identifying heterogeneous biomarkers." *Bioinformatics* **33.22** (2017): 3619-3626.