G2Vec: Distributed gene representations for identification of cancer prognostic genes

User Manual

(version 0.1)

April 23, 2018

Index

1. Installation

This library 'G2Vec' requires:

Python 3,
Numpy (≥1.6.1),
Scikit-learn (≥0.18),
Tensorflow (≥1.4)

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

pip install numpy scikit-learn tensorflow

Download the G2Vec module from https://github.com/mathcom/G2Vec.
If successfully downloaded, a user can find the following 4 files:

G2Vec.py,
ex_EXPRESSION.txt,
ex_CLINICAL.txt,
ex_NETWORK.txt

2. Quick start

1) Run

$ **python G2Vec.py** [**-h**] [**-p** *LENPATH*] [**-r** *NUMREPETITION*]
[**-s** *SIZEHIDDENLAYER*] [**-l** *LEARNINGRATE*]
[**-n** *NUMBIOMARKER*]
EXPRESSION_FILE ***CLINICAL_FILE*** ***NETWORK_FILE***
RESULT_NAME

Option	Name	Description
	EXPRESSION_FILE *(positional argument)*	Tab-delimited file for gene expression profiles as below: PATIENT TCGA-2Y-A9GS TCGA-2Y-A9GU TCGA-2Y-A9GX A1BG 14.467 13.187 14.775 A1CF 10.620 10.291 10.08 A2LD1 6.638 11.509 6.882
	CLINICAL_FILE *(positional argument)*	Tab-delimited file for patient's clinical data as below (LABEL= **0**:*good* prognosis and **1**:*poor* prognosis): PATIENT LABEL TCGA-2Y-A9GS 1 TCGA-2Y-A9GU 0 TCGA-2Y-A9GX 0
	NETWORK_FILE *(positional argument)*	Tab-delimited file for gene interaction network as below: GENE1 GENE2 EIF2A YWHAE HNF4A HIST1H2AC TRA2B POLR2F

Option	Name	Description
	RESULT_NAME *(positional argument)*	The results of G2Vec are saved with the following names: *1) RESULT_FILE_biomarkers.txt* *2) RESULT_FILE_lgroups.txt* 3) *RESULT_FILE_vectors.txt*
-h	***Help message*** *(optional argument)*	Show this help message and exit
-p	***Length of random paths*** *(optional argument)*	This parameter represents the maximum length of random paths generated from gene correlation networks. (default=80)
-r	***Repetition number of random walk procedure*** *(optional argument)*	This parameter decides how many random paths are generated from each gene correlation network. For a given positive integer *r*, a random walker departs from all genes *r* times, resulting that a maximum of *r* random paths will be generated for each gene. (default=10)
-s	***Size of hidden layer*** *(optional argument)*	This parameter decides the number of hidden neurons, which is equal to the dimension of distributed representations. (default=128)
-l	***Learning rate*** *(optional argument)*	This parameter decides how quickly weights in a neural network converge. (default=0.005)
-n	***Number of biomarkers*** *(optional argument)*	This parameter is number of biomarkers identified from each *L-group*. If *N* is given, then the total of biomarkers is *2N*. (default=50)

2) Example

$ **python G2Vec.py** ex_EXPRESSION.txt ex_CLINICAL.txt ex_NETWORK.txt ex_RESULT

(1) Log in command line
- The number of random paths can be different from the below log due to random walk algorithm.
- The results of training can be different because of variable initialization and early stopping.

```
>>> 0. Arguments
Namespace(CLINICAL_FILE='ex_CLINICAL.txt',
EXPRESSION_FILE='ex_EXPRESSION.txt', NETWORK_FILE='ex_NETWORK.txt',
RESULT_FILE='ex_RESULT', epoch=500, learningRate=0.005, lenPath=80,
numBiomarker=50, numRepetition=10, sizeHiddenlayer=128)
>>> 1. Load data
>>> 2. Preprocess data
   n_samples: 135
   n_genes  : 7523    (common genes in both EXPRESSION and NETWORK)
   n_edges  : 216540  (edges with the common genes)
>>> 3. Generate random paths from each group
   *** most time consuming step ***
   n_paths : 45402
   n_genes : 3773     (genes in good or poor random paths)
>>> 4. Compute distributed representations using modified CBOW
   Start training the modified CBOW with early stopping
   - Epoch: 000       ACC[val]=0.6336 ACC[tr]=0.6310 (2.369 sec)
   - Epoch: 005       ACC[val]=0.8044 ACC[tr]=0.8232 (10.459 sec)
   - Epoch: 010       ACC[val]=0.8434 ACC[tr]=0.8633 (11.008 sec)
```

```
    - Epoch: 015       ACC[val]=0.8626 ACC[tr]=0.8860 (10.811 sec)
    - Epoch: 020       ACC[val]=0.8728 ACC[tr]=0.9006 (11.119 sec)
    - Epoch: 025       ACC[val]=0.8812 ACC[tr]=0.9106 (10.898 sec)
    - Epoch(stop): 027  ACC[val]=0.8837 ACC[tr]=0.9142 (6.811 sec)
    Optimization Finish
>>> 5. Find L-groups
>>> 6. Select biomarkers with gene scores
>>> 7. Save results
    ex_RESULT_biomarkers.txt
    ex_RESULT_lgroups.txt
    ex_RESULT_vectors.txt
```

(2) ex_RESULT_biomarkers.txt

- A list of biomarkers identified by G2Vec.py

```
GeneSymbol
ADH1C
AKAP13
ARHGEF3
ARTN
ATG7
```

(3) ex_RESULT_lgroups.txt

- A list of whole genes and their *L-group* labels
- tab-delimited format

```
GeneSymbol    Lgroup(0:good,1:poor,2:other)
A1CF          2
A2M           2
AAAS          2
AAK1          1
AARS          2
```

(4) ex_RESULT_vectors.txt

- A list of distributed gene representations computed by G2Vec
- tab-delimited format

GeneSymbol	V0	V1	V2
A1CF	0.092807	-0.044005	-0.027056
A2M	0.098863	-0.061118	0.02870
AAAS	-0.044920	-0.086036	0.100735
AAK1	-0.025246	0.031078	0.09121

3. Contact

Bug reporting, questions or any suggestions are highly appreciated.

Jonghwan Choi (mathcom@inu.ac.kr)

Jaegyoon Ahn (jgahn@inu.ac.kr)

4. Reference

Jonghwan Choi, et al. " G2Vec: Distributed gene representations for identification of cancer prognostic genes" (submitted)