Instruction for the computer codes of "D-CCA: A Decomposition-based Canonical Correlation Analysis for High-Dimensional Datasets"

July 3, 2018

This is the "Readme" file for the computer codes to reproduce the simulation and real data results of the paper.

1. Software Requirements

Python: version 3.5.2

Matlab: version R2014b

R: version 3.3.3

The following packages are used for the 7 competing methods mentioned in the paper.

JIVE & R.JIVE algorithm: R package **r.jive** with version 2.1

AJIVE algorithm: version on 03/21/2018 from
https://github.com/MeileiJiang/AJIVE_Project

OnPLS algorithm: version on 09/26/2017 from https://github.com/tomlof/OnPLS

DISCO-SCA algorithm: KULEUVEN's software with version on 12/10/2013 from
http://ppw.kuleuven.be/okp/software/disco-sca/download/ and the R package **RegularizedSCA** with version 0.5.3

COBE algorithm: version on 10/13/2015 from
http://bsp.brain.riken.jp/~zhougx/resources/mcode/demo_CIFE.zip

GDFM algorithm: version on 01/26/2018 from
http://www.barigozzi.eu/BHS_final_codes.zip

2. Simulation Studies (in the folder simulation)

You need to change all the file paths in the codes.

Our proposed algorithm is written in the python file **dcca.py**

For **setupX** with **X=1,2**, use **simulationX_dcca.py/simulationX_dcca_AR1.py** and then **result_plot.R/result_plot_AR1.R** to generate Figures 3/S.1 and 4/S.2 in the paper. Use **simulationX_dcca2.py/simulationX_dcca2_AR1.py** to generate the results for Table 1/S.1. Use **simulationX_data.py** to simulate the data for Table 2, where D-CCA and the 7 other competing methods were respectively applied using **simulationX_data.py**,

simulationX_JIVE_and_RJIVE.R, **simulationX_AJIVE.m**, **simulationX_OnPLS.py**, **simulationX_DISCOsca.R**, **simulationX_COBE.m** and **simulationX_gdfm.m**. Then use **result_table.R/result_table_AR1.R** to generate the summary results shown in Tables 1/S.1, 2 and 3. You may use the R files with the prefix "**jobs_**" to submit jobs for the corresponding code files prefixed with "**simulationX_**" to run 1000 replications with different seeds in your computer UNIX cluster.

For **setup3,** use **simulation3_dcca.py** to generate the data and run our proposed D-CCA method. Run **simulation3_JIVE_and_RJIVE.R**, **simulation3_OnPLS.py** and **simulation3_gdfm.m** for JIVE/R.JIVE, OnPLS and GDFM methods, respectively. Use the MATLAB code **simulation3_others.m** to run the other methods and also to reproduce the Figure 5 for Setup 3. Use **result_table.R** to generate the summary results shown in Table 3 for Setup 3.

3. Real-Data Analysis (in the folder realdata)

Download the following datasets:

BRCA817_20140528_log_medcntr.txt: the gene expression matrix used for the paper[txt] on https://tcga-data.nci.nih.gov/docs/publications/brca_2015/

BRCA_freeze_3.26.2014_ver06102014.xlsx: Data freeze details[excel] on https://tcga-data.nci.nih.gov/docs/publications/brca_2015/

BRCA.methylation.27k.450k.txt: BRCA.methylation.27k.450k.zip - Full Methylation Data Set (139M) on https://tcga-data.nci.nih.gov/docs/publications/brca_2012/

You need to change all the filepaths in the codes.

Preprocess the datasets by **BRCAdata_preprocess.R**. Run **realdata_analyze_dcca.py** for the proposed D-CCA method. Run JIVE/R.JIVE, OnPLS and GDFM methods by code files with name starting with **realdata_analyze_JIVEandRJIVE realdata_analyze_OnPLS** and **realdata_analyze_gdfm,** respectively. Run the other methods and the information in Tables 4 and 5 of the paper by **BRCAdata_analysis1_above90.m** and **BRCAdata_analysis1_below90.m**. Use the Matlab codes with name starting with **realdata_analyze_gdfm** to generate the results in Table 6 for GDFM**.** Use the R codes with name starting with **BRCAdata_analysis2 to** generate the summary results shown in Table 3 for TCGA datasets.