NOTE 1: The program was developed on the Ubuntu Linux 16.04 LTS platform. It is written in MATLAB and last tested under the version 2017b.

NOTE 2: The benchmarks were implemented in parallel, using MATLAB "parfor" loops. To run the same benchmarks without changing the code, the user needs MATLAB Parallel Computing Toolbox.

A. Downloading and installing the program (Linux)

- Download the package "cs_code_data.tar.gz" from http://bioinfo.cs.uni.edu/CS.html
- In the terminal window, "cd" into the directory where you downloaded the above file
- "unzip" and "untar" the file as follows:
  - % gunzip cs_code_data.tar.gz
  - % tar -xvf cs_code_data.tar
- Note that the "cs_code_data" directory contains two subdirectories, "cs_code" and "cs_data". The data files are inside "cs_data" whereas the benchmarking scripts and other MATLAB routines are inside "cs_code".

B. Contents of "cs_data"

- "INTERACTION_MATRIX" is the binary matrix of drug-ADR associations (1 represents a known association and 0 represents no known association) extracted from the SIDER 4.1 database.
- "IMPUTE_MATRIX" is the binary matrix of drug-ADR associations observed in the post-marketing phase (also derived from SIDER 4.1).
- "TANIMOTO_MATRIX" specifies pairwise drug similarities. Each entry $s(i, j)$ represents the Tanimoto similarity (Jaccard index) of drugs i and j [.
- ADR_PATH_LESK_MATRIX is the matrix of pairwise ADR similarity scores computed using the UMLS-Similarity program. Each score is the weighted average of the scores obtained using the "path" and "lesk" measures.
- "FEATURE_VECTORS" is the binary matrix consisting of drug's PubChem descriptors and is needed by the CCA program.
- The subdirectory "small example" contains "toy" files that are much smaller in size and is useful in debugging and testing the program.
- Other files (most of which have self-explanatory names) are not used by the program.

C. Contents of the program directory "cs_code"

- The “cs_code” directory contains the MATLAB code itself (a set of files with “.m” extensions”). The code contains all routines necessary for running the three methods benchmarked in the paper, namely CS, ML and CCA.

- **Compressed sensing (CS)** method requires two routines:

    - *WeightImputeLogFactorization* routine takes as input an incomplete matrix of drug-ADR associations (along with required parameters) and outputs the component matrices of drugs’ and ADRs’ latent preferences. Those two matrices are then combined into the matrix of predicted drug-ADR associations.

    - *WeightedProfile* routine improves the prediction obtained by the previous routine in the case of a “cold-start”. The “cold-start” refers to the situation where a drug does not have any known ADRs. In other words, an entire row of an input association matrix corresponding to a drug consists of all zeros. The *WeightedProfile* routine predicts those scores based upon the ADR scores assigned to the drug’s nearest neighbors.

- **Multi-label (ML)** learning program requires two routines:

    - *MLKNN_TEST* takes as input an incomplete matrix of drug-ADR associations R, a pairwise drug similarity matrix, and a set of indices $\{(i,j)\}$ of the matrix R.

    - For the indices $\{(i,j)\}$, the above routine makes predictions $\{R(i,j)\}$ based upon the prior and posterior probabilities of drug-ADR associations computed in the subroutine *MLKNN_TRAIN*.

- **Canonical Correlation Analysis (CCA)** is ran as follows:

    - *ComputeCCA and GenerateCCAMatrix functions* perform Penalized Matrix Decomposition (PMD) of the matrix $R'Y$ where R is the incomplete matrix of drug-ADR associations and Y is the matrix whose rows represent drugs’ PubChem fingerprints.

    - To obtain drug-ADR predictions, the transpose of the output matrix of the above function is multiplied from the left by the drug-feature matrix Y.

D. <u>Running the program</u>

- **Note: Due to the repository size limit, before running the code, the files ADR_PATH_LESK_MATRIX.gz and ADR_PATH_MATRIX.gz should be unzipped.**

- We implemented parallel version of the benchmarking experiments, taking advantage of the MATLAB Parallel Computing Toolbox.

- The user can take advantage of routines available in “cs_code” to run cross-validation benchmarks in different settings. To mask out (hide) different drug-ADR associations, one

can choose between the *OffTargetIndicesTestSet* and *ColdStartTestIndicesSet* routines. The first routine selects a set of indices from the input association matrix purely at random, whereas the second routine selects indices from the same rows (or columns) and is used in the “cold-start” setting.

- The script *ScriptRunCrossValADR.m* will perform a CV segregated by ADRs.

- The script “ScriptRunCrossVal.m” will perform a default CV with test indices chosen at random if the “mode” parameter is set to “0”. If “cold-start” setting on drugs is desired, the mode parameter should be set to 1.