Deep Learning Features Encode Interpretable Morphologies within Histological Images

Ali Foroughi pour, Brian S. White, Jonghanne Park, Todd B. Sheridan, and Jeffrey H. Chuang

The Jackson Laboratory for Genomic Medicine, Farmington, CT

**Introduction:**

The code for generating results of the manuscript is provided. Each code is a script, internally reading all the relevant data and setting all necessary parameters.

In addition to the code, sample data to reproduce the results and validate the outputs is provided. Given the large size of all data processed in the manuscript, only a subset of data for demo simulations is provided. The data of TCGA-BRCA is provided for differential mone analysis and mone-mone correlation analysis. To train the 38 class logistic regression and MLDA classifiers we have provided a small subset of mone values for all cancers. For mone-gene cross correlation analysis only the processed data for the LUAD immune response is provided.

Upon publication, all code for mone and integrative mone-gene analysis will be made available on github at <https://github.com/aforoughipour/mone>. All scripts will also be converted to python functions allowing the user to adjust the parameters.

Given python is a cross platform programming language our code neither has any specific operating system dependency not any non-standard hardware.

**Python installation:**

A python installation is required for running the scripts (ver=3.7 or 3.8). The following packages need to be installed prior to running the code:

* numpy, ver ≥1.19
* scipy, ver ≥1.6.2
* scikit learn, ver≥0.22
* pickle-mixin, ver=1.0.2
* statsmodels, ver ≥0.12.2
* pandas, ver ≥1.2.4
* matplotlib, ver ≥3.3.4
* seaborn, ver ≥ 0.11.1
* venn, ver≥0.1.3
* random and os packages are required for running the code, but they are usually provided with anaconda installations by default.

Installation of the above packages should not take more than a few minutes on a normal computer. Note these packaged may have additional dependencies. The dependencies are automatically installed while installing these packages through pip.

**Running each script:**

Each python code is a script which reads the necessary data, runs the analysis, and saves the results. The folder “ExampleCode” should be the working directory of python. All data are read from, and all results are saved in this folder. All code was tested in the spyder IDE (ver=4.1.5) installed as part of Anaconda (4.10.1).

**Code description:**

Each script, its function, and its details are provided below. All scripts except “lr\_mlda\_38class.py” should not take more than 3 minutes on a normal computer. “lr\_mlda\_38class.py” takes ~1hour for each iteration (default iteration number =10) on a normal computer using the provided sample data.

**cluster\_map.py:**

This script generates the cluster map of each cancer. In the provided demo the clustermap of TCGA-BRCA is generated. Data (‘brca.p’) is read from, and the clustermap (‘cancername\_cmap.png’, here generating ‘brca\_cmap.png’) is saved in the current folder.

**diff\_mone\_analysis.py**

This script performs differential mone analysis for a given cancer, and outputs (prints) the number of mones with distributional differences by each statistical test. It also saves the venn diagram of the number and ratios of overlapping mones across the statistical tests. The diagram is saved as a png file under the name “venn\_cancer\_frozen\_rawnum.png” and “venn\_cancer\_frozen\_ratio.png” for the number and ratios of statistically significant mones, respectively. “cancer” is replaced with the cancer being analyzed. The scipy implementations of t-test, KS test, and WRS test are used. The BH procedure of the statsmodels package is used for FDR correction for t-test, KS test, and WRS test. FDR-OBF is implemented by the authors as described in the “methods” section. All methods bound FDR by 5%.

**pairwise\_mone\_corr\_analysis.py:**

This script loads the data of each cancer, computes the correlation matrices of tumor and normal slides as described in the “methods” section of the manuscript, generates the clustermap of the correlation matrices, and outputs (prints) the number of significant mone-mone correlations as well as their ratio. The clustermap of the tumor slides is saved as ‘cancer\_tumorclustergram.png’ and the differential clustermap is saved as ‘cancer\_ differentialclustergram.png’. For the provided BRCA data “cancer” is replaced with “brca” in the file names.

**lr\_mlda\_38class.py:**

This script reads the mone values of 19 cancers, splits the data to train and test at the patient level, runs MLDA and logistic regression with LASSO penalty, computes the AUCs for test data, and saves the computed AUCs as a python dictionary in a pickle file. For iteration “cnt”, the pickle file is named: MLDA\_LR\_runF“cnt”\_down\_samp.p. This process iterates 10 times. The following AUCs are computed:

* MLDA and LR-LASSO models separating all 38 classes (19 cancers, tumor/normal status) are trained using training data. The one versus rest AUC of each cancer is computed using test data. These AUCs are saved in aucMLDA38 and aucLR38 dictionaries for MLDA and logistic regression, respectively.
* The AUC of a universal tumor detector, which sums the tumor probability of the 38class logistic regression model across all 19 cancer types, is computed as saved. This value is saved as a dictionary called “aucLRtn38”.
* The AUC of a cancer type detector, summing the tumor and normal probabilities of each cancer type. This dictionary is saved as “aucLRtissue”.
* The confusion matrix of the LR-LASSO model on the test data. This is saved as “cm38”.
* A tumor/normal classifier trained using the training data of each cancer, and applied to test data of all cancers. The cross-classification AUC matrix is saved as a dictionary called “aucCC”. In the aucCC dictionary, each key is the “train cancer+test cancer+the value of C used to train the classifier”. For example, brcaucec10, is the AUC of a tumor/normal classifier trained on BRCA (with C=10 in sklearn logistic regression) that is applied to the UCEC test slides.

Given the large size of data to run the code, a subset of computed mones for all cancers is provided. These mone values are provided as pickle files in the “down\_samp\_cans” folder. To account for the smaller data size C is reduced from 100 to 10 (the results provided in the manuscript use C=100). Although the value of C is altered, the code clearly shows the high AUC of linear models for separating classes. Note an example output for one iteration of the code is provided (‘MLDA\_LR\_runF0\_down\_samp.p’).

**luad\_immune.py**

The code for integrative mone-gene analysis of immune response in LUAD is provided. Given the large size of gene expression vectors, the log-normalized FPKMs of immune related genes, and other variables within the code necessary to reproduce the results, are saved and provided as a pickle file (LUADimmuneData.p). The portion of the code performing this task is kept, but is commented in the provided script. The script reads the necessary data, computes the correlation matrix, identifies the significant correlations, prints the number of statistically significant correlations, and produces the correlation and adjusted p-value clusterings that are provided in the manuscript. The correlation matrix is saved as 'luad\_immune\_MG.png', and the adjusted p-value matrix is saved as 'luad\_immune\_MG\_adj\_pvals.png'