# **Software Description**

# Overview

This is the software package for implementing the methods and reproducing the results described in the manuscript. The main functions of the software package include:

* Train a deep neural network to predict the clinical endpoint outcomes from multi-omics data.
* Compare the performance of multiethnic machine learning schemes.
* Use deep transfer learning to improved machine learning model performance on data-disadvantaged racial groups.
* Use simulation modeling to study the key factors influencing the performance of the multiethnic machine learning schemes.

# Software Structure

The example folder contains the scripts for reproducing the result in Fig 3 and extended Fig 4.

The data folder contains the script file to read from TCGA and MMRF-COMMPASS data.

The model folder contains the files for deep neural network and deep transfer learning implementation.

The simulation folder contains two synthetic datasets simulated from ssizeRNA (a Bioconductor package), a data sampler, and four files, for reproducing the results in Fig 4.

|  |  |  |
| --- | --- | --- |
| **Entity** | **Path/location** | **Note** |
| Deep neural network | ./model/mlp.py | The deep network model |
| Logistic regression | ./model/LogisticRegression.py | The logistic regression layer |
| Stacked auto-encoder | ./model/SdA.py | Functions to layerwise train a stacked de-noising auto-encoder. |
| Feature selection | ./ model/mlp.py/selectKBest | Feature selection for training and testing datasets. |
| Synthetic datasets 1, 3 | ./simulation/ PanGyn-DFI-5.mat | The simulation dataset using parameters estimated from PanGyn-AA/EA-Protein-DFI-5YR. |
| Synthetic datasets 2, 4 | ./simulation/ PanGyn-DFI-5-base.mat | The simulation dataset with no distribution difference. |
| Fine-tuning 1 | ./examples/classify\_util.py/ run\_supervised\_transfer\_cv | The supervised transfer learning method |
| Fine-tuning 2 | ./examples/classify\_util.py/ run\_unsupervised\_transfer\_cv | The unsupervised transfer learning, stacked auto-encoder. |
| CCSA | ./examples/classify\_util.py/ run\_CCSA\_transfer  ./model/CCSA/Initialization.py | The Contrastive Classification Semantic Alignment transfer learning method. |

# System Requirements

## Software dependency

The system relies on the following software, reagent, or resources.

|  |  |  |
| --- | --- | --- |
| **REAGENT or RESOURCE** | **SOURCE** | **IDENTIFIER** |
| TCGA | Genomic Data Commons data portal | <https://portal.gdc.cancer.gov/> |
| MMRF | Genomic Data Commons data portal | <https://portal.gdc.cancer.gov/projects/MMRF-COMMPASS> |
| TCGA Cancer Types | Broad Institute | <https://gdac.broadinstitute.org/> |
| American Cancer Types | Cancer Treatment Centers of America | <https://www.cancercenter.com/cancer-types> |
| TCGA Ancestry | TCGAA19 | <http://52.25.87.215/TCGAA/index.php> |
| TCGA Protein | Genomic Data Commons data portal | <https://portal.gdc.cancer.gov/> |
| TCGA mRNA | Genomic Data Commons data portal | <https://portal.gdc.cancer.gov/> |
| TCGA Clinical | Genomic Data Commons data portal | <https://portal.gdc.cancer.gov/> |
| MMRF tool | Genomic Data Commons data portal | <https://github.com/cpreid2/gdc-rnaseq-tool> |
| TCGA Clinical Endpoints | TCGA Pan-Cancer Clinical Data Resource | <https://www.sciencedirect.com/science/article/pii/S0092867418302290> |

## Software version

Our software has been tested on the following software version.

|  |  |  |
| --- | --- | --- |
| **Software and Hardware** | **SOURCE** | **IDENTIFIER** |
| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
| Python 2.7 | Python Software Foundation | <https://www.python.org/download/releases/2.7/> |
| Computational Facility | The National Institute for Computational Sciences | <https://www.nics.tennessee.edu/computing-resources/acf> |
| Numpy 1.15.4 | Tidelift, Inc | https://libraries.io/pypi/numpy/1.15.4 |
| Numpydoc 0.9.1 | Tidelift, Inc | <https://libraries.io/pypi/numpydoc> |
| Scipy 1.2.1 | The SciPy community | <https://docs.scipy.org/doc/scipy-1.2.1/reference/> |
| Seaborn 0.9.0 | Michael Waskom | <https://seaborn.pydata.org/installing.html> |
| Sklearn 0.0 | The Python community | <https://pypi.org/project/sklearn/> |
| Skrebate 0.6 | Tidelift, Inc | <https://libraries.io/pypi/skrebate> |
| Theano 1.0.3 | LISA lab | <http://deeplearning.net/software/theano/install.html> |
| Keras 2.2.4 | GitHub, Inc. | <https://github.com/keras-team/keras/releases/tag/2.2.4> |
| Keras-Applications 1.0.8 | GitHub, Inc. | <https://github.com/keras-team/keras-applications> |
| Keras-Preprocessing 1.1.0 | GitHub, Inc. | <https://github.com/keras-team/keras-preprocessing/releases/tag/1.1.0> |
| Tensorboard 1.13.1 | GitHub, Inc. | <https://github.com/tensorflow/tensorboard/releases/tag/1.13.1> |
| Tensorflow 1.13.1 | tensorflow.org | <https://www.tensorflow.org/install/pip> |
| Tensorflow-estimator 1.13.1 | The Python community | <https://pypi.org/project/tensorflow-estimator/> |
| Statsmodels 0.9.0 | Statsmodels.org | <https://www.statsmodels.org/stable/release/version0.9.html> |
| Lifelines 0.16.3 | Cam Davidson-Pilon Revision | <https://lifelines.readthedocs.io/en/latest/Changelog.html> |
| Optunity 1.1.1 | The Python community | <https://pypi.org/project/Optunity/> |
| Xlrd 1.2.0 | The Python community | <https://pypi.org/project/xlrd/> |
| XlsxWriter 1.1.8 | The Python community | <https://pypi.org/project/XlsxWriter/> |
| Xlwings 0.15.8 | The Python community | <https://pypi.org/project/xlwings/> |
| Xlwt 1.3.0 | The Python community | <https://pypi.org/project/xlwt/> |
| Lasagne 0.2.dev1 | GitHub, Inc. | <https://github.com/Lasagne/Lasagne> |

## Hardware requirements

We recommend use a GPU (V100) to speed up the running process of our software.

# Installation Guide

Our software package can be downloaded from the following github page: <https://github.com/AtlasGao/TransferV1>. This package contains the source code and demo datasets to reproduce the results represented in our paper. Our software can run on Windows and Ubuntu, but we suggest using Linux system which is easier for environment configuration.

Conda –install requirements.txt

Requirements.txt

numpy==1.15.4

numpydoc==0.9.1

scipy==1.2.1

seaborn==0.9.0

sklearn==0.0

skrebate==0.6

Theano==1.0.3

Keras==2.2.4

Keras-Applications==1.0.8

Keras-Preprocessing==1.1.0

tensorboard==1.13.1

tensorflow==1.13.1

tensorflow-estimator==1.13.0

statsmodels==0.9.0

lifelines==0.16.3

Optunity==1.1.1

xlrd==1.2.0

XlsxWriter==1.1.8

xlwings==0.15.8

xlwt==1.3.0

pip install -r https://raw.githubusercontent.com/Lasagne/Lasagne/master/requirements.txt

# Demo

## Instructions to run on data

The python scripts used to generate the Figure 3 in our paper can be found in the following folder

cd /TransferLearning/examples

python GBMLGG/GBMLGG\_ OS.py

python PRAD/PRAD\_PFI.py

python KIPAN/KIPAN\_DSS.py

python PanGyn/PanGyn\_DFI.py

The python scripts used to generate the Figure 4 in our paper can be found in the following folder

cd /TransferLearning/simulation

python PanGyn-DFI-5-MC-inequal.py

python PanGyn-DFI-5-noMC-inequal.py

python PanGyn-DFI-5-MC-equal.py

python PanGyn-DFI-5-noMC-equal.py

After the execution, the result will also be saved under the “Result” folder.

## Expected output

The output of each task will be a dataframe with 20 rows, in which each row contains an independent execution of 3 different learning schemas, mixture learning, independent learning, and transfer learning. The dataframe data can be used to generate Fig 3 and Fig 4.

# Instructions for Use

## How to run the software

To run our software with different diseases, endpoints, or feature, you need to download our dataset from a shared location (probably 10 GB) and put it under the TransferLearning/data/datasets/ folder. You can simply specify the task you want to run by passing the location of your interested dataset to the *read\_data* function.

## Reproduction instructions

The key point to reproduce the result in our paper is follow the configuration process strictly.

## Authors

Yan Gao, Yan Cui.

## License

This project is covered under the GNU General Public License (GPL).