**SardiNIA/inCHIANTI Age Rates Tutorial**

Eric Sun (updated 6/1/2018)

**Dependencies**

*Developed in Python 3.5 (Anaconda distribution)*

If independent Python 3.5, packages required:

* numpy
* scipy
* matplotlib
* pandas
* sklearn
* warnings, os, shutil, csv, math, random, time, subprocess
* OPTIONAL: WND5, xgboost

If Anaconda 3.5, packages required:

* OPTIONAL: WND5, xgboost

**Overview**

Intermediate code versions and experimental results can be found at our GitHub repository:

<https://github.com/sunericd/SardiNIAgeRates>

**runModel.py** [previously “BestModelsTurnkey.py”]

* Contains main functions for running the pipeline
* Optimizes sample and feature numbers
* Checks input parameters and returns warnings/errors
* Creates results files

**AgeRatesTurnkey.py** [previously “inCHIANTICommonTurnkey.py”

* Contains all the technical functions for:
  + Binning and cleaning the dataset
  + Feature selection
  + Training and testing the machine learning model
  + Deriving effective rates of aging

**run\_spec.txt**

* Tab separated text file for specifying model parameters:
  + First row: Descriptions of parameters (tab-separated)
  + Second row: Parameter values (tab-separated)
* “results\_dir” – relative path to directory for results saving (no “/” at end)
* “data\_dir” – relative path to data (no “/” at end)
* “data\_file” – name of data file to be read from “data\_dir” path
* “bin\_info” – binning information: bin size, start age, end age, train size, test size (comma separated) [SEE “BINNING STRATEGIES”]
* “num\_splits” – number of training-testing splits to run
* “model\_name” – specify model to run
  + “RandForClf” – random forest classifier
  + “elastic\_net” – elastic net regression
  + “lasso” – LASSO regression
  + “KNeighReg” – K-nearest neighbors regressor
  + “WND5” – WND5 (weighted distance) classifier
  + “linreg” – Linear regression
  + “SVR” – support vector machine regressor
  + “xgboost” – XGBoost gradient boosting algorithm
* “feature\_type” – type of features to use
  + “normal” – model-optimized set of features selected from all original features
  + “sardinia\_common” – 54 common clinical traits in SardiNIA dataset
  + “sardinia\_common\_cardio” – 67 common clinical or cardiovascular traits in SardiNIA

**Tutorial**

* Have “runModel.py”, “AgeRatesTurnkey.py”, and “run\_spec.txt” in the same working directory
* Input the correct values into the “run\_spec.txt” file
* Run in working directory with: python runModel.py
* Results will be saved in the specified results directory after the run is finished:
  + PNG image files of feature and sample saturation
  + PNG image files of predicted vs actual ages, distribution of aging rates by age
  + TSV file of results with columns: [ID] [Age] [Predicted Age] [Effective Rate of Aging]
  + Copy of “run\_spec.txt” for future reference
  + Text file with best feature numbers and LDA scores
  + PICKLE (python) file of predicted ages [WILL BE REMOVED SOON]

**Binning Strategies**

These were heuristically determined “best” binning strategies for SardiNIA and inCHIANTI and we recommend their usage for maximum data usage and age bin representation:

SardiNIA

* Normal
  + W4 BINS: 5, 21, 81, 49, 5
  + W3 BINS: 5, 20, 75, 80, 9
  + W2 BINS: 5, 16, 81, 83, 9
  + W1 BINS: 5, 12, 77, 120, 13
* Common + Cardio Features
  + W4 BINS: 5, 21, 81, 64, 7
  + W3 BINS: 5, 20, 75, 213, 24
  + W2 BINS: 5, 16, 81, 107, 12
  + W1 BINS: 5, 12, 77, 160, 18

inCHIANTI

* Normal
  + W3 BINS: 6, 31, 91, 20, 2
  + W2 BINS: 6, 27, 87, 19, 2
  + W1 BINS: 6, 26, 86, 19, 2
  + W0 BINS: 6, 23, 89, 19, 2