

Software documentation

These source codes are made available without any guarantee. They were written by Ka Yee Yeung on a linux box running Splus 6. Our experience is that they also work on solaris running Splus 6. In this documentation, we assume the readers are familiar with in S+ or R. If you would like to learn R, please visit <http://www.r-project.org/>.

Format of input files:

1. Training and test sets
 - Tab-delimited text files.
 - The first row is a header row, consisting of ordered gene (variable) names. We assume that the genes have been ordered using a univariate feature selection method, e.g. BSS/WSS (Dudoit et al. 2002), such that the best gene candidate is listed first.
 - No sample names in columns.
 - Each column corresponds to a sample (experiment).
 - Example: *t_selectedBSSWSS_ALL_AML_train_1_50.txt*
This is the training subset corresponding to $Y=0$ vs. $Y=1$ on the leukemia 3-class data. In this case, $p=50$ and the genes have been ordered by BSS/WSS ratios. There are a total of 30 samples in this training subset.
2. Class files (for both training and test sets)
 - A text file with one line for the class (Y) of each sample.
 - Example: *response_1_filtered_ALL_AML_train3051.txt*
This is the class file for the training subset *t_selectedBSSWSS_ALL_AML_train_1_50.txt*. There are 30 samples in this data subset, hence there are 30 lines in this file.

Iterative BMA algorithm

To uncompress the archive:

- On linux:
`tar -xzf binaryItBMA.tar.gz`
- On windows: WinZip can extract these files. There are probably other windows applications that can do it as well.

We will first describe the functions in each Splus file, and then we will describe how these functions can be used together.

IterateBMA.splus

- Implements the training step binary of the iterative BMA algorithm
- The idea is to use the training set to select genes (variables) and models for future prediction.
- In our empirical experiments, we ran into memory management problems with Splus. In particular, after we iteratively apply "bic.glm" many times, Splus will complain. To get around this problem, we used the parameter "maxIter" to limit the maximum number of times we apply "bic.glm". If we have to exit the while loop applying "bic.glm", we store the data matrix of the most recent selected genes in "myX.txt"

- “**iterateBMAinit**”: a function that produces a text file called “myX.txt” consisting of the top 30 genes from the input training set. This is the **initialization step** before calling “iterateBMA”.
- “**iterateBMA**”: a function that iteratively applies “bic.glm” to a moving window of at most 30 variables. (*updated 8/5/05*)
 - We might need to call this function many times until all p genes are processed. This is due to the memory management problems with Splus. Please refer to “sampleUseFunctions.splus” for an example.
 - Input parameters:
 - nbest: used in “bic.glm”. The leaps and bounds algorithm returns the best “nbest” models of each size (up to 30 variables). Recommended values: 20 or 50 from our empirical results.
 - maxNvar: size of moving window. Default is 30. Set to small numbers only if we have less than 30 samples in the training set.
 - myY: class vector for the training set
 - mysortedData: data matrix of the input data, assuming that the columns have been sorted by BSS/WSS ratios.
 - sortedFileName: file name of the data matrix “mysortedData”.
 - The selected genes are contained in the header of the output data matrix “final_myX_nbest<nbest>_thres1_<input training set>.txt”.
 - Eg. final_myX_nbest20_thres1_t_selectedBSSWSS_ALL_AML_train_1_50.txt
 - The number of genes explored is printed out in the Splus output file. “explored up to gene #”. Please see “sampleUseFunctions.out”.

ApplyIterateBMAprintRank.splus

- For the test phase of classification.
- Assume that we have called “iterateBMA” and have selected relevant genes in “final_myX_nbest<nbest>_thres1_<input training set>.txt”.
- “applyIterateBMAprintRank”: a function that reads in the selected genes, re-run “bic.glm” and predict the classes of the test set
- Input parameters:
 - inputTest:: file name of the test data
 - train.class: vector of the response variables (class, Y) for the training set
 - test.class: vector of the response variables (class, Y) for the test set
 - sortedGeneNameFile: text file containing gene names in sorted order from BSS/WSS
 - outItBMAfile: file containing the relevant genes. Output from “iterateBMA”.
- Output:
 - Predicted probabilities: in a text file. E.g. pred_final_myX_nbest20_thres1_t_selectedBSSWSS_ALL_AML_train_1_50.txt
 - Tabulated results: classes versus predicted classes
 - Brier score

Post_predict.splus

- To predict for the test set, using the regression parameters, and model posterior probabilities estimated from “bic.glm”
- Called in “applyIterateBMAprintRank”

PredictionScores.splus

- Computes the Brier Score

- Called in “applyIterateBMAprintRank”

RankSelectedGene.splus

- Given the selected genes and sorted gene names, return the BSS/WSS ranks.
- Called in “applyIterateBMAprintRank”

SampleUseFunctions.splus

- An example file running the subset $Y=0$ vs. $Y=1$ for the 3-class leukemia dataset.
- To run this file, “*Splus BATCH sampleUseFunctions.splus sampleUseFunctions.out*”
- Source all the functions in Splus before running.
- This directory contains all necessary files to run this sample file, except “bic.glm”, which can be downloaded from <http://www.research.att.com/~volinsky/software/bic.glm>.
- We have documented the Splus output file “*sampleUseFunctions.out.doc*”, and highlighted points of interest in yellow.
- For illustration purposes, we used $p=50$ in the example for computational efficiency. We would recommend using $p=1000$ or higher.

SampleUseFunctions_nbest100.splus

- An example file illustrating the *adaptive threshold step* on the subset $Y=0$ vs. $Y=1$ for the 3-class leukemia dataset.
- To run this file, “*Splus BATCH sampleUseFunctions_nbest100.splus sampleUseFunctions_nbest100.out*”
- Source all the functions in Splus before running.
- This directory contains all necessary files to run this sample file, except “bic.glm”, which can be downloaded from <http://www.research.att.com/~volinsky/software/bic.glm>.
- We have documented the Splus output file “*sampleUseFunctions_nbest100.out.doc*”, and highlighted points of interest in yellow.

Multi-class Iterative BMA algorithm

To uncompress the archive:

- On linux:
tar -xvzf multiltBMA.tar.gz
- On windows: WinZip can extract these files. There are probably other windows applications that can do it as well.

Unfortunately, we don’t have integrated software for the entire process at this point. Each step in the algorithm is described and illustrated with an example.

Outline of the multi-class iterative BMA approach

1. From the original training set consisting of all K classes ($K \geq 3$), choose a baseline class (e.g. AML in the case of the 3-class leukemia data). Create training subsets for each of ($Y=0$ vs. $Y=k$) where $k=1, 2, \dots, (K-1)$.
2. In the case of thresholded data (e.g. 3-class leukemia data), some genes may have identical values across all the samples within each training subset. Remove such genes. In the case of the 3-class leukemia data, there are 3041 genes (out of 3051) left for $Y=0$ vs. $Y=1$, and there are 3017 genes (out of 3051) left for $Y=0$ vs. $Y=2$.
3. For each of the training subset ($Y=0$ vs. $Y=k$), where $k=1, 2, \dots, (K-1)$:

- a. Rank the genes (variables) in using a univariate feature selection method (e.g. BSS/WSS ratio).
- b. Run the binary iterative BMA algorithm to select relevant genes.

* This step can be accomplished using the code in *binaryIterBMA.tar.gz*.

For example,

- The function “BssWss” in BSS_WSS.r can be used to compute the BSS/WSS ratios.
- To run the binary iterative BMA algorithm for **Y=0 vs. Y=1** and **Y=0 vs. Y= 2** on the 3-class leukemia data using $p=50$:
“*Splus BATCH sampleUseFunctions1.splus sampleUseFunctions1.out*”.
- Source “bic.glm” and “iterateBMA.splus” before running sampleUseFunctions1.splus.
“*iterateBMA.splus*” updated on 8/5/05.
- Output files:
 - final_myX_nbest20_thres1_t_selectedBSSWSS_ALL_AML_train_1_50.txt
data matrix and relevant genes for Y=0 vs. Y=1
 - final_myX_nbest20_thres1_t_selectedBSSWSS_ALL_AML_train_2_50.txt
data matrix and relevant genes for Y=0 vs. Y=2

4. Using the selected genes from each of the training subset (Y=0 vs. Y=k), create an augmented data matrix (described in Section 2.3 and Figure 1 in the manuscript).

This step can be accomplished using the R code “sortBssWssFinalFile.r”. We tested this code using R version 1.8.1 on linux. To run this step:

“*R CMD BATCH runSortBssWss.r runSortBssWss.out*”

Output files:

- merged_final_myX_nbest20_thres1_t_selectedBSSWSS_ALL_AML_train_50.txt: the resulting augmented data matrix
- merged_response_final_myX_nbest20_thres1_t_selectedBSSWSS_ALL_AML_train_50.txt: binary response variables
- unsorted_final_myX_nbest20_thres1_t_selectedBSSWSS_ALL_AML_train_50.txt
- combined_sorted_final_myX_nbest20_thres1_t_selectedBSSWSS_ALL_AML_train_50.txt

5. Run the iterative BMA algorithm on the augmented data matrix.

6. Using the selected genes from step 5, produce predicted probabilities using the function “applyIterateBMAprintRankMultiClass”.

* Steps 5 and 6 are illustrated in “sampleUseFunctions2.splus”:

For example,

- “*Splus BATCH sampleUseFunctions2.splus sampleUseFunctions2.out*”.
- Source “applyIterateBMAprintRankMultiClass.splus”,
“rankSelectedGeneMultiClass.splus” and “predictMultiClass.splus” before running sampleUseFunctions2.splus.
- Output files:
 - pred_final_myX_nbest20_thres1_merged_final_myX_nbest20_thres1_t_selectedBSSWSS_ALL_AML_train_50.txt
 - Brier Score and # classification errors in the Splus output file “sampleUseFunctions2.out”.
 - In this example, there are 3 classification errors on the test set and a generalized Brier Score = 3.249.

applyIterateBMAprintRankMultiClass.splus

- For the test phase of multi-class classification.
- Assume that we have called “iterateBMA” on the augmented data matrix
- “applyIterateBMAprintRankMultiClass”: a function that reads in the selected genes, re-run “bic.glm” and predict the classes of the test set
- Gene names selected in training subset ($Y=0$ vs. $Y=k$) have suffix “.k” where $k = 1, 2, \dots, K-1$
- Input parameters:
 - inputTest:: file name of the test data (full data)
 - train.class: vector of the binary response variables (class, Y) for the training set
 - test.class: vector of the multi-class response variables (class, Y) for the test set
 - sortedGeneNamePrefix: prefix of the text file containing gene names in sorted order from BSS/WSS
 - outItBMAfile: file containing the relevant genes. Output from “iterateBMA” on the augmented matrix.
 - numClass: number of classes
- Output:
 - Predicted probabilities: in a text file. E.g. pred_final_myX_nbest20_thres1_t_selectedBSSWSS_ALL_AML_train_1_50.txt
 - Tabulated results: classes versus predicted classes
 - Brier score

rankSelectedGeneMultiClass.splus

- Given the selected genes and sorted gene names, return the BSS/WSS ranks.
- If a gene belong to a different training subset, return -1.
- Called in “applyIterateBMAprintRankMultiClass.splus”

predictMultiClass.splus

- To predict for the test set, using the regression parameters, and model posterior probabilities estimated from “bic.glm”
- Also has a routine “genBrierScore” to compute the generalized Brier Score
- Called in “applyIterateBMAprintRankMultiClass.splus”

Note: Different OS and Splus versions may produce slightly different results. The genes and models selected may not be identical, but the Brier Scores should be close.