Consider the HIV drug resistance mutation data described on coursespace.

1. Load the data using the code given in slides #6.

*You need to download the helperfunction.R file from course space, specify correct work path so that R can find this file on your computer.*

*The code in example of slides #6 load data for only one drug, you need to run it multiple times to load data for all five drugs 3TC, ABC, AZT, D4T, DDI*

Update:

I believed everyone have encountered and solved many problems with processing the data.

Data manipulation and QC are “the” most important part of data analysis in real life applications. Once the data is ready in analysis-ready form, your analysis job is mostly done. Calling a few functions to analyze data and summarizing results can be much easier than data manipulation.

I hope you already get hand on experience on data manipulation. For this assignment, I do not want any students fail in the first step. So, I uploaded a new file **load.data.R**.

Running that file, you can load data, extract common columns in QC’ed mutation data, and common rows in drug IC50 data.

After you run that code, you will get two rda files, which contains XX (genetic mutation data) and YY (drug IC50 data). You can use R function “load” to read in the two matrixes.

1. Compare combination of the following method using 10-fold cross validation, and prepare 3 tables as in excel file.

One table for MSE defined as average of over all isolates and all drugs.

One table for average bias, defined as average of

The last table for variance defined as

is the cross validation predicted values, and is the observed values of IC50 ratio. The index i is for isolate, and index j is for drug.

*See R example/tutorial of how to conduct cross validation in slides #5. You can also refer to the R code provided by owner of data, see the link in data description posted on coursespace.*

Update:

As I posted in announcements and in-class discussions, I need to clarify slides #5 compared MSE of multiple methods using training-testing data approach. This is a simulation study, so you can obtain as many as samples you want. In this case, training-testing approach is preferred. But in our assignment, we have limited data, hence you need to change training-testing approach to cross-validation approach.

The Stanford website used function “train” to implicitly conducted cross validation. But that function might not work well for stacking approach. Hence you need to explicitly implement CV.

Please refer to “[Example Code for a few classification methods discussed in classFile](https://coursespaces.uvic.ca/mod/resource/view.php?id=1054410)”, and “[More tutorial for CVFile](https://coursespaces.uvic.ca/mod/resource/view.php?id=1055256)” that I posted on coursespace (Oct11-17) to see example I used cross validation to compare lm and LASSO.

I found some students still confused about stacking, I will give more examples tomorrow in class.

1. Comments on comparison results you got. If you could try to explain some finding using theoretical calculations. Theoretical calculation part is bonus. Following are questions I expect you to answer from your comparison results:
   1. Which regression methods consistently works better than others?
   2. Does stacking really always do something? When not, try to figure out why
   3. Does improved stacking really improve performance of predictions? Try to answer why
   4. Does number of predictors affect your answer above? (i.e. complete mutation data vs expert mutation data)
   5. Anything else you’d like to discuss from your finding.
2. Design additional tables/figures to present your results better when needed.

The assignment will be marked according to “presentation is clear and precise”, “results are correct and reproducible”, “Code are well written”, “Novelty and quality in your optional work”

Here are some suggestions:

1. Describe your findings in the beginning (like abstract of paper), and then give the details.
2. Write your code well (try to use functions to module your code, avoid using global variables if possible) with comments to explain what you are doing.
3. Results are reproducible by just running your code. Use

set.seed(10, kind = "Mersenne-Twister", normal.kind = "Inversion")

1. I suggest you to use R markdown, but it is optional.