BIOLOGY 664: Integrated Bioinformatics Using R for Both Wet and Dry Scientists

Problem Set 6: Microarray Analysis Using data.frames

Due: At the beginning of class Thursday, April 9th.

Chapter 6 Exercises Using data.frames

Answer questions 1, 2, 4, in the Chapter 6 Exercises using a data.frame for every answer. You should never refer to a matrix in your solutions. Also, create named rows and named columns, and use named indexing whenever possible.

At the very least, you need to hand in an R Markdown File (.Rmd) that logically delineates and presents you R code as well as the R output and plots generated by your code for each exercise.

Hardcopy

Also, in addition to submitting a softcopy through blackboard I'd also like to get a printed hardcopy of your Problem Set. Please print double-sided and use Arial font size 7 or 8 in order to save trees.

Extra Credit

I'll be giving extra credit for those Problem Sets that use Latex (pronounced “lay-tech)”. I'm also going to give extra credit for plots that have been "fancied up" to produce more publication-like figures. These plotting enhancements include colors, titles, legends, axis labels, p-values, and R^2s. Being able to create publication-quality figures in R is an important skill, and this is an opportunity to earn extra points while learning to do so. Look at the "Quick-R" and "Producing Simple Graphs with R" links on the online syllabus for good online references for how to fancy-up R plots.

Important notes and hints:

1. Use ***ALL$BT %in% c("B","B1","B2","B3","B4")*** to get the patients in just those stages.
2. To setup your all.df data.frame from the ALL ExprSet do the following:
    a. **all.df = data.frame(ALL)**
    b. **names(all.df)[1:length(featureNames(ALL))] = featureNames(ALL)**
        i. Gets rid of the X's that are prepended in front of integers
    c. ***all.df[all.df == "NA"] = NA***
        i. replaces "NA" strings in the data with real NAs.
    d. Now use your all.df data.frame to answer the questions (do not use the ALL ExpSet)
3. Use ***grep()*** to get only the columns that contain expression data, ***has.expression = grep("_at$|_st$",names(allb.df))*** gets rid of columns that have non-expression data (age, gender, remission status etc.). ("$" in the regular expression signifies end of string and “|” is logical “or”. Therefore, with this regex we’re grabbing only the columns that end in “_at” or “_st”.)

4. Also, ***has.expression = grep("_at$|_st$",names(allb.df), invert=TRUE)*** uses ***invert=TRUE*** to get the columns that DON'T match the regular expression.
5. You can work together, but all your written work (including R code) must be your own.