BMI 551/651 Final Data Challenge

Kristen Stevens

March 3, 2016

Goals

Predict whether a breast cancer cell line will respond to treatment with a given drug using the subtype of the tumor and the gene expression data provided. Submissions can either be binary or contain values between 0 and 1 so that the area under the ROC curve can be computed for different threshold cutoffs between 0 and 1.

Response in this context means that the concentration of drug needed to inhibit cell growth by 50% was above the median for all cell lines tested (not just those used above). There is a lot to be said about whether this measure reflects how the drug will work in patients.

One important aspect of this type of challenge is to ascertain whether the data is sufficient to make meaningful inferences.

Data

The data used comes from this study below by Dr. Joe Gray, Dr. Laura Heiser and many others of whom are here at OHSU: Anneleen Daemen et al., "Modeling Precision Treatment of Breast Cancer," Genome Biology 14, no. 10 (2013): R110, doi:10.1186/gb-2013-14-10-r110.

There are 25 cell lines and 12 drugs in the training set. The responses are coded as 0 = cell line doesn't respond to drug, 1 = cell line does respond to drug. (This data originally had 70 cell lines and 90 drugs, but in order to avoid issues with missing data we have restricted the challenge to those data seen here. It is generally too small to be of use in a real study.)

You are provided with:

1) expression.txt - a tab-delimited text file containing expression values for 18,632 genes for each of the 39 cell lines.

2) subtypes.txt - a tab-delimited text file of subtypes (basal, luminal, claudin-low and normal-like) for each of 39 cell lines.

3) training_set_answers.txt - a tab-delimited text file of the correct classification of 0 (non-responsive) or 1 (responsive) for each combination of 25 cell lines and 12 drugs.

4) scoring_and_test_set_id_mappings.csv - a comma-delimited text file of the id used by Kaggle for each of the cell line/drug combinations in the scoring set and test set. The first 108 values are the scoring set (9 cell lines and 12 drugs) and the last 60 are the final test set (5 cell lines 12 drugs). Scores on the final test set will not be shown until the competition is over.

5) rand_sub_cont.csv - a sample submission file in the correct format with random predictions between 0 and 1. The calculation of the AUROC value summarizes the performance of these guesses at all thresholds between 0 and 1.

Exploratory Data Analysis

```r
knitr::opts_chunk$set(fig.path = "Figs/", message = FALSE, warning = FALSE, echo = TRUE, error = TRUE,
```

```r
library(dplyr)
library(plyr)
library(ggplot2)
library(psych)
library(GGally)
library(ggfortify)
library(gsl)
library(MASS)
library(MBESS)
library(broom)
```

```r
expression <- read.table(file = "/Users/kstevensvt/bmi_551_651_final/expression.txt", sep = "\t")
subtypes <- read.table(file = "/Users/kstevensvt/bmi_551_651_final/subtypes.txt", header=TRUE, sep = "\
answers <- read.table(file = "/Users/kstevensvt/bmi_551_651_final/training_set_answers.txt", sep = "\t"
kaggleID <- read.table(file = "/Users/kstevensvt/bmi_551_651_final/scoring_and_test_set_id_mappings-2.c
sample <- read.table(file = "/Users/kstevensvt/bmi_551_651_final/rand_sub_cont.csv", header=TRUE, sep =
```

```r
dim(expression)
```

```
[1] 6427   39
```

```r
names(expression)
```

```
 [1] "X184A1"      "X600MPE"     "AU565"       "BT474"       "CAMA1"
 [6] "HCC70"       "HCC1143"     "HCC1187"     "HCC1395"     "HCC1419"
[11] "HCC1428"     "HCC1806"     "HCC1937"     "HCC1954"     "HCC2185"
[16] "HCC3153"     "HS578T"      "LY2"         "MCF12A"      "MCF10F"
[21] "MCF7"        "MDAMB134VI"  "MDAMB157"    "MDAMB175VII" "MDAMB231"
[26] "MDAMB361"    "MDAMB415"    "MDAMB453"    "SKBR3"       "SUM52PE"
[31] "SUM149PT"    "SUM159PT"    "SUM185PE"    "SUM1315MO2"  "T47D"
[36] "ZR751"       "ZR75B"       "BT549"       "MCF10A"
```

```r
rownames(expression)[1:10]
```

```
 [1] "C9orf152" "ELMO2"    "RPS11"    "CREB3L1"  "PNMA1"    "MMP2"
 [7] "C10orf90" "ERCC5"    "ZHX3"     "GPR98"
```

```r
dim(subtypes)
```

```
[1] 39  2
```

```r
head(subtypes)
```

```
  cellline     subtype
1    184A1 Normal-like
2   600MPE     Luminal
```

```
3    AU565     Luminal
4    BT474     Luminal
5    BT549 Claudin-low
6    CAMA1     Luminal
```

```
dim(answers)
```

```
[1] 25 12
```

```
names(answers)
```

```
 [1] "CGC.11047"    "Carboplatin"  "Cisplatin"    "GSK1070916"
 [5] "GSK1120212"   "GSK461364"    "Geldanamycin" "Oxaliplatin"
 [9] "PF.3084014"   "PF.3814735"   "PF.4691502"   "Paclitaxel"
```

```
rownames(answers)[1:10]
```

```
 [1] "CAMA1"      "ZR751"      "HCC1419"    "184A1"      "HCC1428"
 [6] "SUM52PE"    "SUM149PT"   "MDAMB134VI" "HCC70"      "SUM1315MO2"
```

```
dim(kaggleID)
```

```
[1] 168   4
```

```
head(kaggleID)
```

```
  cellline      drug id  Usage
1  HCC1187 CGC-11047  1 Public
2     MCF7 CGC-11047  2 Public
3 MDAMB361 CGC-11047  3 Public
4 MDAMB231 CGC-11047  4 Public
5    BT549 CGC-11047  5 Public
6   600MPE CGC-11047  6 Public
```

```
dim(sample)
```

```
[1] 168   2
```

```
head(sample)
```

```
  id     value
1  1 0.9638433
2  2 0.7745915
3  3 0.2088763
4  4 0.3087868
5  5 0.9713425
6  6 0.5849001
```

There are only 6427 observation in expression.txt, not 18,632 genes. Remaining data imported as expected.

```r
sum(is.na(expression))
```

```
[1] 39
```

```r
sapply(expression, function(x)  sum(is.na(x)))
```

```
     X184A1     X600MPE       AU565       BT474       CAMA1       HCC70 
          1           1           1           1           1           1 
    HCC1143     HCC1187     HCC1395     HCC1419     HCC1428     HCC1806 
          1           1           1           1           1           1 
    HCC1937     HCC1954     HCC2185     HCC3153      HS578T         LY2 
          1           1           1           1           1           1 
     MCF12A      MCF10F        MCF7  MDAMB134VI    MDAMB157 MDAMB175VII 
          1           1           1           1           1           1 
   MDAMB231    MDAMB361    MDAMB415    MDAMB453       SKBR3     SUM52PE 
          1           1           1           1           1           1 
   SUM149PT    SUM159PT    SUM185PE  SUM1315MO2        T47D       ZR751 
          1           1           1           1           1           1 
      ZR75B       BT549      MCF10A 
          1           1           1 
```

```r
sum(is.na(subtypes))
```

```
[1] 0
```

```r
sum(is.na(answers))
```

```
[1] 0
```

```r
sum(is.na(kaggleID))
```

```
[1] 0
```

```r
sum(is.na(sample))
```

```
[1] 0
```

There are no missing data except for expression data for 1 gene for each cell line.

```r
t_expression <- data.frame(t(expression)) # code does not work
```

```
Error: C stack usage  8200352 is too close to the limit
```

```r
full <- join(subtypes, t_expression)
```

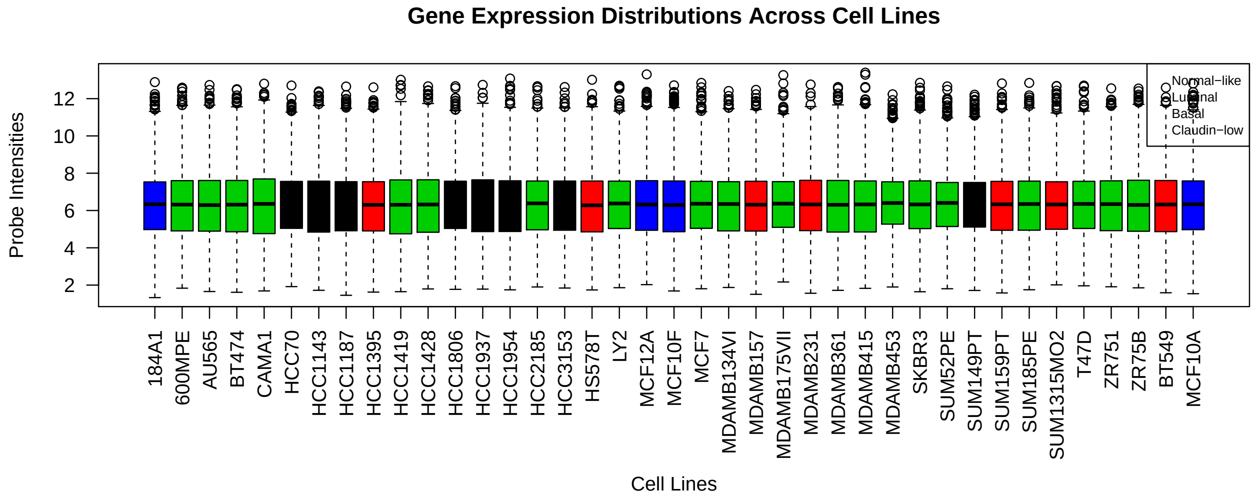
```
Error in as.vector(y): object 't_expression' not found
```

Tried to add subtype and drug response to expression data but failed.

```
subtypes <- subtypes[with(subtypes, order(subtype)),]
colnames(expression)[1] <- "184A1"
colnames(expression)[2] <- "600MPE"
rownames(subtypes) <- subtypes[,1]
subtypes[colnames(expression),2]
```

```
 [1] Normal-like Luminal     Luminal     Luminal     Luminal
 [6] Basal       Basal       Basal       Claudin-low Luminal
[11] Luminal     Basal       Basal       Basal       Luminal
[16] Basal       Claudin-low Luminal     Normal-like Normal-like
[21] Luminal     Luminal     Claudin-low Luminal     Claudin-low
[26] Luminal     Luminal     Luminal     Luminal     Luminal
[31] Basal       Claudin-low Luminal     Claudin-low Luminal
[36] Luminal     Luminal     Claudin-low Normal-like
Levels: Basal Claudin-low Luminal Normal-like
```

```
boxplot(expression, main = "Gene Expression Distributions Across Cell Lines", ylab = "Probe Intensities
mtext("Cell Lines", side = 1, line = 7)
legend("topright", legend = unique(subtypes[colnames(expression),2]), cex = 0.7)
```



Reorder cell lines so they are grouped by subtype. Add legend.

Make another boxplot, this time group by 25 training, 14 test displayed, 5 test hidden (if possible, keep color coding for subtypes). Report number of each subtype within each of the 3 groups.

We should split training set into training and validation sets.

subtypes <- subtypes[with(subtypes, order(subtype)),]