AIDS Antiviral Screen Data

Steven Burnett, Ph.D. http://www.linkedin.com/in/sburnettchemistry

February 3, 2016

Summary

The goal of this project is to use machine learning to predict the results of Development Therapeutics Program (DTP) AIDS Antiviral Screening data that is publicly available. Namely, the DTP provides services and data to facilitate the discovery and development of new therapeutic agents. Moreover, this report documents both the predictive power of the model and all corresponding data used for the analysis in an effort to identify effective therapeutic agents for treating patients with AIDS and/or cancer.

The NIH website hosting the data is located at https://wiki.nci.nih.gov/display/NCIDTPdata/AIDS+Antiviral+Screen+Data, and the May 2004 data can be found at https://wiki.nci.nih.gov/pages/viewpageattachments.action?pageId=158204006&metadataLink=true. The target classes for prediction are:

- CA - Confirmed active
- CM - Confirmed moderately active
- CI - Confirmed inactive

Using a rule-based algorithm (C5.0) and cross-validation, the statistics of the model chosen has an accuracy of 96.23% (95% CI of .9589- .9656), and a Kappa value of .7645. Lastly, the analysis uncovered that high EC50 concentrations can be used to indicate a drug's potency and classification.

Gathering and Cleaning the Data

The data:

Because the results of the screening (classes) were provided separately from the rest of the predictors. The data was joined by the variable NSC (the NCI's internal ID number) to create a new data frame for the analysis.

```
#Screening Results
#download.file("https://wiki.nci.nih.gov/download/attachments/158204006/aids_conc_may04.txt?version=1&m

# concentrations necessary to see a protective effect on the infected cells
#download.file("https://wiki.nci.nih.gov/download/attachments/158204006/aids_ec50_may04.txt?version=1&m

#concentrations necessary to inhibit the growth of uninfected cells
#download.file("https://wiki.nci.nih.gov/download/attachments/158204006/aids_ic50_may04.txt?api=v2","ai

aids_classes <- read.csv("aids_conc_may04.txt", stringsAsFactors = F,strip.white = T)
aids_inhiit <- read.csv("aids_ic50_inhibit_may04.txt", stringsAsFactors = F)
aids_protective <- read.csv("aids_ec50_protective_may04.txt", stringsAsFactors = F)
aids_df <- Reduce(function(...) merge(..., by="NSC"), list(aids_classes, aids_inhiit, aids_protective))
```

From the information on NIH's website, some of the features will need to be coded as factors:

1. ConcUnit

- M = molar
- u = micrograms per ml
- V = Volumetric

2. Flag
   - > indicates EC50 would be higher than the highest concentration tested in at least one of the experiments
   - < indicates EC50 would be less than the lowest concentration tested in at least one of the experiments
   - = indicated that all the experiments reached EC50

```
aids_df$Conclusion<- as.factor(aids_df$Conclusion);
aids_df$Flag.x <- as.factor(aids_df$Flag.x);aids_df$Flag.y <- as.factor(aids_df$Flag.y);
rm(aids_classes,aids_inhiit,aids_protective);
aids_df$ConcUnit.x <- as.factor(aids_df$ConcUnit.x);
aids_df$ConcUnit.y <- as.factor(aids_df$ConcUnit.y)
head(aids_df);summary(aids_df)
```

```
##   NSC Conclusion Log10HiConc.x ConcUnit.x Flag.x Log10IC50 NumExp.x
## 1  48         CI          -3.7          M      =     -4.61        3
## 2  78         CI          -3.7          M      =     -4.02        3
## 3 128         CI          -3.7          M      =     -4.34        4
## 4 164         CI          -3.7          M      >     -3.70        2
## 5 180         CI          -2.7          M      =     -3.16        2
## 6 186         CI          -3.8          M      =     -4.60        2
##   StdDev.x Log10HiConc.y ConcUnit.y Flag.y Log10EC50 NumExp.y StdDev.y
## 1     0.02          -3.7          M      >     -3.70        3        0
## 2     0.09          -3.7          M      >     -3.70        3        0
## 3     0.06          -3.7          M      >     -3.70        4        0
## 4     0.00          -3.7          M      >     -3.70        2        0
## 5     0.07          -2.7          M      >     -2.74        2        0
## 6     0.02          -3.8          M      >     -3.84        2        0
```

```
##       NSC         Conclusion Log10HiConc.x    ConcUnit.x Flag.x
##  Min.   :    48   CA: 1809   Min.   :-9.700   M:42230    <: 1056
##  1st Qu.:601986   CI:39103   1st Qu.:-4.000   u:  686    =:20160
##  Median :650291   CM: 2011   Median :-3.700   V:    7    >:21707
##  Mean   :541787              Mean   :-3.873
##  3rd Qu.:674608              3rd Qu.:-3.700
##  Max.   :722245              Max.   : 5.600
##    Log10IC50          NumExp.x          StdDev.x       Log10HiConc.y
##  Min.   :-11.000   Min.   :   1.000   Min.   :0.00000   Min.   :-9.700
##  1st Qu.: -4.940   1st Qu.:   2.000   1st Qu.:0.00000   1st Qu.:-4.000
##  Median : -4.170   Median :   2.000   Median :0.00000   Median :-3.700
##  Mean   : -4.426   Mean   :   4.008   Mean   :0.04626   Mean   :-3.875
##  3rd Qu.: -3.700   3rd Qu.:   2.000   3rd Qu.:0.04000   3rd Qu.:-3.700
##  Max.   :  5.640   Max.   :9195.000   Max.   :2.34000   Max.   : 5.600
##  ConcUnit.y Flag.y      Log10EC50          NumExp.y
##  M:42229    <:  174   Min.   :-10.150   Min.   :  1.000
##  u:  687    =: 2000   1st Qu.: -4.220   1st Qu.:  2.000
##  V:    7    >:40749   Median : -3.700   Median :  2.000
##                       Mean   : -3.988   Mean   :  2.416
##                       3rd Qu.: -3.700   3rd Qu.:  2.000
```

```
##                       Max.   :  5.600   Max.   :261.000
##     StdDev.y
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.03972
##  3rd Qu.:0.00000
##  Max.   :2.38000
```

Because of the similarities of the screenings, a check was done to identify if the log variables were highly correlated, or possibly identical.

```
data.frame(tail(aids_df$Log10HiConc.x,20),tail(aids_df$Log10HiConc.y,20))#can drop one
```

```
##    tail.aids_df.Log10HiConc.x..20. tail.aids_df.Log10HiConc.y..20.
## 1                             -3.7                            -3.7
## 2                             -3.7                            -3.7
## 3                             -3.7                            -3.7
## 4                             -3.7                            -3.7
## 5                             -3.7                            -3.7
## 6                             -3.7                            -3.7
## 7                             -3.7                            -3.7
## 8                             -3.7                            -3.7
## 9                             -3.7                            -3.7
## 10                            -3.7                            -3.7
## 11                            -3.7                            -3.7
## 12                            -3.7                            -3.7
## 13                            -3.7                            -3.7
## 14                            -3.7                            -3.7
## 15                            -3.7                            -3.7
## 16                            -3.7                            -3.7
## 17                            -3.7                            -3.7
## 18                            -3.7                            -3.7
## 19                            -3.7                            -3.7
## 20                            -3.7                            -3.7
```

```
data.frame(tail(aids_df$Log10IC50,20),tail(aids_df$Log10EC50,20))# must keep both
```

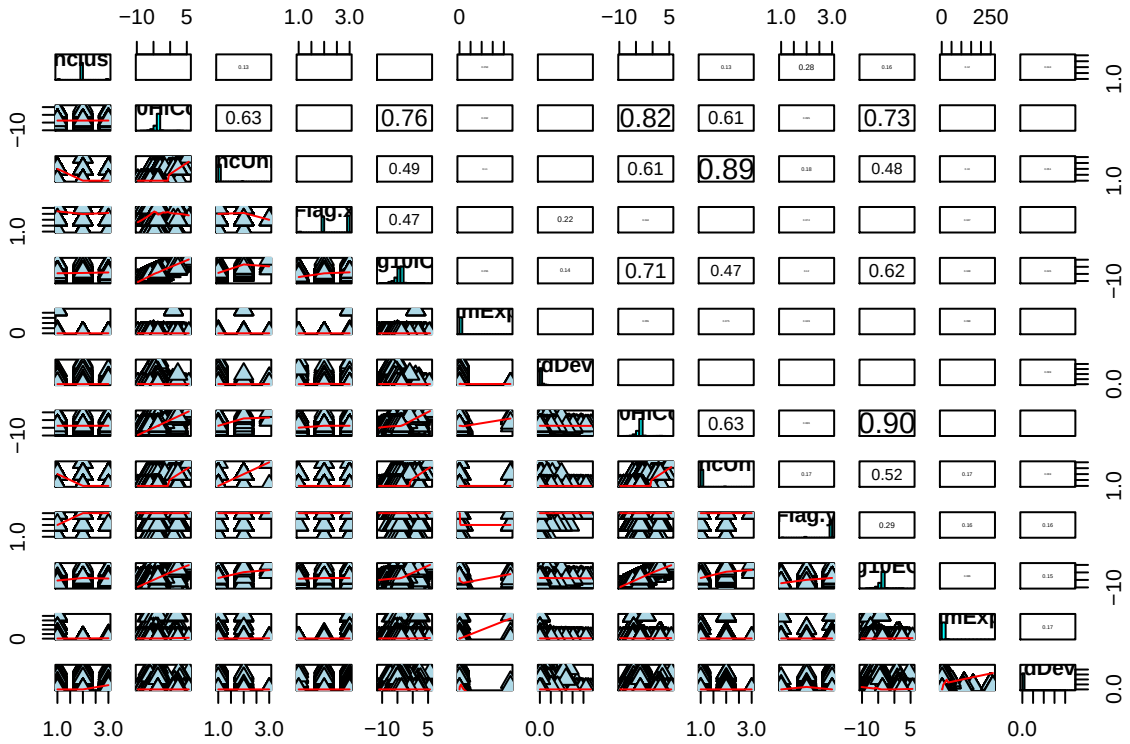
```
##    tail.aids_df.Log10IC50..20. tail.aids_df.Log10EC50..20.
## 1                        -3.70                       -3.70
## 2                        -4.91                       -3.70
## 3                        -3.70                       -3.70
## 4                        -4.43                       -3.70
## 5                        -4.48                       -3.70
## 6                        -4.46                       -3.70
## 7                        -3.70                       -3.70
## 8                        -3.70                       -3.70
## 9                        -3.70                       -3.70
## 10                       -3.70                       -3.70
## 11                       -4.97                       -3.70
## 12                       -4.97                       -3.70
## 13                       -4.96                       -4.48
```

```
## 14                      -3.70                      -3.70
## 15                      -3.74                      -3.70
## 16                      -3.70                      -3.70
## 17                      -4.44                      -3.70
## 18                      -5.32                      -3.70
## 19                      -4.49                      -3.70
## 20                      -3.70                      -3.70
```

```
panel.hist <- function(x, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}


pairs(aids_df[-1], panel = panel.smooth,
      cex = 1.5, pch = 24, bg = "light blue",
      diag.panel = panel.hist, upper.panel = panel.cor,cex.labels = 1, font.labels = 2)
```



It is apparent that the log10 High Concentrations for X and Y are identical, and only one will be kept. However, log IC50 and EC50 are different and both must be kept.

A plot of the relationships between all of the variables was made, having the correlations in the upper triangle, scatter plots in the lower triangle, and variable names and distributions on the main diagonal. Based on the summary, and visually inspecting HiConc, IC50 and EC50 variables, only the HiConc.y variable can be removed. Additionally, the variables NSC, stddev, numexp will also get removed as they are unnecessary for the prediction.

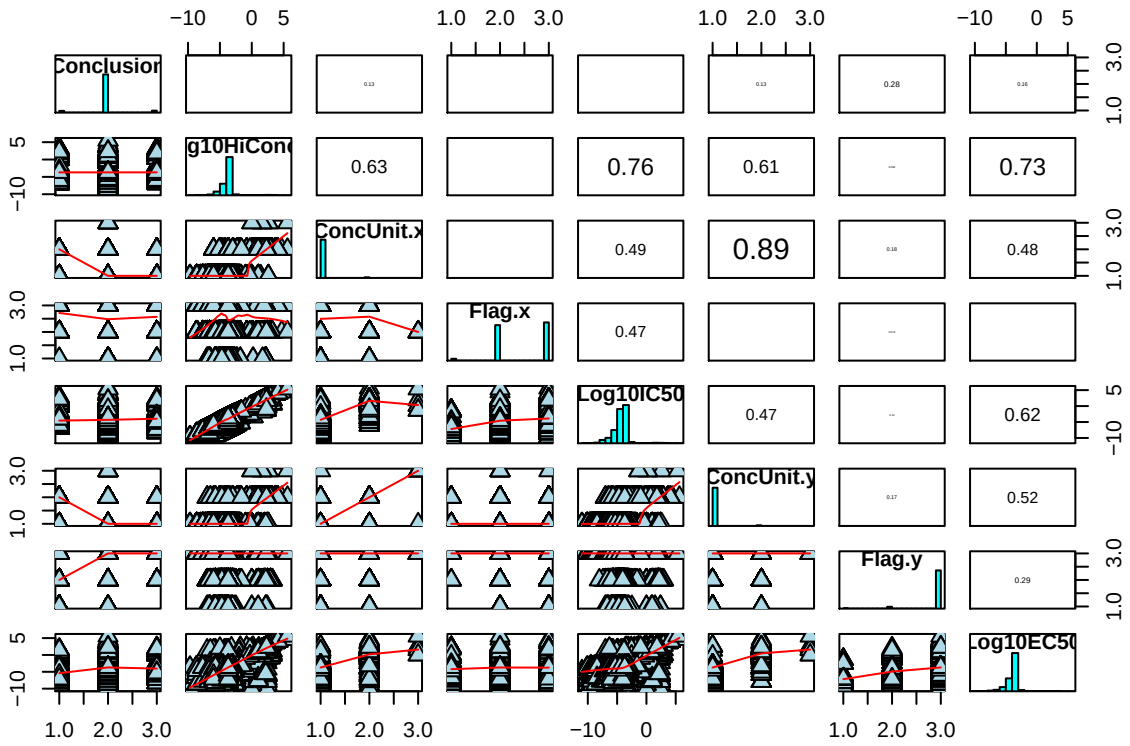
Next, the variables StdDev.x, StdDev.y, NumExp.x, NumExp.y, NSC, and Log10HiConc.y were selected, removed, and a new relationship plot was made.

```
suppressMessages(library(dplyr))
aids_df<- aids_df %>% select(c(-StdDev.x,-StdDev.y,-NumExp.x,-NumExp.y,-NSC,-Log10HiConc.y))
panel.hist <- function(x, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
```

```
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}


pairs(aids_df, panel = panel.smooth,
      cex = 1.5, pch = 24, bg = "light blue",
      diag.panel = panel.hist, upper.panel = panel.cor,cex.labels = 1, font.labels = 2)
```



The plot shows that the concentrations units are correlated, but that is to be expected and does not represent a real issue.

Model Selection and Tunning

Because the conclusion data was determined by inspection of individual dose response curves by subject matter experts (SME) in this field, the results correspond to their overall judgment. Furthermore, the EC50 and IC50 data are computer generated averages and don't necessarily capture everything that was considered when making the judgment. Therefore, a rule-based system was chosen to aid the SMEs for predicting future classes, based on the nature of the Flag variable and what it represents.

The C5.0 rule-set was selected because of its scalability and it's known performance with generating rules that are easy to interpret. http://perun.pmf.uns.ac.rs/radovanovic/dmsem/cd/install/Weka/doc/classifiers-papers/trees/ADTree/atrees.pdf

```
suppressMessages(library(caret))
suppressMessages(library(doMC))
registerDoMC(cores=7)
set.seed(323)
M.intrain<- createDataPartition(aids_df$Conclusion,p=.70,list = F)
```

```
M.training <- aids_df[M.intrain,]
M.testing <- aids_df[-M.intrain,]
cvCtrl <- trainControl(method = "repeatedcv", number = 10, repeats=3, allowParallel = T)
```

The data are split 70-30 to create the training and testing sets, respectively. In addition, 10 fold cross-validation was selected with repeats to train the model on the training data, thus reducing the out of sample error.

```
fit_M.c5 <- train(Conclusion~., method="C5.0Rules", data = M.training, trControl= cvCtrl, metric= "Accu
```

```
## Loading required package: C50
```

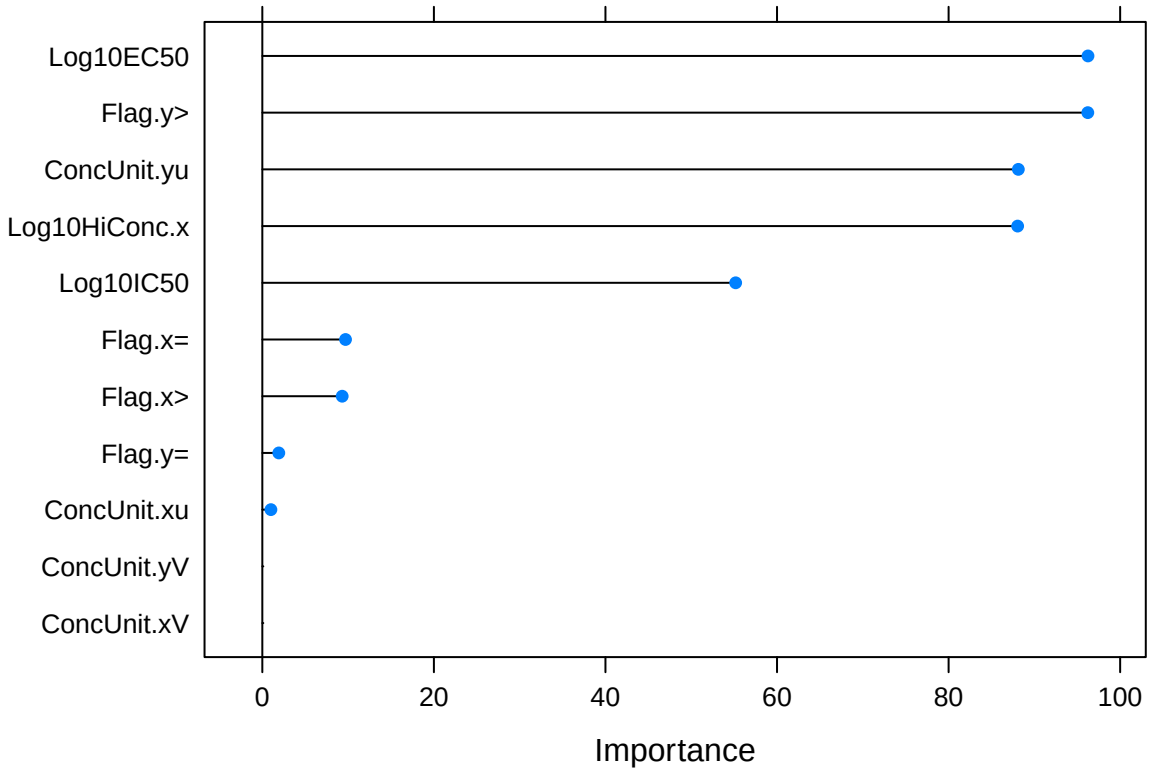
Training Data

Classification analysis from the training data was utilized to identify the important features, and predict with high accuracy the classification of the screening results. The tuning parameters were selected automatically based on repeated cross-validation to evaluate the models accuracy.

```
getTrainPerf(fit_M.c5)
```

```
##   TrainAccuracy TrainKappa    method
## 1     0.9608626   0.752214 C5.0Rules
```

```
plot(varImp(fit_M.c5,scale = F))
```



```
predictions.c5<- predict(fit_M.c5,M.training)
print(confusionMatrix(predictions.c5,M.training$Conclusion))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    CA    CI    CM
##         CA  1051    68    92
##         CI   102 27070   433
##         CM   114   235   883
##
## Overall Statistics
##
##                Accuracy : 0.9653
##                  95% CI : (0.9631, 0.9673)
##     No Information Rate : 0.911
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7821
##  Mcnemar's Test P-Value : 1.239e-14
##
## Statistics by Class:
##
##                      Class: CA Class: CI Class: CM
## Sensitivity            0.82952    0.9889   0.62713
## Specificity            0.99444    0.8000   0.98781
## Pos Pred Value         0.86788    0.9806   0.71672
## Neg Pred Value         0.99251    0.8760   0.98178
## Prevalence             0.04217    0.9110   0.04686
## Detection Rate         0.03498    0.9009   0.02939
## Detection Prevalence   0.04030    0.9187   0.04100
## Balanced Accuracy      0.91198    0.8945   0.80747
```

Based on the plot, the top 5 variables of importance are Log10EC50, Flag.EC50= >, ConcUnit.EC50= u, Log10HiConc.IC50, and Log10IC50. The plot also suggests that when the EC50 concentration is highest than the highest concentration tested, it will have a substantial impact on the classification on the antiviral screening. Moreover, the cross-validated model has an **accuracy of 0.9653 (96.53%)** and a **kappa value of .7821**.

Testing Data

The final model was built from the training set as selected by a majority voting scheme to evaluate its accuracy. Next, that final model was then assessed using a testing set to determine its true accuracy.

```
predictions.c5<- predict(fit_M.c5,M.testing[-1])# removing the class column
print(confusionMatrix(predictions.c5,M.testing$Conclusion))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    CA    CI    CM
```

```
##         CA   446    49    58
##         CI    47 11586   187
##         CM    49    95   358
##
## Overall Statistics
##
##                Accuracy : 0.9623
##                  95% CI : (0.9589, 0.9656)
##     No Information Rate : 0.9111
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7645
##  Mcnemar's Test P-Value : 9.308e-07
##
## Statistics by Class:
##
##                      Class: CA Class: CI Class: CM
## Sensitivity            0.82288    0.9877   0.59370
## Specificity            0.99132    0.7956   0.98827
## Pos Pred Value         0.80651    0.9802   0.71315
## Neg Pred Value         0.99221    0.8635   0.98020
## Prevalence             0.04210    0.9111   0.04683
## Detection Rate         0.03464    0.8999   0.02781
## Detection Prevalence   0.04295    0.9181   0.03899
## Balanced Accuracy      0.90710    0.8917   0.79098
```

The model was applied to the testing set, and the prediction accuracy deviates only slightly from the training set with an **accuracy of 0.9623 (96.23%)** and a **kappa value of .7645**

Concluding Remarks

The analysis from this report presented an accurate model with the aim of giving the SMEs in this field a better method for identifying the active class other than from their collective agreement.