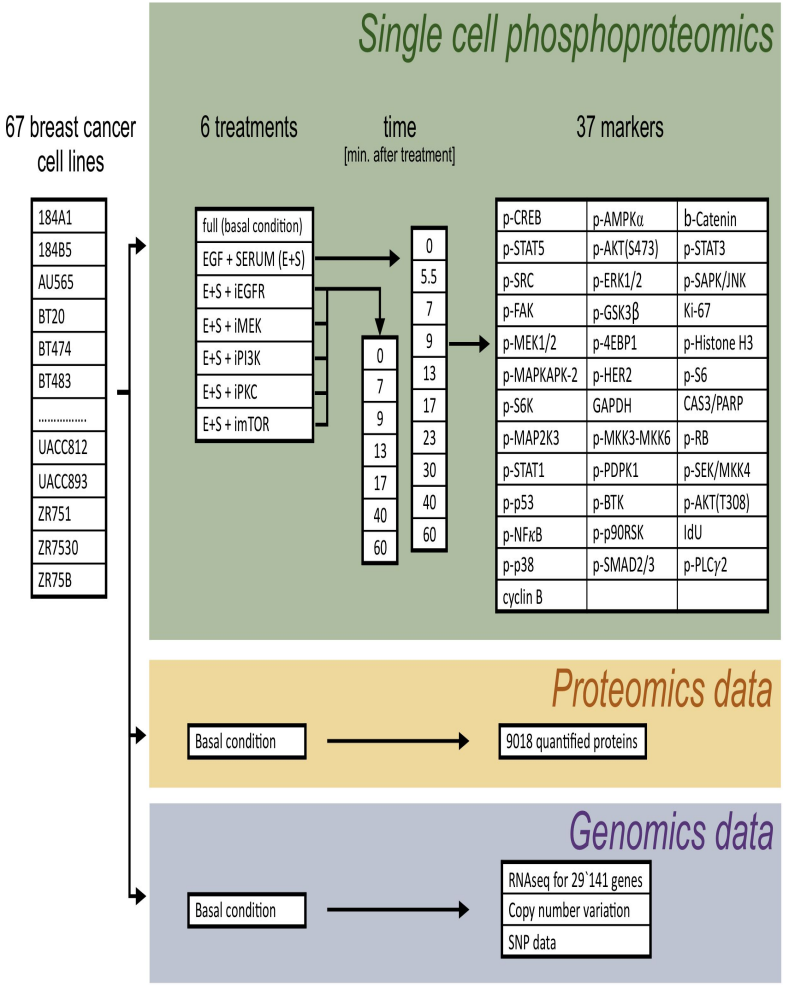
DREAM: Single-Cell Signaling in Breast Cancer

Dimitrius Raphael

NYU

Recap

- DREAM Challenge Seeking to analyze cell line specific signaling response.
- Single cell Phosphoprotemomics data, as well as population-based measurements
- Composed of 4 Subchallenges
  - **Predict Missing Markers (I)**
  - Predict Kinase inhibitor response (II and III)
  - Predict Cell-line response (IV)



Datasets

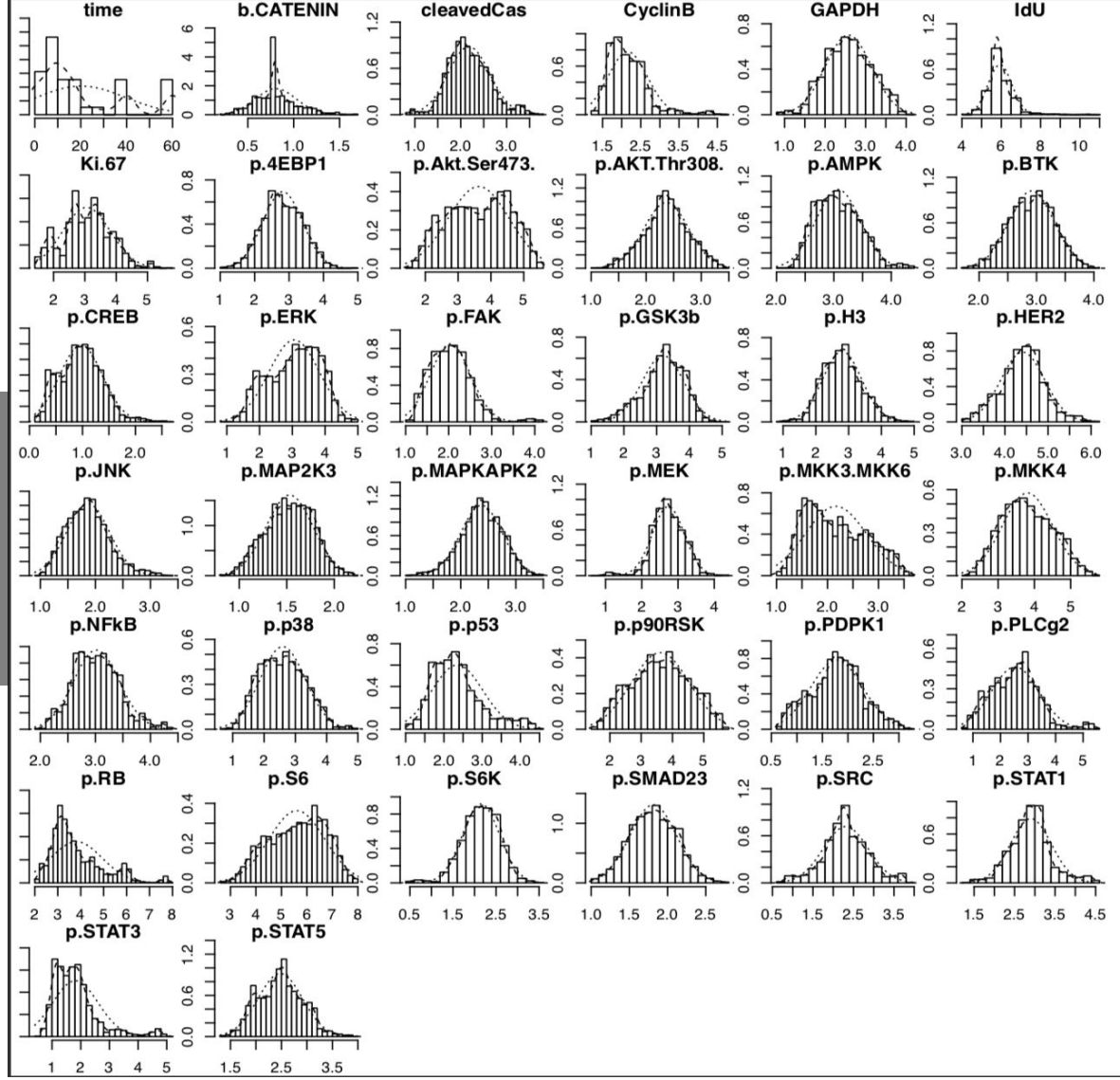
- Given CSV files of single cell data for each sub-challenge variable of interest (i.e AU565 cell-line in sub-challenge I lacking marker data)
  - Data under particular experimental conditions for different time points
  - Some NA values within data-Decide whether to impute or omit those rows
- Also given .CSV files for each complete cell line
  - Population-based measurement files
  - CSV files for RNA-Seq data, CNV data, SNP data, and proteomics

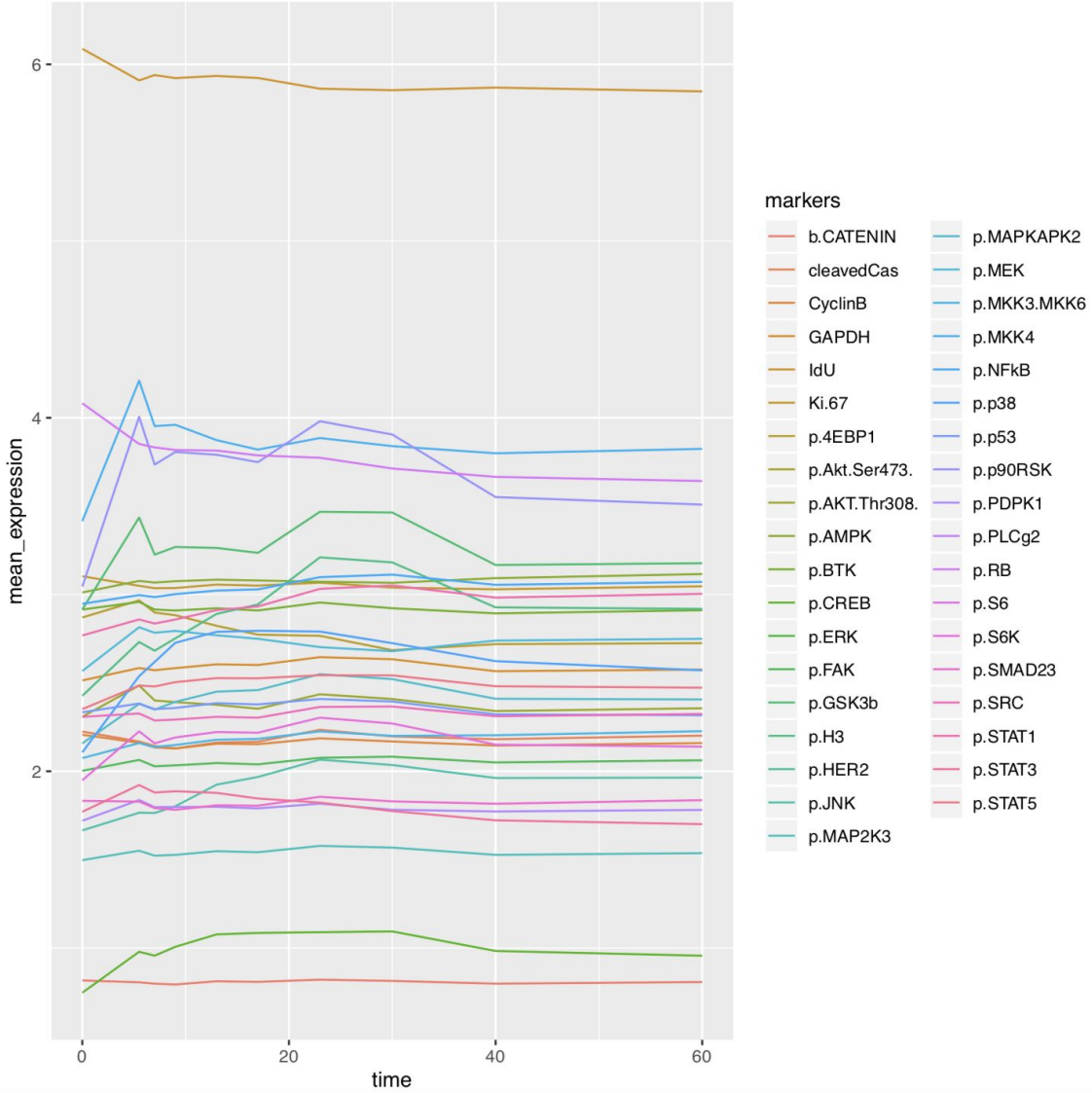
Process

Question: Can we predict missing marker values with the datasets given?

1. Exploratory Data Analysis
    a. PCA
    b. Distribution Plots
    c. Data PreProcessing
2. Feature Engineering/Selection
    a. Correlation Matrix
    b. Sequential Forward and Sequential Backward Selection
    c. Boruta
3. Building the Predictive Models (Multivariate Multiple Regression Method)
    a. Neural Nets
    b. SVMs
    c. RandomForest
4. Validation and Comparison

Distribution Plots




mean_expression
6
4
2
0
20
40
60
time
markers
b.CATENIN
cleavedCas
CyclinB
GAPDH
IdU
Ki.67
p.4EBP1
p.Akt.Ser473.
p.AKT.Thr308.
p.AMPK
p.BTK
p.CREB
p.ERK
p.FAK
p.GSK3b
p.H3
p.HER2
p.JNK
p.MAP2K3
p.MAPKAPK2
p.MEK
p.MKK3.MKK6
p.MKK4
p.NFkB
p.p38
p.p53
p.p90RSK
p.PDPK1
p.PLCg2
p.RB
p.S6
p.S6K
p.SMAD23
p.SRC
p.STAT1
p.STAT3
p.STAT5

Data Imputation using KNN

- Wanted to re-perform exploratory data analysis by imputing NA values of Median Single Cell Data using a more complex/accurate method than using solely row means
- 1987 NA values in the dataset (91,573 other data points)
- NA values intentional-omitted for purposes of the challenge

```
#Imputation using KNN
```{r}
#Imputation using the knn algorithm
set.seed(12345)

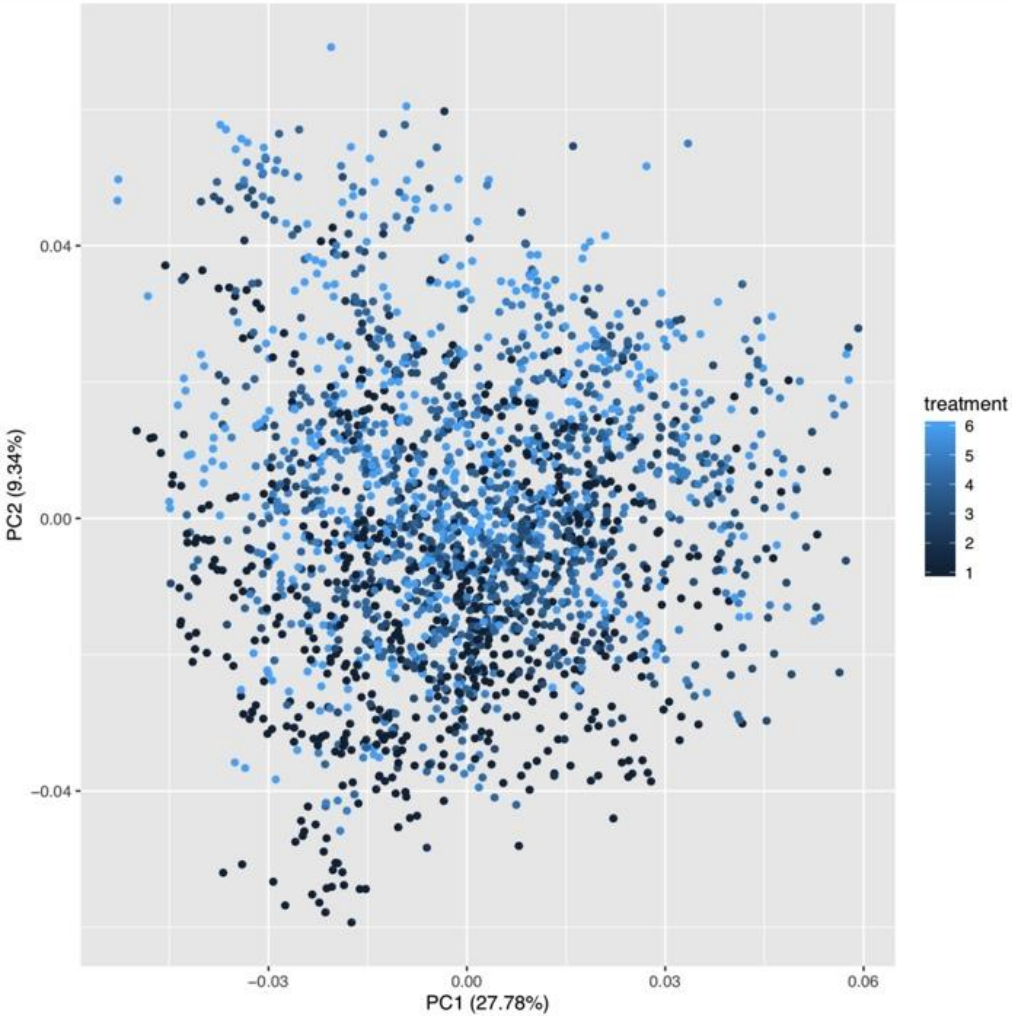
#install.packages("caret")
#install.packages("RANN")
#install.packages("data.table")

library(caret)
library(RANN)


Filtered_scData_knn_Model <- preProcess(Median_Single_Cell, "knnImpute")
Filtered_scData_pred <- predict(Filtered_scData_knn_Model, Median_Single_Cell)
```
```

Principal Component Analysis

- Next Step was to perform dimensionality reduction
  - Very high dimensionality within this DREAM challenge (i.e. some datasets with ~60 variables and ~23,000 observations)
- Low redundancy within this dataset
  - Clustered by treatment Types
  - Treatment Types:
    i. EGF
    ii. Full
    iii. iEGFR
    iv. iMEK
    v. iPI3K
    vi. iPKC
  - ~37% of the variance can be explained by the first two PCs



Feature Selection

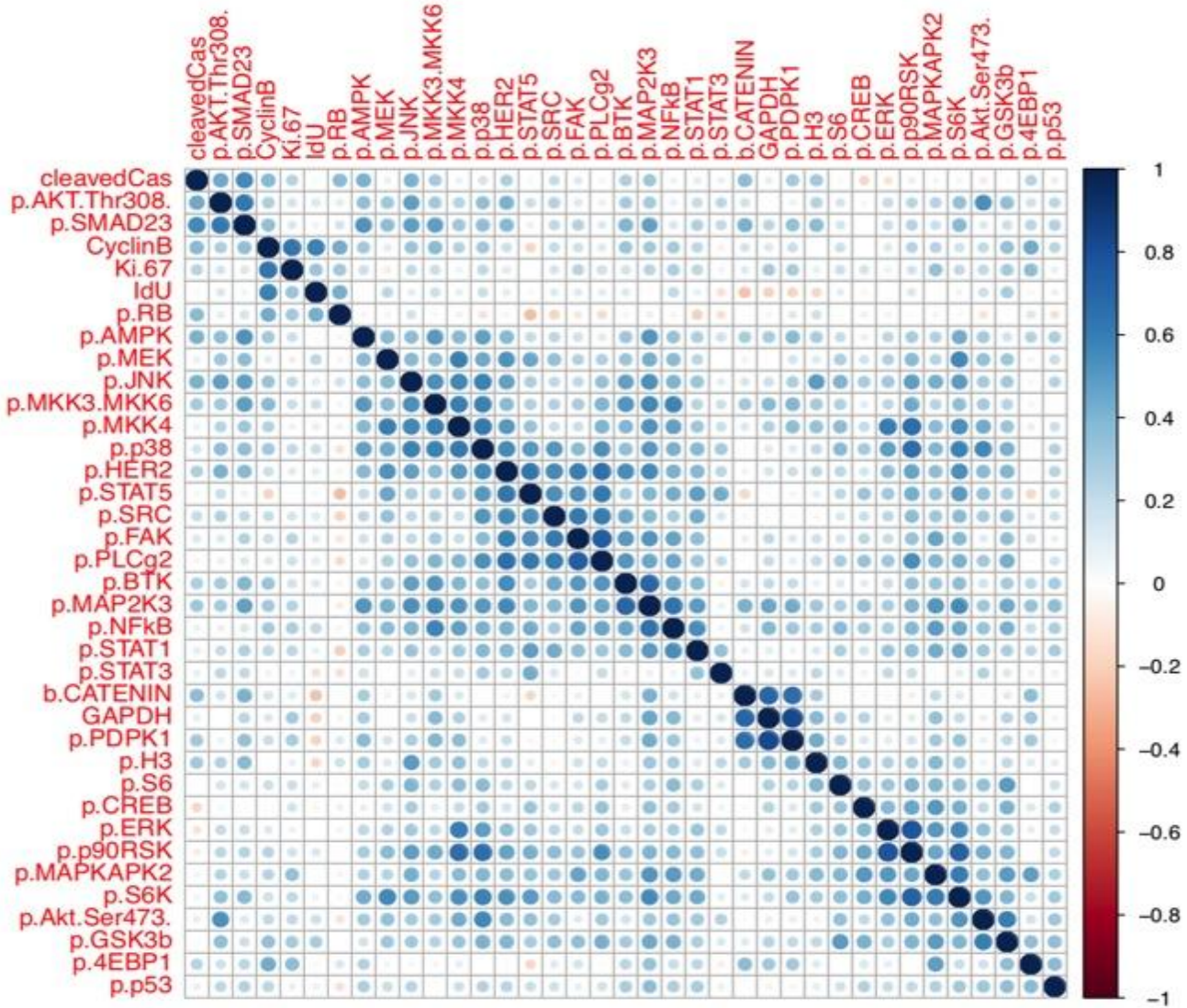
- Modeled after Nature Biotechnology paper on drug sensitivity challenge
- First, produced a correlation matrix in order to see highly correlated variables and remove those when building the model

```
#Correlation
```{r}
correlation_matrix = cor(single_cell_num)

#install.packages("corrplot")
library(corrplot)

pdf("Correlation_Median_sc", height = 8, width = 8)
corrplot(correlation_matrix, order="hclust")
dev.off()
```
```

Correlation Matrix



- Created a correlation matrix using corrplot to observe if any markers were highly correlated

Highly Correlated Variables:

- p.S6K
- p.p90RSK
- p.PLCg2
- GAPDH

Feature Selection (cont.)

- Next step, decided to use method described in Nature Biotechnology Paper
  - Employed both Sequential Forward and Sequential Backward methods simultaneously using MASS package in R

```
#Feature selection
```{r}
#Detailed as one of the kernel based methods used in the Nature Biotechnology paper, feature
selection was performed by using both sequential forward selection and sequential backward
selection.

library(MASS)

res.lm <- lm(p.Akt.Ser473. + p.ERK + p.HER2 + p.PLCg2 + p.S6 ~., data= single_cell_num)

step <- stepAIC(res.lm, direction = "both", trace = F)
step

summary(step)
```
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-4.6730 -0.7558  0.0247  0.7811  5.3680

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -0.05167    0.02395  -2.157 0.031078 *
b.CATENIN       -0.12070    0.05048  -2.391 0.016879 *
cleavedCas      -0.41624    0.04754  -8.756  < 2e-16 ***
CyclinB          0.36199    0.05981   6.053 1.66e-09 ***
GAPDH            0.62420    0.06534   9.553  < 2e-16 ***
IdU             -0.19477    0.04202  -4.636 3.76e-06 ***
Ki.67           -0.07833    0.03982  -1.967 0.049273 *
p.4EBP1         -0.71357    0.04495 -15.874  < 2e-16 ***
p.AKT.Thr308.    0.55629    0.03919  14.193  < 2e-16 ***
p.AMPK           0.27989    0.03778   7.408 1.79e-13 ***
p.BTK            0.37338    0.04745   7.869 5.47e-15 ***
p.CREB          -0.25684    0.03703  -6.935 5.25e-12 ***
p.FAK            0.27366    0.04749   5.762 9.41e-09 ***
p.GSK3b          0.81215    0.03786  21.453  < 2e-16 ***
p.H3             0.27452    0.03937   6.972 4.06e-12 ***
p.JNK           -0.08547    0.04727  -1.808 0.070727 .
p.MAP2K3        -0.33499    0.05760  -5.816 6.86e-09 ***
p.MAPKAPK2       0.43106    0.04747   9.080  < 2e-16 ***
p.MEK           -0.13891    0.05046  -2.753 0.005956 **
p.MKK3.MKK6     -0.64533    0.05638 -11.446  < 2e-16 ***
p.MKK4           0.89963    0.05377  16.731  < 2e-16 ***
p.NFkB           0.26974    0.05205   5.183 2.38e-07 ***
p.p38            0.42156    0.05333   7.905 4.10e-15 ***
p.p53            0.20137    0.03758   5.358 9.26e-08 ***
p.p90RSK         0.46656    0.05827   8.007 1.84e-15 ***
p.PDPK1         -0.24712    0.06043  -4.089 4.48e-05 ***
p.RB             0.14851    0.03966   3.745 0.000185 ***
p.S6K            0.32969    0.05673   5.812 7.04e-09 ***
p.SMAD23        -0.18295    0.05298  -3.453 0.000565 ***
p.SRC            0.44913    0.04580   9.807  < 2e-16 ***
p.STAT1         -0.06593    0.04268  -1.545 0.122502
p.STAT3          0.19202    0.04113   4.669 3.20e-06 ***
p.STAT5          0.32886    0.06203   5.302 1.26e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.158 on 2306 degrees of freedom
Multiple R-squared:  0.8825,    Adjusted R-squared:  0.8809
F-statistic: 541.3 on 32 and 2306 DF,  p-value: < 2.2e-16
```

Feature Selection (cont.)

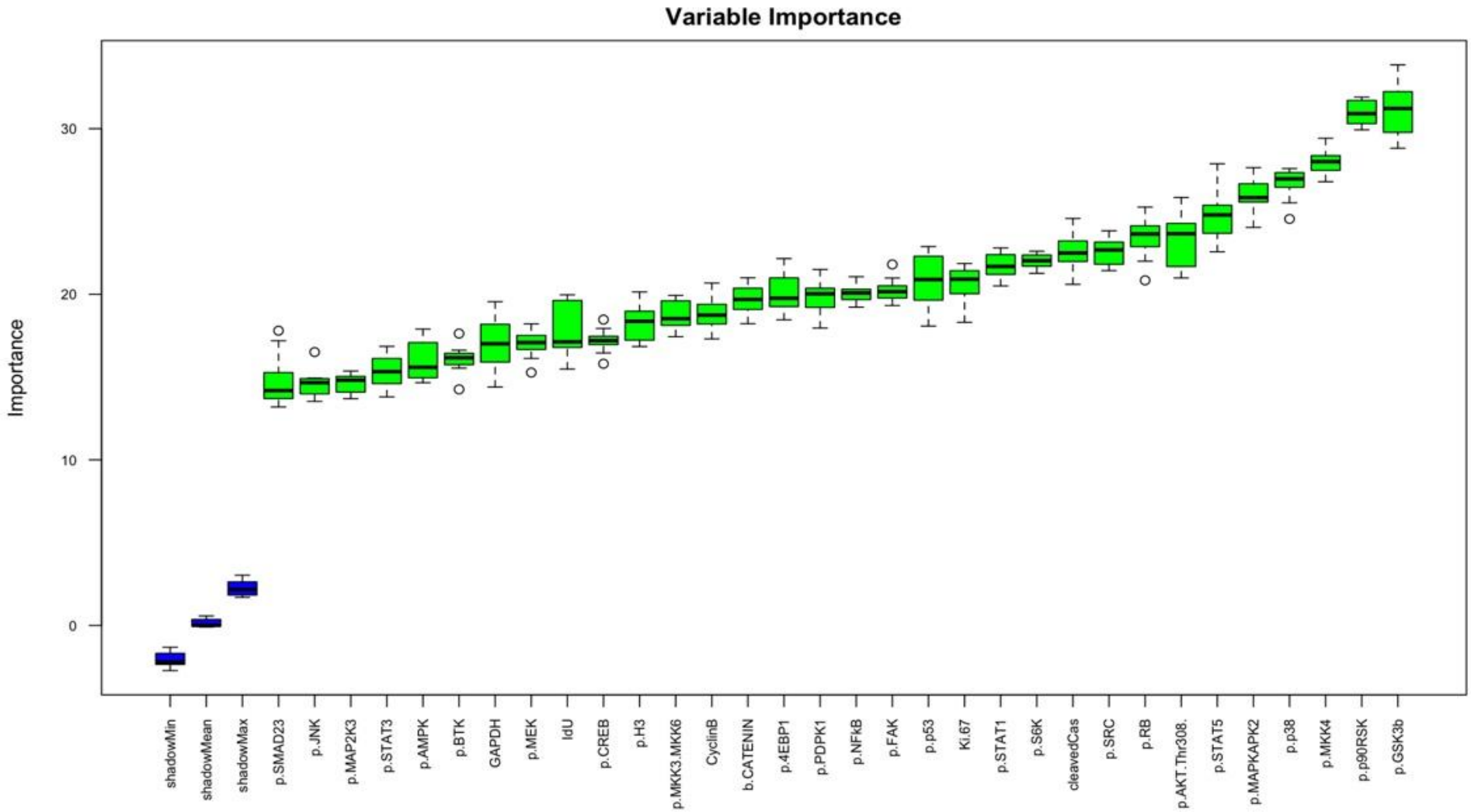
- Used summary() function to observe which variables were statistically significant.
- Ultimately, decided to include those that were very highly statistically significant (***)
- Variables that didn’t make the cut 😢:
  - b.CATENIN
  - Ki.67
  - p.JNK
  - p.STAT1

Feature Selection pt. 3

- Made use of the Boruta package in R
  - Feature wrapper model that uses randomForest to perform feature selection
  - Produces Variable importance plot
    - No strong conclusions drawn from this, however.

```
library(Boruta)

boruta_output <- Boruta(p.Akt.Ser473. + p.ERK + p.HER2 + p.PLCg2 + p.S6 ~., data=
single_cell_num, doTrace=2)
```


Variable Importance
Importance
30
20
10
0
shadowMin
shadowMean
shadowMax
p.SMAD23
p.JNK
p.MAP2K3
p.STAT3
p.AMPK
p.BTK
GAPDH
p.MEK
IdU
p.CREB
p.H3
p.MKK3.MKK6
CyclinB
b.CATENIN
p.4EBP1
p.PDPK1
p.NFkB
p.FAK
p.p53
Ki.67
p.STAT1
p.S6K
cleavedCas
p.SRC
p.RB
p.AKT.Thr308.
p.STAT5
p.MAPKAPK2
p.p38
p.MKK4
p.p90RSK
p.GSK3b

Model Construction Using Neural Networks

- **Built the model both before and after feature selection**
- **Normalized the data (Not required, but it helps).**

```
setwd("~/Desktop/Master's/Masters_Year_2/Fall_2019/Machine_Learning/Project/Subchallenge I")
#Must use Neural Nets to build this model
library(neuralnet)

##VERY IMPORTANT TO NORMALIZE DATA FOR NEURAL NETS

Normalize <- function(x){
  return((x-min(x))/(max(x)-min(x)))
}

Normalized_sc_Data <- as.data.frame(lapply(single_cell_num, Normalize))
#Next, Must create training and test sets

#Set training size to same size used in the Nature paper
training_size = floor(0.75*nrow(Normalized_sc_Data))

set.seed(123)
train_ind <- sample(seq_len(nrow(Normalized_sc_Data)), size = training_size)

training_set=Normalized_sc_Data[train_ind,]
test_set= Normalized_sc_Data[-train_ind,]

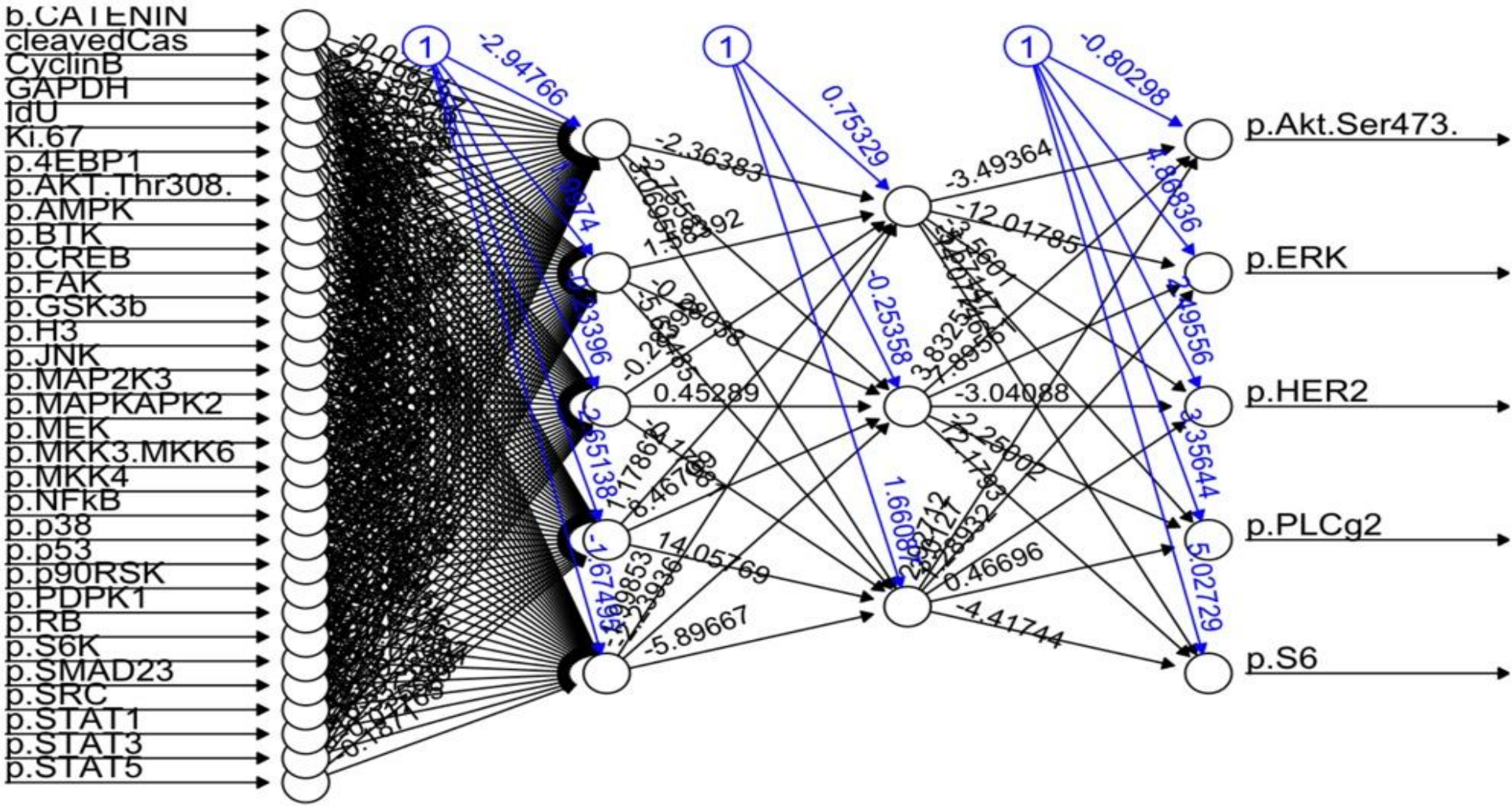


#Create Formula

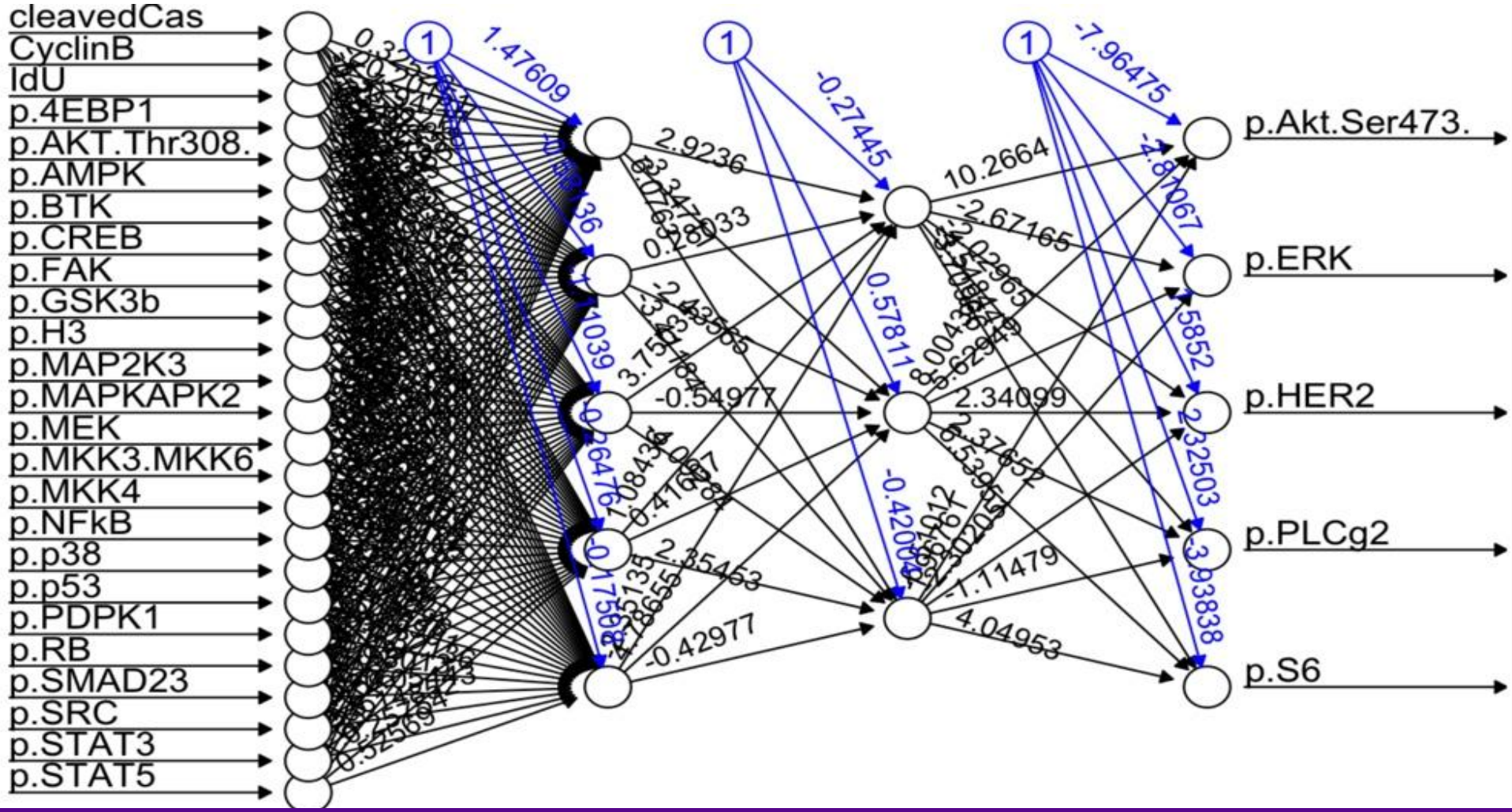
nn_formula= paste("p.Akt.Ser473. + p.ERK + p.HER2 + p.PLCg2 + p.S6 ~", Predictor_Variables,
collapse = "+")


scPhospho_Model <- neuralnet(nn_formula, data = training_set, hidden = c(5,3), linear.output =
F, stepmax = 1e6)
```

Neural Network **BEFORE** Feature Selection



Neural Network **AFTER** Feature Selection



Average Predicted Values and RMSE (After Feature Selection)

RMSE of Model: 0.00313177

	p.Akt.Ser473.	p.ERK	p.HER2	p.PLCg2	p.S6
AU565	0.3489693	0.4322657	0.3910538	0.3301062	0.4713397
EFM19	0.3093835	0.3826701	0.4179834	0.3504332	0.3893327
HCC2218	0.3053547	0.3382196	0.2639658	0.1941608	0.3863747
LY2	0.3416314	0.4078072	0.3853052	0.3196609	0.4382133
MACLS2	0.3394914	0.4110384	0.4546170	0.3916200	0.4138294
MDAMB436	0.3071842	0.3576960	0.2492930	0.1848483	0.4233025

randomForestSRC

- **Package that makes it possible to perform multivariate regression using a randomForest model**

```
#Random Forest
```{r}
training_set=Normalized_sc_Data[train_ind,]
test_set= Normalized_sc_Data[-train_ind,]
#install.packages("randomForestSRC")
library(randomForestSRC)
rf <- rfsrc(Multivar(p.Akt.Ser473. + p.ERK + p.HER2 + p.PLCg2 + p.S6) ~
cleavedCas+CyclinB+IdU+p.4EBP1+p.AKT.Thr308.+p.AMPK+p.BTK+p.CREB+p.FAK+p.GSK3b+p.H3+p.MAP2K3+p.
MAPKAPK2+p.MEK+p.MKK3.MKK6+p.MKK4+p.NFkB+p.p38+p.p53+p.PDPK1+p.RB+p.SMAD23+p.SRC+p.STAT3+p.STAT
5, data = training_set)

pred = predict.rfsrc(rf, newdata=test_set)

print(rf)
print(pred)



RMSE(pred$regrOutput$p.Akt.Ser473.$predicted, pred_test_set$p.Akt.Ser473.)
RMSE(pred$regrOutput$p.ERK$predicted, pred_test_set$p.ERK)
RMSE(pred$regrOutput$p.HER2$predicted, pred_test_set$p.HER2)
RMSE(pred$regrOutput$p.PLCg2$predicted, pred_test_set$p.PLCg2)
RMSE(pred$regrOutput$p.S6$predicted, pred_test_set$p.S6)
```
```

Average Predicted Values and RMSE

RMSE of Model: 0.0855
R-Squared: 0.920

	p.Akt.Ser473 .	p.ERK	p.HER2	p.PLCg2	p.S6
AU565	0.258	0.209	0.192	0.197	0.217
EFM19	0.275	0.226	0.193	0.195	0.224
HCC2218	0.269	0.219	0.209	0.206	0.226
LY2	0.267	0.217	0.193	0.198	0.222
MACLS2	0.254	0.209	0.184	0.191	0.220
MDAMB436	0.286	0.225	0.228	0.218	0.237

Challenges

- **Datasets are extremely large**
  - High-dimensionality within datasets as well as between datasets
- **Modeling cell-line response is especially difficult due to the conglomeration of factors.**

Conclusions

- **Neural Networks performed the best when compared to RandomForest.**
- **Other methods attempted did not work**
  - SVM
  - XGB
- **Multi-target multiple regression analysis is extremely difficult and time-consuming, yet could be effective in creating more robust models.**
  - Using Python might've been better.
- **Never doing a DREAM challenge again.**

Hypothetical Next Steps

- **Perhaps create a function and/or algorithm to, instead, analyze each response variable individually, then combine results.**
- **Test using a ‘Gold Standard’ dataset**
- **Find a way to make use of genomics and transcriptomics data to improve the models.**
  - Packages such as mixOmics and DIABLO in R
  - Or maybe use one of those datasets to build the model