Data prep

Data Setup and Loading Libraries

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.1.3
```

```
## randomForest 4.7-1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
raw_data <- data.table::fread("data_raw/expression.csv")
raw_data <- raw_data[,-1]
label <- raw_data$label
genes <- raw_data[,-"label"]
gene_names <- colnames(genes)

b2 <- data.table::fread("data_raw/b2_cell_info.csv")
b3 <- data.table::fread("data_raw/b3_cell_info.csv")
genes$time <- c(b2$sim_time, b3$sim_time)

b3_grn <- data.table::fread("data_raw/b3_cellwise_grn.csv")
b2_grn <- data.table::fread("data_raw/b2_cellwise_grn.csv")

b2_data <- genes[label == "D",]
b3_data <- genes[label == "C",]
```

Multiple Regression

Elastic Net

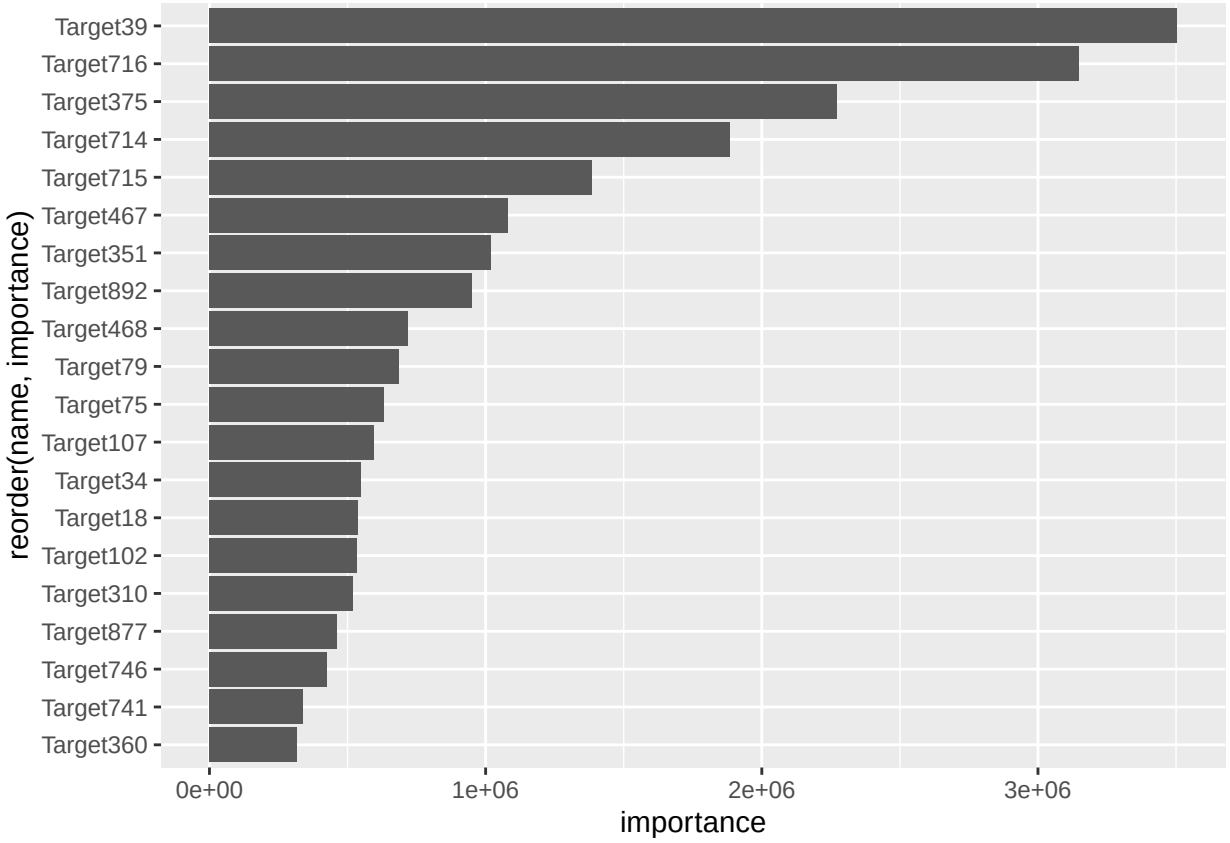
Random Forest

Random Forest on Full Data

```
full_forest <- randomForest(time ~ ., data = genes)

print(full_forest)
```

```
##
## Call:
##  randomForest(formula = time ~ ., data = genes)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1677
##
##           Mean of squared residuals: 6144.398
##                     % Var explained: 72.87
```

```
ind <- order(importance(full_forest,type = 2), decreasing=T)[1:20]
bar_plot <- as.data.frame(cbind(importance(full_forest,type = 2)[ind], gene_names[ind]))
colnames(bar_plot) <- c("importance", "name")
bar_plot$importance <- as.numeric(bar_plot$importance)

ggplot(bar_plot, aes(x = reorder(name, importance), y = importance)) +
  geom_bar(stat= "identity") +
  coord_flip()
```




```
time_pred <- predict(full_forest, newdata = genes[,-"time"])
mean(((time_pred - genes$time)^2)^.5)
```

```
## [1] 18.88018
```

```
b3_grn$regulator[b3_grn$target %in% bar_plot$name[1:7]]
```

```
##  [1] "D3_TF1"   "D3_TF1"   "B6_TF1"   "D1_TF1"   "B8_TF1"   "B8_TF1"
##  [7] "B8_TF1"   "B2_TF1"   "Target23" "C1_TF1"
```

```
b2_grn$regulator[b2_grn$target %in% bar_plot$name[1:7]]
```

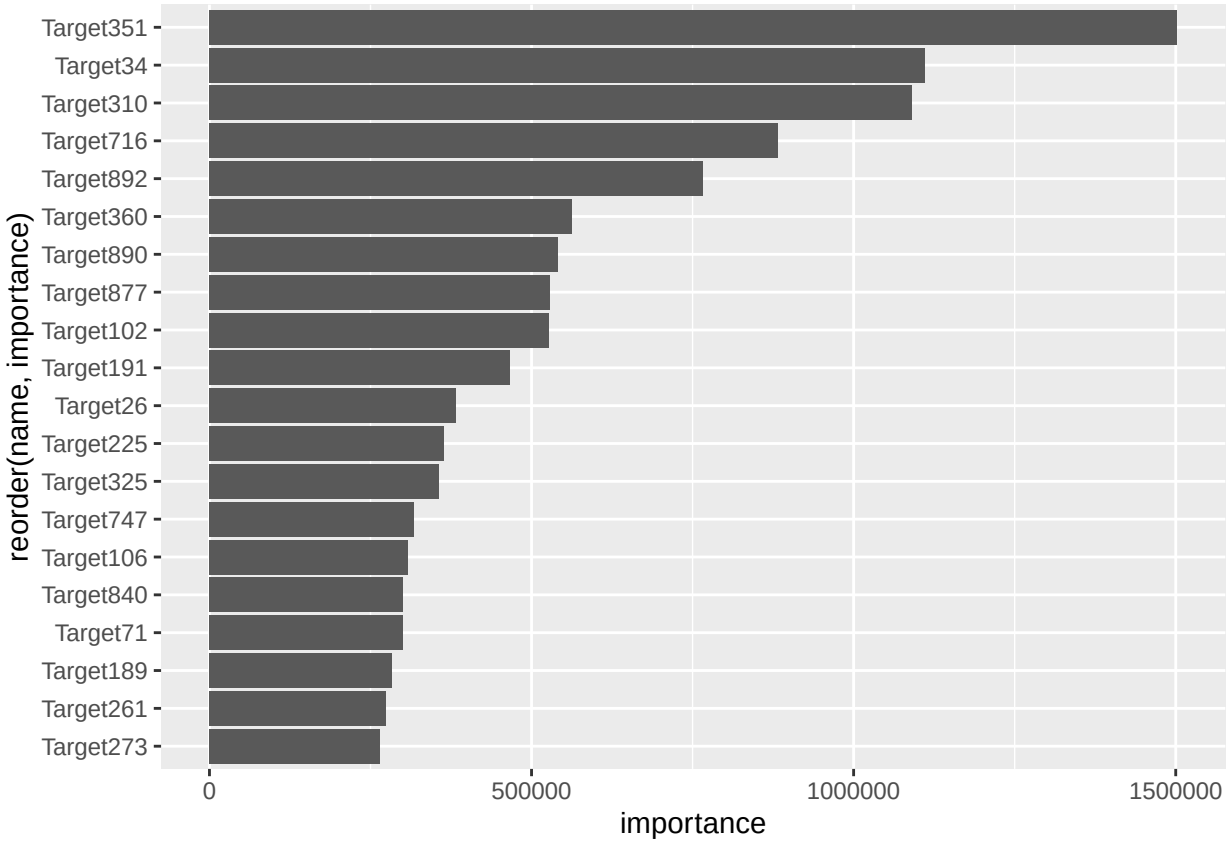
```
##  [1] "D3_TF1"   "D3_TF1"   "B6_TF1"   "D1_TF1"   "B8_TF1"   "B8_TF1"
##  [7] "B8_TF1"   "B2_TF1"   "Target23" "C1_TF1"
```

Random Forest on B2 Data

```
b2_forest <- randomForest(time ~ ., data = b2_data)

print(b2_forest)
```

```
##
## Call:
##  randomForest(formula = time ~ ., data = b2_data)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1677
##
##           Mean of squared residuals: 6542.022
##                     % Var explained: 74.96
```

```
b2_ind <- order(importance(b2_forest,type = 2), decreasing=T)[1:20]
bar_plot_b2 <- as.data.frame(cbind(importance(b2_forest,type = 2)[b2_ind], gene_names[b2_ind]))
colnames(bar_plot_b2) <- c("importance", "name")
bar_plot_b2$importance <- as.numeric(bar_plot_b2$importance)

ggplot(bar_plot_b2, aes(x = reorder(name, importance), y = importance)) +
  geom_bar(stat= "identity") +
  coord_flip()
```




```
time_pred_b2 <- predict(b2_forest, newdata = b2_data[,-"time"])
mean(((time_pred_b2 - b2_data$time)^2)^.5)
```

```
## [1] 20.02522
```

```
b2_grn$regulator[b2_grn$target %in% bar_plot_b2$name[1:7]]
```

```
## [1] "D3_TF1" "D3_TF1" "D3_TF1" "B8_TF1" "B2_TF1" "B3_TF1" "B3_TF1" "B3_TF1"
```

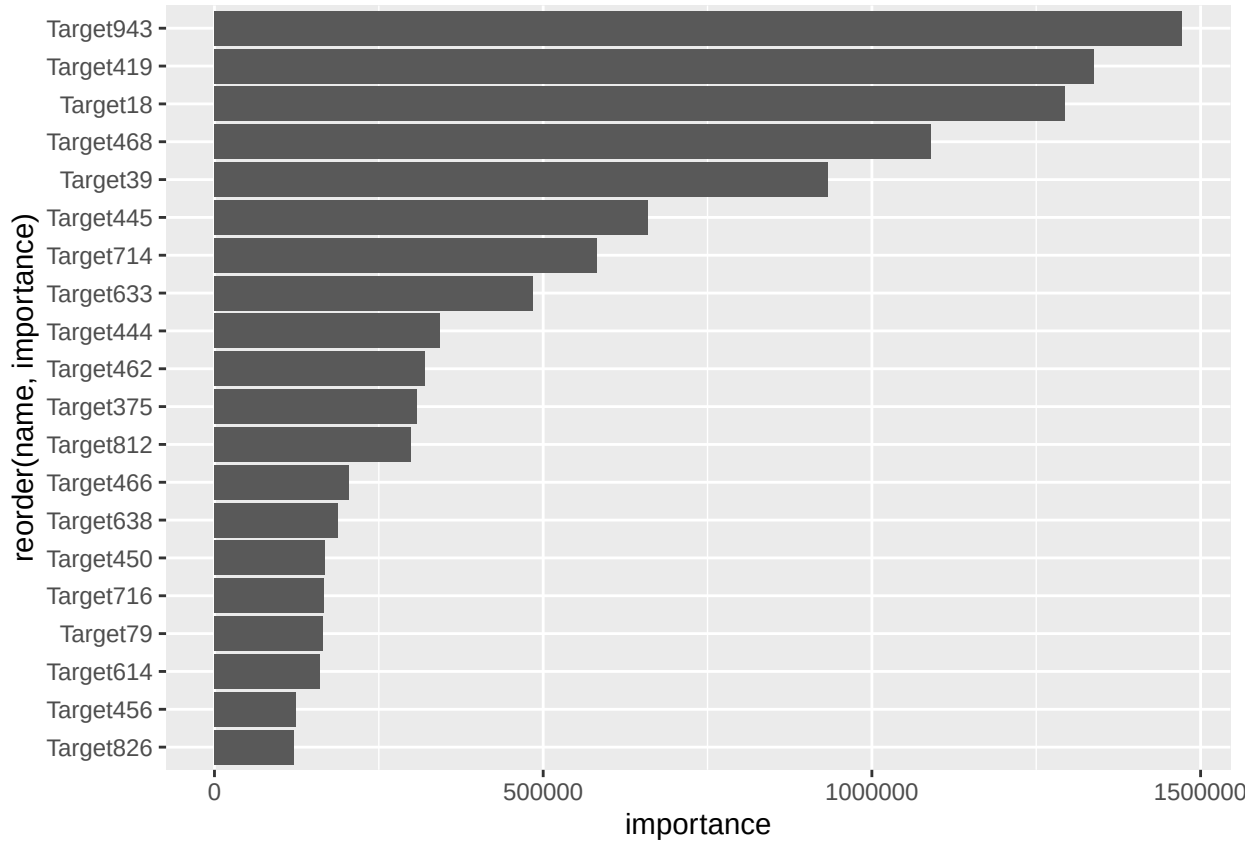
Random Forest on B3 Data

```
b3_forest <- randomForest(time ~ ., data = b3_data)

print(b3_forest)
```

```
##
## Call:
##  randomForest(formula = time ~ ., data = b3_data)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1677
##
##           Mean of squared residuals: 5160.318
##                     % Var explained: 72.47
```

```
b3_ind <- order(importance(b3_forest,type = 2), decreasing=T)[1:20]
bar_plot_b3 <- as.data.frame(cbind(importance(b3_forest,type = 2)[b3_ind], gene_names[b3_ind]))
```

```
colnames(bar_plot_b3) <- c("importance", "name")
bar_plot_b3$importance <- as.numeric(bar_plot_b3$importance)

ggplot(bar_plot_b3, aes(x = reorder(name, importance), y = importance)) +
  geom_bar(stat= "identity") +
  coord_flip()
```




```
time_pred_b3 <- predict(b3_forest, newdata = b3_data[,-"time"])
mean(((time_pred_b3 - b3_data$time)^2)^.5)
```

```
## [1] 17.38207
```

```
b3_grn$regulator[b3_grn$target %in% bar_plot_b3$name[1:7]]
```

```
##  [1] "B6_TF1" "B6_TF1" "B6_TF1" "B6_TF1" "C4_TF1" "C4_TF1" "D1_TF1" "B8_TF1"
##  [9] "B2_TF1" "C2_TF1"
```