Diabetes Kidney Tumor Gene Expressions ML

Janis Corona

12/23/2019

### This R-Markdown file uses R to analyze kidney tumor tissue gene expression profiles of Diabetes patients in Advanced, Early, or healthy states, performs predictive analytics on those genes, text mining, and compares Diabetic kidney disease to renal disease kidney tumor tissue from a previous study.

### GSE142025 from <https://www.ncbi.nlm.nih.gov/geo>

Overall design of the GSE142025 study is using kidney tumor and healthy kidney samples to analyze stages of diabetes mellitis. A total of 28 patients with biopsy-proven DN hospitalized from January 2015 to December 2016 in Shanghai Jiao Tong University Affiliated Sixth People’s Hospital were enrolled in the study. Nine control human kidney samples were obtained from the unaffected portion of tumor nephrectomies. RNA-seq was performed on 28 DN and 9 control samples.

Later on, machine learning with KNN and Random Forest is used to test out these samples on identifying the disease state derived, and also using a separate kidney tumor study that analyzed kidney tumors from healthy and renal disease patients. The actual data wasn’t used, just the gene targets identified as the top 20 up regulated and top 20 increased fold change as well as top 20 down regulated genes from that study that were available in this data set to use machine learning on.

#### Read in the three data sets for Advanced, early, and control kidney tumor samples

Advanced <- read.csv('Advanced\_Diabetes.csv',  
 sep=',', header=TRUE,  
 na.strings=c('',' ','NA'))  
head(Advanced)

## Symbol A11A A12A A13A A14A A15A A17A  
## 1 A1BG 6.866819 6.947427 6.824098 7.386014 7.428343 7.168584  
## 2 A1BG-AS1 5.569898 6.295194 5.738570 6.510046 6.930343 4.605649  
## 3 A1CF 10.919599 11.271185 12.103053 11.525433 10.544823 10.869673  
## 4 A2M 16.263047 16.355500 16.324612 16.485816 16.529039 15.626727  
## 5 A2M-AS1 8.006878 7.350193 7.855322 8.051924 8.763796 7.867349  
## 6 A4GALT 11.661596 11.672175 11.106141 11.104311 11.499382 11.937155  
## A18A A19A A20A A21A A22A A23A A24A  
## 1 7.917317 7.725042 7.419004 6.785917 6.940540 6.736050 7.717509  
## 2 7.436713 6.695364 6.227500 5.870013 6.391545 6.604291 5.885054  
## 3 9.253155 10.306971 10.487491 11.288373 11.174769 11.997887 11.056253  
## 4 16.286824 16.263047 16.582387 16.136284 16.228550 16.010469 16.286824  
## 5 8.820032 8.474119 8.265362 7.680565 7.722889 7.939991 7.399227  
## 6 11.532014 11.519927 11.297063 11.738423 11.387692 11.300646 11.194402  
## A25A A26A A28A A29A A30A A31A A32A  
## 1 7.133951 7.736946 6.914328 6.119405 7.355977 7.246352 7.072140  
## 2 6.289514 6.895121 6.035207 5.929707 6.140356 6.845240 6.316558  
## 3 11.383361 10.468693 10.683659 12.141484 11.385980 9.702990 11.854978  
## 4 16.760883 16.529039 17.281969 14.682377 16.286824 17.151443 16.286824  
## 5 8.191375 8.778039 8.367761 7.734992 7.820654 7.293855 7.696993  
## 6 11.475803 11.471783 11.069388 11.439888 11.309553 12.005243 11.476316  
## A33A  
## 1 6.702656  
## 2 5.510968  
## 3 12.102831  
## 4 15.242800  
## 5 7.388275  
## 6 11.504378

tail(Advanced)

## Symbol A11A A12A A13A A14A A15A A17A  
## 17179 ZXDC 12.450259 12.378594 12.32860 12.269347 12.423913 12.137397  
## 17180 ZYG11A 9.139439 9.214225 10.76203 9.642019 8.514463 9.041981  
## 17181 ZYG11B 11.987720 11.917192 12.14657 12.049625 11.753244 11.985167  
## 17182 ZYX 12.547317 12.845804 11.97220 12.516154 12.962498 12.327639  
## 17183 ZZEF1 13.540235 13.462678 13.35209 13.427467 13.627756 13.243968  
## 17184 ZZZ3 11.666347 11.393298 11.64141 11.654008 11.423713 11.554180  
## A18A A19A A20A A21A A22A A23A A24A  
## 17179 12.499739 12.318893 12.134444 12.236595 12.324117 12.39920 12.409500  
## 17180 6.569694 8.065765 8.522942 9.169757 8.715488 10.76369 8.601861  
## 17181 11.813660 11.742540 12.045764 12.081938 11.826927 12.10949 11.866919  
## 17182 13.314676 12.700579 12.555444 12.646099 12.676266 12.02462 12.242290  
## 17183 13.934281 13.666739 13.511316 13.385465 13.399202 13.41879 13.541213  
## 17184 11.503543 11.605744 11.757343 11.535500 11.729530 11.71974 11.670762  
## A25A A26A A28A A29A A30A A31A A32A  
## 17179 12.452553 12.192169 12.248505 12.33562 12.277963 12.278785 12.431563  
## 17180 9.844598 7.704214 8.736429 10.99172 9.600329 7.548216 9.818313  
## 17181 12.046564 11.701432 11.972733 12.21922 12.125794 11.598501 11.819871  
## 17182 12.468584 13.173959 12.837322 10.90698 11.954119 13.316251 12.247485  
## 17183 13.614064 13.495763 13.649331 13.27649 13.565099 13.379326 13.449409  
## 17184 11.490686 11.687320 11.680188 11.82848 11.777059 11.569164 11.613989  
## A33A  
## 17179 12.17309  
## 17180 10.21618  
## 17181 12.24953  
## 17182 11.35655  
## 17183 13.32562  
## 17184 11.65471

Early <- read.csv('Early\_Diabetes.csv',  
 sep=',', header=TRUE,  
 na.strings=c('',' ','NA'))  
head(Early)

## Symbol B3B B4B B5B B6B B8B B9B  
## 1 A1BG 6.392371 6.330405 6.820246 6.563131 7.050308 6.528168  
## 2 A1BG-AS1 5.465347 6.173528 6.190452 5.977362 5.598883 6.029563  
## 3 A1CF 11.956522 12.837708 12.267798 11.832790 11.953380 12.249015  
## 4 A2M 15.229170 15.459651 15.301756 15.670659 15.055246 15.035374  
## 5 A2M-AS1 7.556442 7.750161 7.119460 7.645756 7.556537 7.432906  
## 6 A4GALT 11.529279 10.975482 11.599421 11.810890 11.339914 11.493189

tail(Early)

## Symbol B3B B4B B5B B6B B8B B9B  
## 17179 ZXDC 12.50089 12.35885 12.56098 12.39920 12.53800 12.60243  
## 17180 ZYG11A 10.40099 10.91147 10.99252 11.64653 10.77875 10.85524  
## 17181 ZYG11B 12.16743 12.40055 12.21847 12.24748 12.38329 12.12121  
## 17182 ZYX 11.65988 11.01910 11.37640 11.54641 11.03130 11.43176  
## 17183 ZZEF1 13.37666 13.46086 13.43450 13.41222 13.46425 13.37467  
## 17184 ZZZ3 11.57883 11.94816 11.58914 11.59594 11.70042 11.64043

Control <- read.csv('Control\_Group.csv',  
 sep=',', header=TRUE,  
 na.strings=c('',' ','NA'))  
head(Control)

## Symbol N2A N3A N5B N6B N7B N8B  
## 1 A1BG 6.469842 6.203464 6.888575 5.273028 6.417218 5.997890  
## 2 A1BG-AS1 5.719696 5.456240 6.201278 4.484043 5.512423 5.159483  
## 3 A1CF 12.185284 12.509666 12.130412 12.324429 12.568520 12.353611  
## 4 A2M 15.475354 14.997607 15.376017 15.131920 15.008034 15.242800  
## 5 A2M-AS1 7.450052 7.514131 7.281881 7.258395 7.532757 7.257831  
## 6 A4GALT 11.161922 11.048622 11.169556 11.280299 11.190465 11.010524  
## N10B N12B N13B  
## 1 6.623161 6.391958 5.697620  
## 2 5.978862 5.478157 4.877218  
## 3 12.398768 12.272683 12.245974  
## 4 15.607762 15.348862 15.091870  
## 5 7.753721 7.558049 7.302397  
## 6 12.099354 11.418056 11.279607

tail(Control)

## Symbol N2A N3A N5B N6B N7B N8B N10B  
## 17179 ZXDC 12.34379 12.28780 12.38231 12.33998 12.22138 12.29829 12.24927  
## 17180 ZYG11A 10.70907 10.91134 11.13118 10.48814 10.69882 10.75810 10.59434  
## 17181 ZYG11B 12.12181 12.37859 12.13310 12.28691 12.47953 12.17386 12.43599  
## 17182 ZYX 11.64980 10.94086 11.46237 11.69406 11.58041 11.70548 12.11209  
## 17183 ZZEF1 13.24595 13.38762 13.21031 13.27102 13.16645 13.24036 13.33641  
## 17184 ZZZ3 11.73993 11.87409 11.60744 11.55175 11.61081 11.66594 11.27469  
## N12B N13B  
## 17179 12.38662 12.18997  
## 17180 10.68433 11.24354  
## 17181 12.26879 12.40260  
## 17182 11.79548 12.71955  
## 17183 13.24718 13.17944  
## 17184 11.60293 11.41546

library(dplyr)

## Assign no duplicate instances of genes

Advanced1 <- Advanced[!duplicated(Advanced$Symbol),]

## Check that all the genes have 1 count each, they do

AdvancedCounts <- Advanced1 %>% group\_by(Symbol) %>%   
 summarise(counts = n())  
dim(AdvancedCounts)

## [1] 17182 2

unique(AdvancedCounts$counts)

## [1] 1

Early1 <- Early[!duplicated(Early$Symbol),]

## Check that all the genes have 1 count each, they do

EarlyCounts <- Early1 %>% group\_by(Symbol) %>%   
 summarise(counts = n())  
dim(EarlyCounts)

## [1] 17182 2

unique(EarlyCounts$counts)

## [1] 1

Control1 <- Control[!duplicated(Control$Symbol),]

## Check that all the genes have 1 count each, they do

ControlCounts <- Control1 %>% group\_by(Symbol) %>%   
 summarise(counts = n())  
dim(ControlCounts)

## [1] 17182 2

unique(ControlCounts$counts)

## [1] 1

#### Attach a field to each of the data sets of the row means for those samples unique gene

row.names(Advanced1) <- Advanced1$Symbol  
Advanced1 <- Advanced1[,2:22]  
Advanced1$Gene\_Means <- rowMeans(Advanced1)

row.names(Early1) <- Early1$Symbol  
Early1 <- Early1[,2:7]  
Early1$Gene\_Means <- round(rowMeans(Early1),3)  
Advanced1$Gene\_Means <- round(Advanced1$Gene\_Means,3)

row.names(Control1) <- Control1$Symbol  
Control1 <- Control1[,2:10]  
Control1$Gene\_Means <- round(rowMeans(Control1),3)

colnames(Advanced1)[22] <- "Advanced\_DM\_Means"  
colnames(Early1)[7] <- "Early\_DM\_Means"  
colnames(Control1)[10] <- "Control\_noDM\_Means"

means <- cbind(Advanced1[22], Early1[7], Control1[10])  
Combined <- cbind(means, Advanced1[1:21], Early1[1:6], Control1[1:9])

#### Create the fold change field to compare the change in early Diabetes Mellitis (DM) Kidney tumor gene expressions to healthy non DM kidney tissue gene expressions, Advanced DM to healthy, and Advanced to Early DM expressions.

Fold\_Change0 <- Combined %>% mutate(Fold\_Change\_Early\_Control = Early\_DM\_Means/Control\_noDM\_Means)  
  
Fold\_Change1 <- Fold\_Change0 %>% mutate(Fold\_Change\_Advanced\_Control = Advanced\_DM\_Means/Control\_noDM\_Means)  
  
Fold\_Change2 <- Fold\_Change1 %>% mutate(Fold\_Change\_Advanced\_Early = Advanced\_DM\_Means/Early\_DM\_Means)  
  
row.names(Fold\_Change2) <- row.names(Combined)  
  
Fold\_Changes <- Fold\_Change2[,c(40,41,42,1:39)]

#### Remove NaN’s or Not a number and Inf when dividing by zero or a very small value

Fold\_Changes$Fold\_Change\_Early\_Control <- gsub('NaN',0,Fold\_Changes$Fold\_Change\_Early\_Control)  
Fold\_Changes$Fold\_Change\_Early\_Control <- gsub('Inf', 0, Fold\_Changes$Fold\_Change\_Early\_Control)  
Fold\_Changes$Fold\_Change\_Early\_Control <- round(as.numeric(Fold\_Changes$Fold\_Change\_Early\_Control),3)  
  
Fold\_Changes$Fold\_Change\_Advanced\_Control <- gsub('NaN',0,Fold\_Changes$Fold\_Change\_Advanced\_Control)  
Fold\_Changes$Fold\_Change\_Advanced\_Control <- gsub('Inf', 0, Fold\_Changes$Fold\_Change\_Advanced\_Control)  
Fold\_Changes$Fold\_Change\_Advanced\_Control <- round(as.numeric(Fold\_Changes$Fold\_Change\_Advanced\_Control),3)  
  
Fold\_Changes$Fold\_Change\_Advanced\_Early <- gsub('NaN',0,Fold\_Changes$Fold\_Change\_Advanced\_Early)  
Fold\_Changes$Fold\_Change\_Advanced\_Early <- gsub('Inf', 0, Fold\_Changes$Fold\_Change\_Advanced\_Early)  
Fold\_Changes$Fold\_Change\_Advanced\_Early <- round(as.numeric(Fold\_Changes$Fold\_Change\_Advanced\_Early),3)  
  
write.csv(Fold\_Changes,'Fold\_Changes.csv', row.names=TRUE)

Top20\_FC\_Adv\_Early <- Fold\_Changes[order(Fold\_Changes$Fold\_Change\_Advanced\_Early, decreasing=TRUE)[0:20],]  
write.csv(Top20\_FC\_Adv\_Early,'Top20\_FC\_Adv-Early.csv', row.names=TRUE)

Top20\_FC\_Adv\_Control <- Fold\_Changes[order(Fold\_Changes$Fold\_Change\_Advanced\_Control, decreasing=TRUE)[0:20],]  
write.csv(Top20\_FC\_Adv\_Control,'Top20\_FC\_Adv-Ctrl.csv',row.names=TRUE)

Top20\_FC\_Early\_Control <- Fold\_Changes[order(Fold\_Changes$Fold\_Change\_Early\_Control, decreasing=TRUE)[0:20],]  
write.csv(Top20\_FC\_Early\_Control,'Top20\_FC\_Early-Ctrl.csv',row.names=TRUE)

#### Create the Differential Expression fields for Control-Early, Control-Advanced, and Early-Advanced so that negative differential expression values mean those genes are up regulated in the more diseased state compared to the healthy state.

DE\_Control\_Adv <- Fold\_Changes %>% mutate(DE\_ctrl\_adv = round(Control\_noDM\_Means-Advanced\_DM\_Means,3))  
  
DE\_Early\_Adv <- DE\_Control\_Adv %>% mutate(DE\_early\_adv = round(Early\_DM\_Means-Advanced\_DM\_Means,3))  
  
DE\_Control\_Early <- DE\_Early\_Adv %>% mutate(DE\_ctrl\_early = round(Control\_noDM\_Means-Early\_DM\_Means,3))  
  
Differential <- DE\_Control\_Early[,c(43,45,44,1:42)]  
row.names(Differential) <- row.names(Fold\_Changes)  
  
write.csv(Differential, 'Differential\_FCs.csv', row.names=TRUE)

#### Since this is healthy - diseased, and less diseased-more diseased, positive values mean the diseased gene expression means are lower than the healthy gene expression levels or down regulated in diseased or more diseased. While the negative values mean those genes are up regulated in diseased or more diseased states compared to healthy or less diseased states.

down\_Ctrl\_Adv <- Differential[order(Differential$DE\_ctrl\_adv,   
 decreasing=TRUE),]  
down\_Ctrl\_Early <- Differential[order(Differential$DE\_ctrl\_early,   
 decreasing=TRUE),]  
down\_Early\_Adv <- Differential[order(Differential$DE\_early\_adv,   
 decreasing=TRUE),]  
  
up\_Ctrl\_Adv <- Differential[order(Differential$DE\_ctrl\_adv,   
 decreasing=FALSE),]  
up\_Ctrl\_Early <- Differential[order(Differential$DE\_ctrl\_early,   
 decreasing=FALSE),]  
up\_Early\_Adv <- Differential[order(Differential$DE\_early\_adv,   
 decreasing=FALSE),]

#### Top 20 downgraded genes

Top20\_down\_Ctrl\_Adv <- down\_Ctrl\_Adv[0:20,]  
Top20\_down\_Ctrl\_Early <- down\_Ctrl\_Early[0:20,]  
Top20\_down\_Early\_Adv <- down\_Early\_Adv[0:20,]

#### Top 20 upgraded genes

Top20\_up\_Ctrl\_Adv <- up\_Ctrl\_Adv[0:20,]  
Top20\_up\_Ctrl\_Early <- up\_Ctrl\_Early[0:20,]  
Top20\_up\_Early\_Adv <- up\_Early\_Adv[0:20,]

#### Write these up and down regulated genes to csv files

write.csv(Top20\_up\_Ctrl\_Adv, 'Top20\_up\_Ctrl\_Adv.csv', row.names=TRUE)  
write.csv(Top20\_up\_Ctrl\_Early, 'Top20\_up\_Ctrl\_Early.csv', row.names=TRUE)  
write.csv(Top20\_up\_Early\_Adv, 'Top20\_up\_Early\_Adv.csv', row.names=TRUE)  
  
write.csv(Top20\_down\_Ctrl\_Adv, 'Top20\_down\_Ctrl\_Adv.csv', row.names=TRUE)  
write.csv(Top20\_down\_Ctrl\_Early, 'Top20\_down\_Ctrl\_Early.csv', row.names=TRUE)  
write.csv(Top20\_down\_Early\_Adv, 'Top20\_down\_Early\_Adv.csv', row.names=TRUE)

### What are the top 20 genes up-regulated in Diabetes Mellitis Kidney disease of all three sets?

Control state versus the early DM state of up regulated genes

Up1 <- as.data.frame(row.names(Top20\_up\_Ctrl\_Early))  
colnames(Up1) <- 'Gene'  
Up1$dataType <- rep('early up regulated',20)  
Up1 <- Up1[order(Up1$Gene),]  
  
Up1

## Gene dataType  
## 12 ACTA1 early up regulated  
## 2 ADIPOQ early up regulated  
## 6 ATF6B early up regulated  
## 18 CIDEA early up regulated  
## 4 CIDEC early up regulated  
## 16 GREM1 early up regulated  
## 19 GTF2H4 early up regulated  
## 14 LEP early up regulated  
## 13 MB early up regulated  
## 7 MFAP5 early up regulated  
## 17 MROH2B early up regulated  
## 3 MYBPC1 early up regulated  
## 1 MYH7 early up regulated  
## 5 MYL2 early up regulated  
## 20 PLIN1 early up regulated  
## 15 SCARA5 early up regulated  
## 8 TNNT1 early up regulated  
## 10 TUSC5 early up regulated  
## 11 VARS2 early up regulated  
## 9 XIRP2 early up regulated

Early state versus the Advanced DM state of up regulated genes

Up2 <- as.data.frame(row.names(Top20\_up\_Early\_Adv))  
colnames(Up2) <- 'Gene'  
Up2$dataType <- rep('Advanced Early Up',20)  
Up2 <- Up2[order(Up2$Gene),]  
Up2

## Gene dataType  
## 4 ADAMDEC1 Advanced Early Up  
## 9 CCL19 Advanced Early Up  
## 5 COL6A5 Advanced Early Up  
## 11 CTLA4 Advanced Early Up  
## 2 CXCL6 Advanced Early Up  
## 17 FAM30A Advanced Early Up  
## 20 FCRLA Advanced Early Up  
## 3 GSTM1 Advanced Early Up  
## 7 HLA-DPB1 Advanced Early Up  
## 8 IGLL5 Advanced Early Up  
## 19 IL7R Advanced Early Up  
## 18 IRF4 Advanced Early Up  
## 15 LOC101926964 Advanced Early Up  
## 14 LTF Advanced Early Up  
## 16 MMP9 Advanced Early Up  
## 12 PAX5 Advanced Early Up  
## 6 TCL1A Advanced Early Up  
## 10 TNFRSF13B Advanced Early Up  
## 13 TUBB3 Advanced Early Up  
## 1 XIST Advanced Early Up

Control state versus the Advanced DM state of up regulated genes

Up3 <- as.data.frame(row.names(Top20\_up\_Ctrl\_Adv))  
colnames(Up3) <- 'Gene'  
Up3$dataType <- rep('Advanced Control Up', 20)  
Up3 <- Up3[order(Up3$Gene),]  
Up3

## Gene dataType  
## 17 ACKR1 Advanced Control Up  
## 20 ADIPOQ Advanced Control Up  
## 2 CCL19 Advanced Control Up  
## 12 CCL21 Advanced Control Up  
## 3 CFHR1 Advanced Control Up  
## 19 CLEC4C Advanced Control Up  
## 1 COL6A5 Advanced Control Up  
## 5 DCANP1 Advanced Control Up  
## 18 FCRL2 Advanced Control Up  
## 4 GREM1 Advanced Control Up  
## 6 IGLL5 Advanced Control Up  
## 10 IRF4 Advanced Control Up  
## 8 LINC01426 Advanced Control Up  
## 11 LTF Advanced Control Up  
## 7 PAX5 Advanced Control Up  
## 16 REG1A Advanced Control Up  
## 15 SERPINA3 Advanced Control Up  
## 14 TIFAB Advanced Control Up  
## 13 TUBB3 Advanced Control Up  
## 9 XIST Advanced Control Up

### What are the top 20 genes down-regulated in renal disease compared to healthy?

Control state versus the early DM state of down regulated genes

down1 <- as.data.frame(row.names(Top20\_down\_Ctrl\_Early))  
colnames(down1) <- 'Gene'  
down1$dataType <- rep('Early Control Down', 20)  
down1 <- down1[order(down1$Gene),]  
down1

## Gene dataType  
## 11 ADAMTS4 Early Control Down  
## 7 ATF3 Early Control Down  
## 10 CCL3 Early Control Down  
## 19 CXCL2 Early Control Down  
## 20 CYR61 Early Control Down  
## 9 DUSP1 Early Control Down  
## 14 DUSP2 Early Control Down  
## 3 EGR1 Early Control Down  
## 15 EGR2 Early Control Down  
## 12 EGR3 Early Control Down  
## 2 FOS Early Control Down  
## 1 FOSB Early Control Down  
## 16 JUNB Early Control Down  
## 13 MIR3189 Early Control Down  
## 4 NR4A1 Early Control Down  
## 6 NR4A2 Early Control Down  
## 8 NR4A3 Early Control Down  
## 5 RGS1 Early Control Down  
## 18 SLC2A3 Early Control Down  
## 17 ZFP36 Early Control Down

Early state versus the Advanced DM state of down regulated genes

down2 <- as.data.frame(row.names(Top20\_down\_Early\_Adv))  
colnames(down2) <- 'Gene'  
down2$dataType <- rep('Advanced Early Down', 20)  
down2 <- down2[order(down2$Gene),]  
down2

## Gene dataType  
## 8 C11orf87 Advanced Early Down  
## 20 CYP4A22 Advanced Early Down  
## 6 DDX3Y Advanced Early Down  
## 7 GSTT1 Advanced Early Down  
## 19 GYG2P1 Advanced Early Down  
## 12 HCRTR2 Advanced Early Down  
## 14 ITLN1 Advanced Early Down  
## 3 KDM5D Advanced Early Down  
## 1 KLK1 Advanced Early Down  
## 11 LINC00278 Advanced Early Down  
## 16 LINC01517 Advanced Early Down  
## 10 LOC101927136 Advanced Early Down  
## 18 MYCNOS Advanced Early Down  
## 13 RDH8 Advanced Early Down  
## 4 RPS4Y1 Advanced Early Down  
## 2 SRRM4 Advanced Early Down  
## 17 TBL1Y Advanced Early Down  
## 15 TTTY14 Advanced Early Down  
## 5 USP9Y Advanced Early Down  
## 9 UTY Advanced Early Down

Control state versus the Advanced DM state of down regulated genes

down3 <- as.data.frame(row.names(Top20\_down\_Ctrl\_Adv))  
colnames(down3) <- 'Gene'  
down3$dataType <- rep('Advanced Control Down',20)  
down3 <- down3[order(down3$Gene),]  
down3

## Gene dataType  
## 10 ATF3 Advanced Control Down  
## 20 C11orf87 Advanced Control Down  
## 11 DDX3Y Advanced Control Down  
## 15 DUSP1 Advanced Control Down  
## 7 EGR1 Advanced Control Down  
## 17 FER1L6-AS2 Advanced Control Down  
## 3 FOS Advanced Control Down  
## 1 FOSB Advanced Control Down  
## 16 KDM5D Advanced Control Down  
## 5 KLK1 Advanced Control Down  
## 9 MIR3189 Advanced Control Down  
## 2 NR4A1 Advanced Control Down  
## 6 NR4A2 Advanced Control Down  
## 13 NR4A3 Advanced Control Down  
## 19 RNR1 Advanced Control Down  
## 18 SPDYE7P Advanced Control Down  
## 4 SRRM4 Advanced Control Down  
## 14 TRIM50 Advanced Control Down  
## 8 USP9Y Advanced Control Down  
## 12 UTY Advanced Control Down

### What are the top 20 genes that have the most fold change in the ratio of healthy to renal disease gene expression? Even the inverse fold change of disease to healthy would

FC <- as.data.frame(row.names(Top20\_FC\_Early\_Control))  
colnames(FC) <- 'Gene'  
FC$dataType <- rep('Early Control Fold Change Up',20)  
FC <- FC[order(FC$Gene),]  
FC

## Gene dataType  
## 19 ACTA1 Early Control Fold Change Up  
## 2 ADIPOQ Early Control Fold Change Up  
## 16 ATF6B Early Control Fold Change Up  
## 5 CIDEA Early Control Fold Change Up  
## 1 CIDEC Early Control Fold Change Up  
## 17 DUSP13 Early Control Fold Change Up  
## 20 GABBR1 Early Control Fold Change Up  
## 13 GREM1 Early Control Fold Change Up  
## 12 LEP Early Control Fold Change Up  
## 14 LINC02133 Early Control Fold Change Up  
## 11 MB Early Control Fold Change Up  
## 15 MFAP5 Early Control Fold Change Up  
## 3 MYBPC1 Early Control Fold Change Up  
## 10 MYH7 Early Control Fold Change Up  
## 4 MYL2 Early Control Fold Change Up  
## 18 MYLK2 Early Control Fold Change Up  
## 8 SLN Early Control Fold Change Up  
## 6 TNNT1 Early Control Fold Change Up  
## 9 TUSC5 Early Control Fold Change Up  
## 7 XIRP2 Early Control Fold Change Up

FC1 <- as.data.frame(row.names(Top20\_FC\_Adv\_Early))  
colnames(FC1) <- 'Gene'  
FC1$dataType <- rep('Advanced Early Fold Change Up', 20)  
FC1 <- FC1[order(FC1$Gene),]  
FC1

## Gene dataType  
## 1 ADAMDEC1 Advanced Early Fold Change Up  
## 2 CLC Advanced Early Fold Change Up  
## 6 CLEC4C Advanced Early Fold Change Up  
## 14 COL6A5 Advanced Early Fold Change Up  
## 15 CSNK2B Advanced Early Fold Change Up  
## 12 CTLA4 Advanced Early Fold Change Up  
## 3 CXCL13 Advanced Early Fold Change Up  
## 17 GPR15 Advanced Early Fold Change Up  
## 19 GSTM1 Advanced Early Fold Change Up  
## 5 KCNQ2 Advanced Early Fold Change Up  
## 16 LINC01215 Advanced Early Fold Change Up  
## 18 LINC01924 Advanced Early Fold Change Up  
## 10 PI3 Advanced Early Fold Change Up  
## 20 PLA2G2D Advanced Early Fold Change Up  
## 13 PTCRA Advanced Early Fold Change Up  
## 4 RETN Advanced Early Fold Change Up  
## 7 STMN2 Advanced Early Fold Change Up  
## 9 TCL1A Advanced Early Fold Change Up  
## 8 TNFRSF13B Advanced Early Fold Change Up  
## 11 ZPLD1 Advanced Early Fold Change Up

FC2 <- as.data.frame(row.names(Top20\_FC\_Adv\_Control))  
colnames(FC2) <- 'Gene'  
FC2$dataType <- rep('Advanced Control Fold Change Up', 20)  
FC2 <- FC2[order(FC2$Gene),]  
FC2

## Gene dataType  
## 2 ADIPOQ Advanced Control Fold Change Up  
## 17 CADM3-AS1 Advanced Control Fold Change Up  
## 18 CFHR1 Advanced Control Fold Change Up  
## 1 CIDEC Advanced Control Fold Change Up  
## 13 CLC Advanced Control Fold Change Up  
## 5 CLEC4C Advanced Control Fold Change Up  
## 4 COL6A5 Advanced Control Fold Change Up  
## 16 DCANP1 Advanced Control Fold Change Up  
## 8 GREM1 Advanced Control Fold Change Up  
## 7 KCNQ2 Advanced Control Fold Change Up  
## 12 LINC00402 Advanced Control Fold Change Up  
## 11 LINC01426 Advanced Control Fold Change Up  
## 15 MYBPC1 Advanced Control Fold Change Up  
## 6 PLA2G2A Advanced Control Fold Change Up  
## 9 PTCRA Advanced Control Fold Change Up  
## 20 SCEL Advanced Control Fold Change Up  
## 14 SIRPG-AS1 Advanced Control Fold Change Up  
## 3 STMN2 Advanced Control Fold Change Up  
## 19 TIFAB Advanced Control Fold Change Up  
## 10 TUSC5 Advanced Control Fold Change Up

down <- rbind(down1,down2,down3)  
down <- down[unique(down$Gene),]  
  
up <- rbind(Up1, Up2, Up3)  
up <- up[unique(up$Gene),]  
  
fc <- rbind(FC, FC1, FC2)  
fc <- fc[unique(fc$Gene),]  
  
all <- rbind(down,up,fc)  
  
write.csv(all, 'all\_common\_up\_or\_fc\_and\_down.csv', row.names=FALSE)

#### Common genes to most fold change and up regulated gene expressions are:

head(all)

## Gene dataType  
## 11 ADAMTS4 Early Control Down  
## 7 ATF3 Early Control Down  
## 10 CCL3 Early Control Down  
## 19 CXCL2 Early Control Down  
## 20 CYR61 Early Control Down  
## 9 DUSP1 Early Control Down

tail(all)

## Gene dataType  
## 1122 CIDEC Advanced Control Fold Change Up  
## 132 CLC Advanced Control Fold Change Up  
## 521 CLEC4C Advanced Control Fold Change Up  
## 421 COL6A5 Advanced Control Fold Change Up  
## 162 DCANP1 Advanced Control Fold Change Up  
## 82 GREM1 Advanced Control Fold Change Up

### Now for some machine learning on predicting the dataType as one of these nine types of data sets

unique(all$dataType)

## [1] "Early Control Down" "Advanced Early Down"   
## [3] "Advanced Control Down" "early up regulated"   
## [5] "Advanced Early Up" "Advanced Control Up"   
## [7] "Early Control Fold Change Up" "Advanced Early Fold Change Up"   
## [9] "Advanced Control Fold Change Up"

### lets bring in the data from a kidney tumor study that was done on 12 kidney disease and 4 healthy kidney tissue samples for the top 20 down, up, and fold change increases of diseased to healthy comparisons. Note the gene expression values will differ, because this study used a different platform than the DM study in this Markdown file.

renal\_up20 <- read.csv('Up-regulated-20.csv', sep=',', header=TRUE,  
 na.strings=c('',' '))  
colnames(renal\_up20)[1] <- 'Gene'  
  
renal\_fc20 <- read.csv('Fold-Change-20.csv', sep=',', header=TRUE,  
 na.strings=c('',' '))  
colnames(renal\_fc20)[1] <- 'Gene'  
renal\_down20 <- read.csv('Down-regulated-20.csv', sep=',', header=TRUE,  
 na.strings=c('',' '))  
colnames(renal\_down20)[1] <- 'Gene'

What genes from the renal disease data were top genes for up, down, or increased fold change?

renal\_up20$dataType <- rep('Renal Disease Up regulated', 20)  
renal\_down20$dataType <- rep('Renal Disease Down regulated', 20)  
renal\_fc20$dataType <- rep('Renal Disease Increased Fold Change', 20)  
  
# keep only the Gene field and the dataType fields  
renal\_up20 <- renal\_up20[,c(1,22)]  
renal\_down20 <- renal\_down20[,c(1,22)]  
renal\_fc20 <- renal\_fc20[,c(1,21)]  
  
combined\_Renal <- rbind(renal\_up20,renal\_fc20,renal\_down20)  
  
Renal\_up <- combined\_Renal[0:40,]  
Renal\_down <- combined\_Renal[41:60,]

Lets focus on those genes that were up regulated in both the DM and Renal Disease samples including fold change, and those in common for DM and Renal Disease samples that were down regulated.

Renal\_up$Regulation <- rep('increased', 40)  
Renal\_down$Regulation <- rep('decreased',20)  
  
RENAL <- rbind(Renal\_up,Renal\_down)  
  
RENAL <- RENAL[,c(1,3)]  
RENAL$Disease <- rep('Renal Disease', length(RENAL$Gene))  
  
down <- all[grep('Down',all$dataType),]  
down$Regulation <- rep('decreased', length(down$Gene))  
up <- all[grep('Up', all$dataType),]  
up$Regulation <- rep('increased', length(up$Gene))  
  
DIABETES <- rbind(down, up)  
DIABETES <- DIABETES[,c(1,3)]  
DIABETES$Disease <- rep('Diabetes Mellitis', length(DIABETES$Gene))

Use the unique genes in the DIABETES data set and the RENAL data set and combine

diabetes <- DIABETES[unique(DIABETES$Gene),]  
renal <- RENAL[unique(RENAL$Gene),]  
  
Diabetes\_renal\_common\_genes <- merge(diabetes,renal, by.x='Gene', by.y='Gene')

Only the XIST gene is common to both diseases for most fold change, up, and down regulation. This gene increased and was up regulated in both diabetes mellitis and renal disease tissue gene expressions.

#### Use only the diabetes data set with the unique genes to each of the types of disease states compared to each other and the healthy state.Get the gene expressions for those samples by merging this diabetes data set by Gene with the Differential data set

Differential$Gene <- row.names(Differential)  
  
expressions <- merge(diabetes, Differential, by.x='Gene', by.y='Gene')  
Expressions <- expressions[unique(expressions$Gene),]  
Expressions <- Expressions[complete.cases(Expressions$Fold\_Change\_Early\_Control),]  
Expressions <- Expressions[!duplicated(Expressions$Gene),]

Let us satisfy the curiosity of knowing how the genes that were top expressed in fold change, up regulation, or down regulation from the renal disease data to this data to see how the diabetes data does.

curious <- merge(renal, Differential, by.x='Gene', by.y='Gene')  
Curious <- curious[unique(curious$Gene),]  
Curious <- Curious[complete.cases(Curious$DE\_ctrl\_adv),]  
Curious <- Curious[!duplicated(Curious$Gene),]  
  
KidneyGenesBothSets <- rbind(Expressions, Curious)  
write.csv(KidneyGenesBothSets, 'KidneyDisease\_Diabetes\_Renal.csv',  
 row.names=FALSE)  
KidneyGenesBothSets[,1:3]

## Gene Regulation Disease  
## 85 RETN increased Diabetes Mellitis  
## 45 GPR15 increased Diabetes Mellitis  
## 65 LINC01924 increased Diabetes Mellitis  
## 1 ACKR1 increased Diabetes Mellitis  
## 46 GREM1 increased Diabetes Mellitis  
## 2 ACTA1 increased Diabetes Mellitis  
## 21 COL6A5 increased Diabetes Mellitis  
## 66 LINC02133 increased Diabetes Mellitis  
## 86 RGS1 decreased Diabetes Mellitis  
## 3 ADAMDEC1 increased Diabetes Mellitis  
## 87 RPS4Y1 decreased Diabetes Mellitis  
## 48 GSTM1 increased Diabetes Mellitis  
## 100 ZFP36 decreased Diabetes Mellitis  
## 88 SLC2A3 decreased Diabetes Mellitis  
## 67 LOC101926964 increased Diabetes Mellitis  
## 68 LOC101927136 decreased Diabetes Mellitis  
## 69 LTF increased Diabetes Mellitis  
## 5 ADAMTS4 decreased Diabetes Mellitis  
## 89 SRRM4 decreased Diabetes Mellitis  
## 6 ADIPOQ increased Diabetes Mellitis  
## 95 TNFRSF13B increased Diabetes Mellitis  
## 8 ATF3 decreased Diabetes Mellitis  
## 9 ATF6B increased Diabetes Mellitis  
## 10 C11orf87 decreased Diabetes Mellitis  
## 70 MB increased Diabetes Mellitis  
## 90 STMN2 increased Diabetes Mellitis  
## 11 CCL19 increased Diabetes Mellitis  
## 96 TTTY14 decreased Diabetes Mellitis  
## 50 GSTT1 decreased Diabetes Mellitis  
## 72 MIR3189 decreased Diabetes Mellitis  
## 24 CSNK2B increased Diabetes Mellitis  
## 25 CTLA4 increased Diabetes Mellitis  
## 73 MYBPC1 increased Diabetes Mellitis  
## 74 MYCNOS decreased Diabetes Mellitis  
## 76 MYL2 increased Diabetes Mellitis  
## 27 CXCL13 increased Diabetes Mellitis  
## 13 CCL21 increased Diabetes Mellitis  
## 28 CXCL2 decreased Diabetes Mellitis  
## 29 CXCL6 increased Diabetes Mellitis  
## 52 HCRTR2 decreased Diabetes Mellitis  
## 30 CYP4A22 decreased Diabetes Mellitis  
## 31 CYR61 decreased Diabetes Mellitis  
## 97 USP9Y decreased Diabetes Mellitis  
## 77 MYLK2 increased Diabetes Mellitis  
## 32 DCANP1 increased Diabetes Mellitis  
## 78 NR4A1 decreased Diabetes Mellitis  
## 53 HLA-DPB1 increased Diabetes Mellitis  
## 54 IGLL5 increased Diabetes Mellitis  
## 14 CCL3 decreased Diabetes Mellitis  
## 56 ITLN1 decreased Diabetes Mellitis  
## 33 DDX3Y decreased Diabetes Mellitis  
## 57 JUNB decreased Diabetes Mellitis  
## 58 KCNQ2 increased Diabetes Mellitis  
## 98 UTY decreased Diabetes Mellitis  
## 15 CFHR1 increased Diabetes Mellitis  
## 16 CIDEA increased Diabetes Mellitis  
## 17 CIDEC increased Diabetes Mellitis  
## 34 DUSP1 decreased Diabetes Mellitis  
## 18 CLC increased Diabetes Mellitis  
## 35 DUSP13 increased Diabetes Mellitis  
## 19 CLEC4C increased Diabetes Mellitis  
## 36 DUSP2 decreased Diabetes Mellitis  
## 37 EGR1 decreased Diabetes Mellitis  
## 81 PI3 increased Diabetes Mellitis  
## 82 PLA2G2D increased Diabetes Mellitis  
## 38 EGR2 decreased Diabetes Mellitis  
## 39 EGR3 decreased Diabetes Mellitis  
## 40 FAM30A increased Diabetes Mellitis  
## 84 RDH8 decreased Diabetes Mellitis  
## 371 SLA increased Renal Disease  
## 381 SPP1 decreased Renal Disease  
## 391 TDRD1 increased Renal Disease  
## 22 GAPDH decreased Renal Disease  
## 12 ACTB decreased Renal Disease  
## 23 IFI27 increased Renal Disease  
## 210 ACTG1 decreased Renal Disease  
## 4 AIRN increased Renal Disease  
## 41 TPT1 decreased Renal Disease  
## 26 ITM2B decreased Renal Disease  
## 51 ANGPTL3 increased Renal Disease  
## 61 APP decreased Renal Disease  
## 7 ATP2B3 increased Renal Disease  
## 83 B4GALNT4 increased Renal Disease  
## 271 LBP increased Renal Disease  
## 43 XIST increased Renal Disease  
## 91 CADM3 increased Renal Disease  
## 101 CD24 decreased Renal Disease  
## 111 CD3D increased Renal Disease  
## 291 LOC389332 increased Renal Disease  
## 131 CEBPD increased Renal Disease  
## 141 CFH increased Renal Disease  
## 331 PKM decreased Renal Disease  
## 161 CXCR3 increased Renal Disease  
## 171 EEF1A1 decreased Renal Disease  
## 181 ENO1 decreased Renal Disease  
## 341 PRR7 increased Renal Disease  
## 351 RARRES1 increased Renal Disease  
## 191 FMO1 increased Renal Disease  
## 20 FTH1 decreased Renal Disease

#### Expressions is the Diabetes data set of this study and the KidneyGenesBothSets is the combined data set of Diabetes and Renal disease genes that can be compared side by side.

Using the Expressions data set, make the row names the Gene field while removing the statistically derived data from dplyer after writing to csv file.

write.csv(Expressions, 'Diabetes\_unique\_genes\_stats.csv', row.names=FALSE)  
row.names(Expressions) <- Expressions$Gene  
Expressions <- Expressions[,c(13:48)]

Do the same for the both data set of the two different studies’s prominent genes

row.names(KidneyGenesBothSets) <- KidneyGenesBothSets$Gene  
BothStudies <- KidneyGenesBothSets[,c(13:48)]

Now BothStudies is the data set of genes prominent in both diabetes and renal disease gene expression data from kidney tissue tumors and healthy kidney tissue.

#### Transpose both data sets to see which one will perform better with machine learning on this diabetes kidney disease data, and add in a sample type for Advanced, Early, or healthy diabetes kidney tissue sample type.

Expressions\_t <- as.data.frame(t(Expressions))  
type <- as.data.frame(c(rep('Advanced Diabetes',21),   
 rep('Early Diabetes',6),  
 rep('Healthy', 9)))  
colnames(type) <- 'TYPE'  
Expressions\_ML <- cbind(type, Expressions\_t)

BothStudies\_t <- as.data.frame(t(BothStudies))  
BothStudies\_ML <- cbind(type, BothStudies\_t)

Write both the machine learning ready data sets to csv

write.csv(Expressions\_ML, 'ML\_ready\_DiabetesGenes.csv', row.names=TRUE)  
write.csv(BothStudies\_ML, 'ML\_ready\_DiabetesAndRenalGenes.csv',  
 row.names=TRUE)

The data set that will be used for Machine Learning will predict if the sample is renal disease or healthy. The samples will have to be randomized into 80% train and 20% test

library(caret)  
library(randomForest)  
library(MASS)  
library(gbm)  
library(dplyr)

set.seed(189678345)

#### The Diabetes genes are used first to predict type of sample as Advanced, Early, or healthy diabetes cases

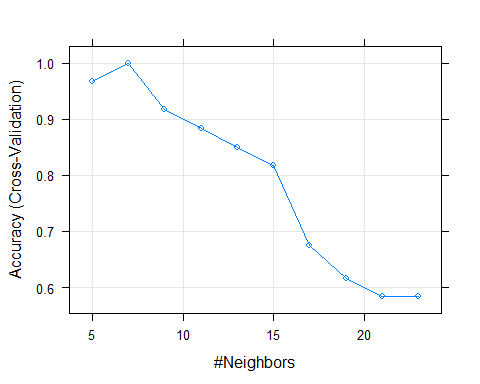
inTrain <- createDataPartition(y=Expressions\_ML$TYPE, p=0.8, list=FALSE)  
  
trainingSet <- Expressions\_ML[inTrain,]  
testingSet <- Expressions\_ML[-inTrain,]

### KNN

system.time(knnMod <- train(TYPE ~ .,  
 method='knn', preProcess=c('center','scale'),  
 tuneLength=10, trControl=trainControl(method='cv'), data=trainingSet))

## user system elapsed   
## 2.39 0.11 2.97

plot(knnMod)



The predicted results with KNN

predKNN <- predict(knnMod, testingSet)  
predKNN

## [1] Advanced Diabetes Advanced Diabetes Advanced Diabetes Advanced Diabetes  
## [5] Early Diabetes Healthy   
## Levels: Advanced Diabetes Early Diabetes Healthy

The actual values in the testing set

testingSet$TYPE

## [1] Advanced Diabetes Advanced Diabetes Advanced Diabetes Advanced Diabetes  
## [5] Early Diabetes Healthy   
## Levels: Advanced Diabetes Early Diabetes Healthy

The accuracy with K-Nearest Neighbors (KNN)

accuracy <- (sum(predKNN==testingSet$TYPE)/length(predKNN))\*100.00  
acc <- paste(paste('The KNN accuracy is ',accuracy, sep=''),'%',sep='')  
acc

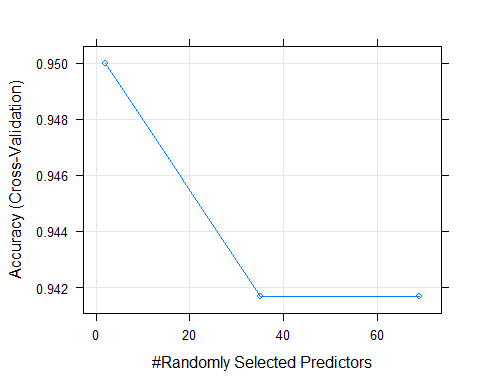
## [1] "The KNN accuracy is 100%"

### Random Forest

system.time(rfMod <- train(TYPE ~., method='rf', data=(trainingSet),   
 trControl=trainControl(method='cv'), number=5))

## user system elapsed   
## 2.47 0.03 2.93

plot(rfMod)

 The predicted Random Forest results and the actual results

predRF <- predict(rfMod, testingSet)  
predRF

## [1] Advanced Diabetes Advanced Diabetes Advanced Diabetes Advanced Diabetes  
## [5] Early Diabetes Healthy   
## Levels: Advanced Diabetes Early Diabetes Healthy

testingSet$TYPE

## [1] Advanced Diabetes Advanced Diabetes Advanced Diabetes Advanced Diabetes  
## [5] Early Diabetes Healthy   
## Levels: Advanced Diabetes Early Diabetes Healthy

The accuracy with Random Forest

accuracy <- (sum(predRF==testingSet$TYPE)/length(predRF))\*100.00  
acc <- paste(paste('The Random Forest accuracy is ',accuracy, sep=''),'%',sep='')  
acc

## [1] "The Random Forest accuracy is 100%"

#### Lastly, the data set with genes prominent in this diabetes study and the study on renal disease on kidney tumor samples will be used to identify the same type of sample derived as Advanced Diabetes, Early Diabetes, or Healthy.

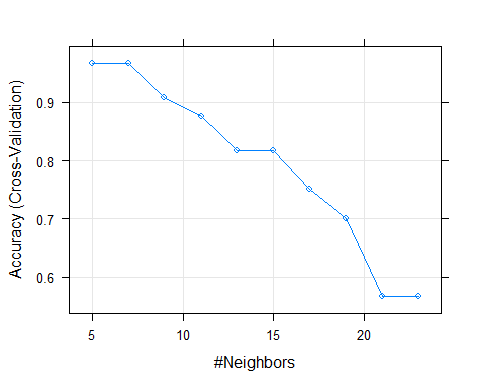
inTrain <- createDataPartition(y=BothStudies\_ML$TYPE, p=0.8, list=FALSE)  
  
trainingSet <- BothStudies\_ML[inTrain,]  
testingSet <- BothStudies\_ML[-inTrain,]

### KNN

system.time(knnMod <- train(TYPE ~ .,  
 method='knn', preProcess=c('center','scale'),  
 tuneLength=10, trControl=trainControl(method='cv'), data=trainingSet))

## user system elapsed   
## 2.52 0.01 2.72

plot(knnMod)



The predicted results with KNN

predKNN <- predict(knnMod, testingSet)  
predKNN

## [1] Advanced Diabetes Advanced Diabetes Advanced Diabetes Advanced Diabetes  
## [5] Early Diabetes Healthy   
## Levels: Advanced Diabetes Early Diabetes Healthy

The actual values in the testing set

testingSet$TYPE

## [1] Advanced Diabetes Advanced Diabetes Advanced Diabetes Advanced Diabetes  
## [5] Early Diabetes Healthy   
## Levels: Advanced Diabetes Early Diabetes Healthy

The accuracy with K-Nearest Neighbors (KNN)

accuracy <- (sum(predKNN==testingSet$TYPE)/length(predKNN))\*100.00  
acc <- paste(paste('The KNN accuracy is ',accuracy, sep=''),'%',sep='')  
acc

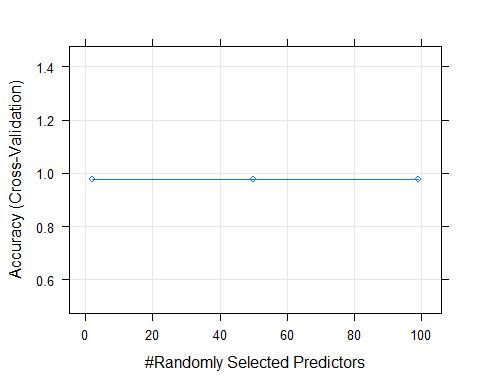
## [1] "The KNN accuracy is 100%"

### Random Forest

system.time(rfMod <- train(TYPE ~., method='rf', data=(trainingSet),   
 trControl=trainControl(method='cv'), number=5))

## user system elapsed   
## 3.05 0.05 3.24

plot(rfMod)

 The predicted Random Forest results and the actual results

predRF <- predict(rfMod, testingSet)  
predRF

## [1] Advanced Diabetes Advanced Diabetes Advanced Diabetes Advanced Diabetes  
## [5] Early Diabetes Healthy   
## Levels: Advanced Diabetes Early Diabetes Healthy

testingSet$TYPE

## [1] Advanced Diabetes Advanced Diabetes Advanced Diabetes Advanced Diabetes  
## [5] Early Diabetes Healthy   
## Levels: Advanced Diabetes Early Diabetes Healthy

The accuracy with Random Forest

accuracy <- (sum(predRF==testingSet$TYPE)/length(predRF))\*100.00  
acc <- paste(paste('The Random Forest accuracy is ',accuracy, sep=''),'%',sep='')  
acc

## [1] "The Random Forest accuracy is 100%"

### Both gene sets are able to predict which type of sample the observation is from of either Advanced Diabetes, Early Diabetes, or Healthy kidney tissue samples.

Some text mining to build word clouds based on the NCBI gene summaries for these genes would be useful to show what genes are expressed more and see if their gene functions in the human body show up in the word clouds for up regulated (including increased fold change) and down regulated.

both\_ML <- KidneyGenesBothSets[,1:2]  
diabetes\_ML <- expressions[,1:2]  
diabetes\_ML <- diabetes\_ML[unique(diabetes\_ML$Gene),]  
  
diabetes\_down <- diabetes\_ML[grep('decreased',diabetes\_ML$Regulation),]  
diabetes\_up <- diabetes\_ML[grep('increased',diabetes\_ML$Regulation),]  
both\_down <- both\_ML[grep('decreased', both\_ML$Regulation),]  
both\_up <- both\_ML[grep('increased', both\_ML$Regulation),]  
  
write.csv(diabetes\_down, 'diabetes\_down.csv', row.names=FALSE)  
write.csv(diabetes\_up, 'diabetes\_up.csv', row.names=FALSE)  
write.csv(both\_down, 'both\_down.csv', row.names=FALSE)  
write.csv(both\_up, 'both\_up.csv', row.names=FALSE)

#### The gene functions were added to the diabetes\_up and diabetes\_down csv files from genecards.org

# These files were modified in Excel to add the gene names and change the file #name so that when this script is ran again the files aren't erased or #replaced with the empty versions.  
  
summ\_up <- read.csv('diabetes\_up1.csv', sep=',', header=TRUE,  
 na.strings=c('',' '))  
summ\_down <- read.csv('diabetes\_down1.csv', sep=',', header=TRUE,  
 na.strings=c('',' '))

Remove the duplicate genes in summ\_up and combine the genes that information was added to in the diabetes data with the ‘both’ versions and write out to replace NAs in Excel manually from genecards.org. The files will be renamed so that they aren’t replaced with empty information when this script runs again.

summ\_up <- summ\_up[!duplicated(summ\_up$Gene),]  
summ\_down <- summ\_down[!duplicated(summ\_down$Gene),]  
  
both\_up <- read.csv('both\_up.csv', sep=',', header=TRUE,   
 na.strings=c('',' '))  
both\_down <- read.csv('both\_down.csv', sep=',', header=TRUE,   
 na.strings=c('',' '))  
  
up\_both <- merge(both\_up, summ\_up, by.x='Gene', by.y='Gene',   
 all.x=TRUE)  
down\_both <- merge(both\_down, summ\_down, by.x='Gene', by.y='Gene',  
 all.x=TRUE)  
up\_both <- up\_both[,c(1,2,4)]  
down\_both <- down\_both[,c(1,2,4)]  
colnames(up\_both) <- c('Gene','Regulation', 'geneCardsFunction')  
colnames(down\_both) <- c('Gene','Regulation', 'geneCardsFunction')  
  
write.csv(up\_both, 'up\_both.csv', row.names=FALSE)  
write.csv(down\_both, 'down\_both.csv', row.names=FALSE)

The files for the both up and both down genes have been filled in manually and renamed to add a 1 at the end. Read those files in now, so that we will have the four data sets to work with to build our word clouds with the genecards/uniprot gene summaries. The Entrez summary was left out because it is too scientific to put in interpretable word clouds based on gene functions for up and down regulated genes in kidney disease samples related to diabetes and/or renal disease.

both\_up <- read.csv('up\_both1.csv', sep=',', header=TRUE,  
 na.strings=c('',' '))  
both\_down <- read.csv('down\_both1.csv', sep=',', header=TRUE,  
 na.strings=c('',' '))

rm(down\_both);rm(up\_both)

#### Now we have the completed gene functions attached to the four separate data sets we will be using for our text mining.

Those being: summ\_up and summ\_down for the diabetic kidney disease targets

both\_up and both\_down for the combination of diabetes and renal disease kidney disease targets

Use lemmatization on the data sets with filled in gene summaries

library(tm)  
library(SnowballC)  
library(wordcloud)  
library(ggplot2)  
library(textstem)

lemma <- lemmatize\_strings(summ\_up$geneCardsFunction, dictionary=lexicon::hash\_lemmas)  
  
Lemma <- as.data.frame(lemma)  
Lemma <- cbind(Lemma, summ\_up)  
  
colnames(Lemma)[1] <- 'lemmatized\_summary'  
  
write.csv(Lemma, 'Lemmatized\_Diabetes\_Up.csv', row.names=FALSE)

lemma1 <- lemmatize\_strings(summ\_down$geneCardsFunction, dictionary=lexicon::hash\_lemmas)  
  
Lemma1 <- as.data.frame(lemma1)  
Lemma1 <- cbind(Lemma1, summ\_down)  
  
colnames(Lemma1)[1] <- 'lemmatized\_summary'  
  
write.csv(Lemma1, 'Lemmatized\_Diabetes\_Down.csv', row.names=FALSE)

lemma2 <- lemmatize\_strings(both\_up$geneCardsFunction, dictionary=lexicon::hash\_lemmas)  
  
Lemma2 <- as.data.frame(lemma2)  
Lemma2 <- cbind(Lemma2, both\_up)  
  
colnames(Lemma2)[1] <- 'lemmatized\_summary'  
  
write.csv(Lemma2, 'Lemmatized\_Diabetes\_Renal\_Up.csv', row.names=FALSE)

lemma3 <- lemmatize\_strings(both\_down$geneCardsFunction, dictionary=lexicon::hash\_lemmas)  
  
Lemma3 <- as.data.frame(lemma3)  
Lemma3 <- cbind(Lemma3, both\_down)  
  
colnames(Lemma3)[1] <- 'lemmatized\_summary'  
  
write.csv(Lemma3, 'Lemmatized\_Diabetes\_Renal\_Down.csv', row.names=FALSE)

## Up regulated and increased fold change genes in diabetes mellitis (DM) samples

dir.create('./DM-Up-Lemma')  
  
ea <- as.character(Lemma$lemmatized\_summary)  
setwd('./DM-Up-Lemma')  
  
for (j in 1:length(ea)){  
 write(ea[j], paste(paste('Up',j, sep='.'), '.txt', sep=''))  
}  
setwd('../')

DiabetesUp <- Corpus(DirSource("DM-Up-Lemma"))  
  
DiabetesUp

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 39

DiabetesUp <- tm\_map(DiabetesUp, removePunctuation)  
DiabetesUp <- tm\_map(DiabetesUp, removeNumbers)  
DiabetesUp <- tm\_map(DiabetesUp, tolower)  
DiabetesUp <- tm\_map(DiabetesUp, removeWords, stopwords("english"))  
DiabetesUp <- tm\_map(DiabetesUp, stripWhitespace)  
  
dtmDiabetesUp <- DocumentTermMatrix(DiabetesUp)  
dtmDiabetesUp

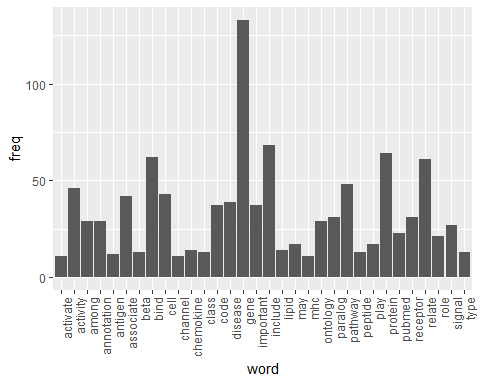
## <<DocumentTermMatrix (documents: 39, terms: 924)>>  
## Non-/sparse entries: 2036/34000  
## Sparsity : 94%  
## Maximal term length: 24  
## Weighting : term frequency (tf)

freq <- colSums(as.matrix(dtmDiabetesUp))  
  
FREQ <- data.frame(freq)  
ord <- order(freq, decreasing=TRUE)  
  
freq[head(ord, 25)]

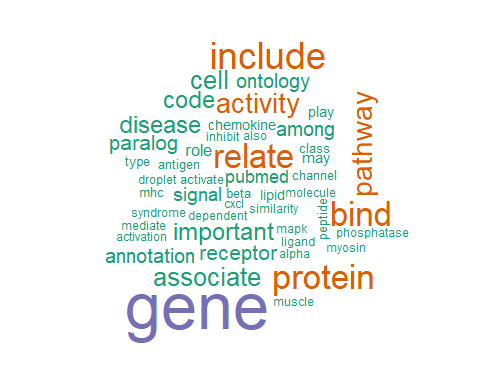
## gene include protein bind relate pathway activity   
## 133 68 64 62 61 48 46   
## cell associate disease code important receptor paralog   
## 43 42 39 37 37 31 31   
## among annotation ontology signal pubmed role may   
## 29 29 29 27 23 21 17   
## play chemokine lipid peptide   
## 17 14 14 13

### Up regulated genes

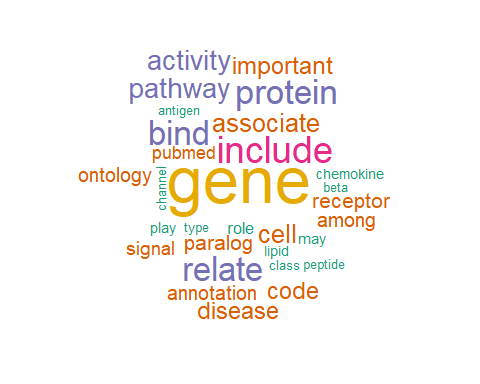
wf <- data.frame(word=names(freq), freq=freq)  
p <- ggplot(subset(wf, freq>10), aes(word, freq))  
p <- p + geom\_bar(stat= 'identity')   
p <- p + theme(axis.text.x=element\_text(angle=90, hjust=1))   
p



wordcloud(names(freq), freq, min.freq=9,colors=brewer.pal(3,'Dark2'))



wordcloud(names(freq), freq, max.words=30,colors=brewer.pal(6,'Dark2'))



## Down regulated genes in diabetes mellitis (DM) samples

dir.create('./DM-Down-Lemma')  
  
ea <- as.character(Lemma1$lemmatized\_summary)  
setwd('./DM-Down-Lemma')  
  
for (j in 1:length(ea)){  
 write(ea[j], paste(paste('Down',j, sep='.'), '.txt', sep=''))  
}  
setwd('../')

DiabetesDown <- Corpus(DirSource("DM-Down-Lemma"))  
  
DiabetesDown

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 30

DiabetesDown <- tm\_map(DiabetesDown, removePunctuation)  
DiabetesDown <- tm\_map(DiabetesDown, removeNumbers)  
DiabetesDown <- tm\_map(DiabetesDown, tolower)  
DiabetesDown <- tm\_map(DiabetesDown, removeWords, stopwords("english"))  
DiabetesDown <- tm\_map(DiabetesDown, stripWhitespace)  
  
dtmDiabetesDown <- DocumentTermMatrix(DiabetesDown)  
dtmDiabetesDown

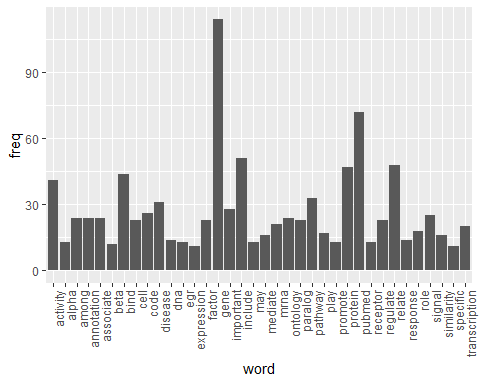
## <<DocumentTermMatrix (documents: 30, terms: 780)>>  
## Non-/sparse entries: 1544/21856  
## Sparsity : 93%  
## Maximal term length: 23  
## Weighting : term frequency (tf)

freq <- colSums(as.matrix(dtmDiabetesDown))  
  
FREQ <- data.frame(freq)  
ord <- order(freq, decreasing=TRUE)  
  
freq[head(ord, 25)]

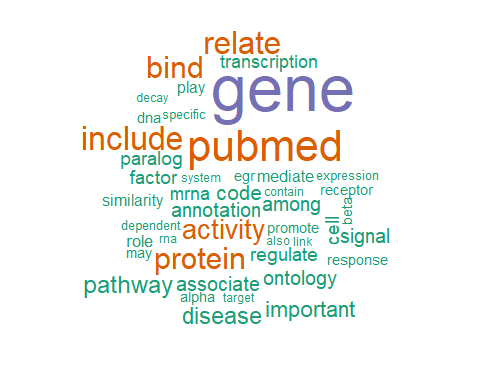
## gene pubmed include relate protein   
## 114 72 51 48 47   
## bind activity pathway disease important   
## 44 41 33 31 28   
## code signal among annotation associate   
## 26 25 24 24 24   
## ontology cell paralog regulate factor   
## 24 23 23 23 23   
## mrna transcription role play similarity   
## 21 20 18 17 16

### Down regulated genes

wf <- data.frame(word=names(freq), freq=freq)  
p <- ggplot(subset(wf, freq>10), aes(word, freq))  
p <- p + geom\_bar(stat= 'identity')   
p <- p + theme(axis.text.x=element\_text(angle=90, hjust=1))   
p



wordcloud(names(freq), freq, min.freq=9,colors=brewer.pal(3,'Dark2'))



wordcloud(names(freq), freq, max.words=30,colors=brewer.pal(6,'Dark2'))



### Both the Renal and Diabetes genes most expressed in Kidney tumor tissue gene expression data

dir.create('./DM-RD-Up-Lemma')  
  
ea <- as.character(Lemma2$lemmatized\_summary)  
setwd('./DM-RD-Up-Lemma')  
  
for (j in 1:length(ea)){  
 write(ea[j], paste(paste('BothUp',j, sep='.'), '.txt', sep=''))  
}  
setwd('../')

BothUp <- Corpus(DirSource("DM-RD-Up-Lemma"))  
  
BothUp

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 57

BothUp <- tm\_map(BothUp, removePunctuation)  
BothUp <- tm\_map(BothUp, removeNumbers)  
BothUp <- tm\_map(BothUp, tolower)  
BothUp <- tm\_map(BothUp, removeWords, stopwords("english"))  
BothUp <- tm\_map(BothUp, stripWhitespace)  
  
dtmBothUp <- DocumentTermMatrix(BothUp)  
dtmBothUp

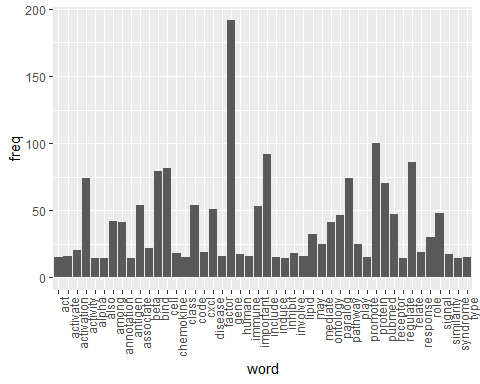
## <<DocumentTermMatrix (documents: 57, terms: 1223)>>  
## Non-/sparse entries: 3044/66667  
## Sparsity : 96%  
## Maximal term length: 31  
## Weighting : term frequency (tf)

freq <- colSums(as.matrix(dtmBothUp))  
  
FREQ <- data.frame(freq)  
ord <- order(freq, decreasing=TRUE)  
  
freq[head(ord, 25)]

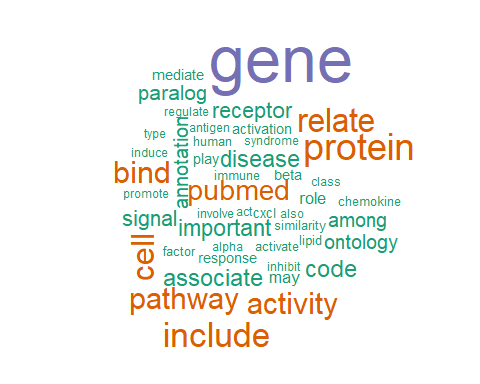
## gene protein include relate cell bind activity   
## 192 100 92 86 81 79 74   
## pathway pubmed associate code important disease signal   
## 74 70 54 54 53 51 48   
## receptor paralog among annotation ontology may role   
## 47 46 42 41 41 32 30   
## play mediate beta activation   
## 25 25 22 20

### Down regulated genes

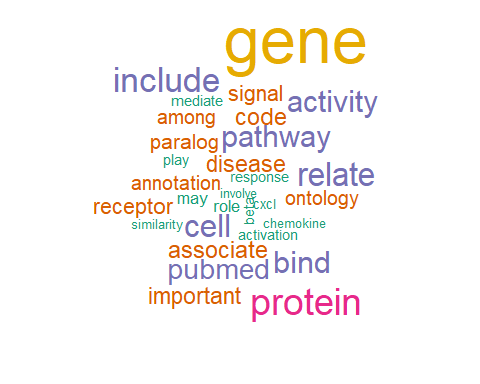
wf <- data.frame(word=names(freq), freq=freq)  
p <- ggplot(subset(wf, freq>13), aes(word, freq))  
p <- p + geom\_bar(stat= 'identity')   
p <- p + theme(axis.text.x=element\_text(angle=90, hjust=1))   
p



wordcloud(names(freq), freq, min.freq=14,colors=brewer.pal(3,'Dark2'))



wordcloud(names(freq), freq, max.words=30,colors=brewer.pal(6,'Dark2'))



### Both the Renal and Diabetes genes Least expressed in Kidney tumor tissue gene expression data

dir.create('./DM-RD-Down-Lemma')  
  
ea <- as.character(Lemma2$lemmatized\_summary)  
setwd('./DM-RD-Down-Lemma')  
  
for (j in 1:length(ea)){  
 write(ea[j], paste(paste('BothDown',j, sep='.'), '.txt', sep=''))  
}  
setwd('../')

BothDown <- Corpus(DirSource("DM-RD-Down-Lemma"))  
  
BothDown

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 57

BothDown <- tm\_map(BothDown, removePunctuation)  
BothDown <- tm\_map(BothDown, removeNumbers)  
BothDown <- tm\_map(BothDown, tolower)  
BothDown <- tm\_map(BothDown, removeWords, stopwords("english"))  
BothDown <- tm\_map(BothDown, stripWhitespace)  
  
dtmBothDown <- DocumentTermMatrix(BothDown)  
dtmBothDown

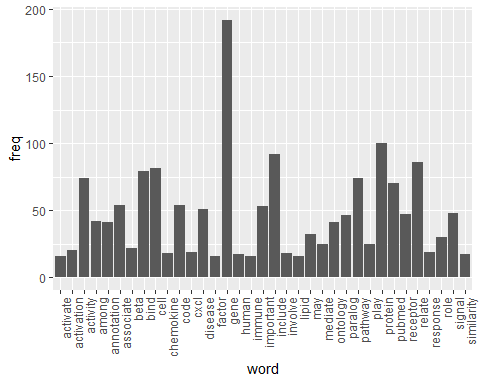
## <<DocumentTermMatrix (documents: 57, terms: 1223)>>  
## Non-/sparse entries: 3044/66667  
## Sparsity : 96%  
## Maximal term length: 31  
## Weighting : term frequency (tf)

freq <- colSums(as.matrix(dtmBothDown))  
  
FREQ <- data.frame(freq)  
ord <- order(freq, decreasing=TRUE)  
  
freq[head(ord, 25)]

## gene protein include relate cell bind activity   
## 192 100 92 86 81 79 74   
## pathway pubmed associate code important disease signal   
## 74 70 54 54 53 51 48   
## receptor paralog among annotation ontology may role   
## 47 46 42 41 41 32 30   
## play mediate beta activation   
## 25 25 22 20

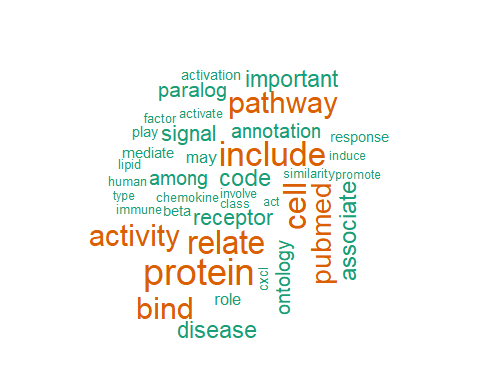
### Down regulated genes

wf <- data.frame(word=names(freq), freq=freq)  
p <- ggplot(subset(wf, freq>15), aes(word, freq))  
p <- p + geom\_bar(stat= 'identity')   
p <- p + theme(axis.text.x=element\_text(angle=90, hjust=1))   
p



wordcloud(names(freq), freq, min.freq=15,colors=brewer.pal(3,'Dark2'))

## Warning in wordcloud(names(freq), freq, min.freq = 15, colors = brewer.pal(3, :  
## gene could not be fit on page. It will not be plotted.



wordcloud(names(freq), freq, max.words=30,colors=brewer.pal(6,'Dark2'))

