**GSE18920 Exon Array Analysis**

We are using GSE18920 obtained from GEO. Using APT, we can obtain normalized exon-level summaries, normalized gene-level summaries, and DABG stats. The normalization method is RMA-sketch for both exon and gene data.

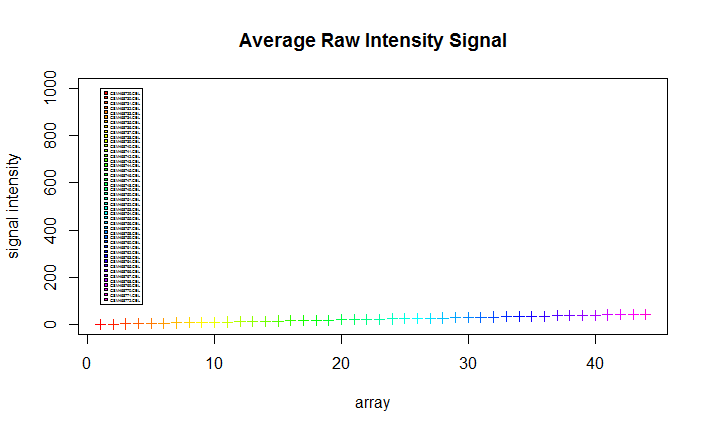
|  |
| --- |
| PS D:\Final Project> apt-probeset-summarize -p .\HuEx-1\_0-st-v2.r2.pgf -c .\HuEx-1\_0-st-v2.r2.clf -b .\HuEx-1\_0-st-v2.r2.antigenomic.bgp --qc-probesets .\HuEx-1\_0-st-v2.r2.qcc -s .\HuEx-1\_0-st-v2.r2.dt1.hg18.core.ps -a rma-sketch -o .\output\exon --cel-files .\cel\_files2.txt  Read 44 cel files from: cel\_files2.txt  Running ProbesetSummarizeEngine...  Opening clf file: HuEx-1\_0-st-v2.r2.clf  Opening pgf file: HuEx-1\_0-st-v2.r2.pgf  Setting analysis info.  Reading and pre-processing 44 cel files............................................Done. (1 min)  Processing 1 chipstream.  Computing sketch normalization for 44 cel datasets............................................Done. (0.41 min)  Applying sketch normalization to 44 cel datasets............................................Done. (1.03 min)  Finalizing 1 chipstream.  Processing Probesets.....................Done. (0.96 min)  Flushing output reporters. Finalizing output.  Done.  Run took approximately: 7.45 minutes.  Done running ProbesetSummarizeEngine. |
| PS D:\Final Project> apt-probeset-summarize -p .\HuEx-1\_0-st-v2.r2.pgf -c .\HuEx-1\_0-st-v2.r2.clf -b .\HuEx-1\_0-st-v2.r2.antigenomic.bgp --qc-probesets .\HuEx-1\_0-st-v2.r2.qcc -m .\HuEx-1\_0-st-v2.r2.dt1.hg18.core.mps -a rma-sketch -o .\output\gene --cel-files .\cel\_files2.txt  Read 44 cel files from: cel\_files2.txt  Running ProbesetSummarizeEngine...  Opening clf file: HuEx-1\_0-st-v2.r2.clf  Opening pgf file: HuEx-1\_0-st-v2.r2.pgf  Setting analysis info.  Reading and pre-processing 44 cel files............................................Done. (0.93 min)  Processing 1 chipstream.  Computing sketch normalization for 44 cel datasets............................................Done. (0.45 min)  Applying sketch normalization to 44 cel datasets............................................Done. (1.16 min)  Finalizing 1 chipstream.  Processing Probesets.....................Done. (0.53 min)  Flushing output reporters. Finalizing output.  Done.  Run took approximately: 7.48 minutes.  Done running ProbesetSummarizeEngine. |
| PS D:\Final Project> apt-probeset-summarize -p .\HuEx-1\_0-st-v2.r2.pgf -c .\HuEx-1\_0-st-v2.r2.clf -b .\HuEx-1\_0-st-v2.r2.antigenomic.bgp --qc-probesets .\HuEx-1\_0-st-v2.r2.qcc -s .\HuEx-1\_0-st-v2.r2.dt1.hg18.core.ps -a dabg -o .\output\exon --cel-files .\cel\_files2.txt  Read 44 cel files from: cel\_files2.txt  Running ProbesetSummarizeEngine...  Opening clf file: HuEx-1\_0-st-v2.r2.clf  Opening pgf file: HuEx-1\_0-st-v2.r2.pgf  Setting analysis info.  Reading and pre-processing 44 cel files............................................Done. (0.38 min)  Processing Probesets.....................Done. (0.76 min)  Flushing output reporters. Finalizing output.  Done.  Run took approximately: 2.26 minutes.  Done running ProbesetSummarizeEngine. |

We continue by conducting QC in R:

|  |
| --- |
| > setwd("D:/Final Project")  > exon = read.table("D:/Final Project/output/exon/rma-sketch.summary.txt", header=T, quote="\"", row.names=1)  > gene = read.table("D:/Final Project/output/gene/rma-sketch.summary.txt", header=T, quote="\"", row.names=1)  > dabg = read.table("D:/Final Project/output/exon/dabg.summary.txt", header=T, quote="\"", row.names=1)  > qc = read.table("D:/Final Project/output/exon/rma-sketch.report.txt")  > palette(rainbow(50)) |

Let’s look at a plot of the average raw intensity signal. The color legend will be the same for all the QC plots.

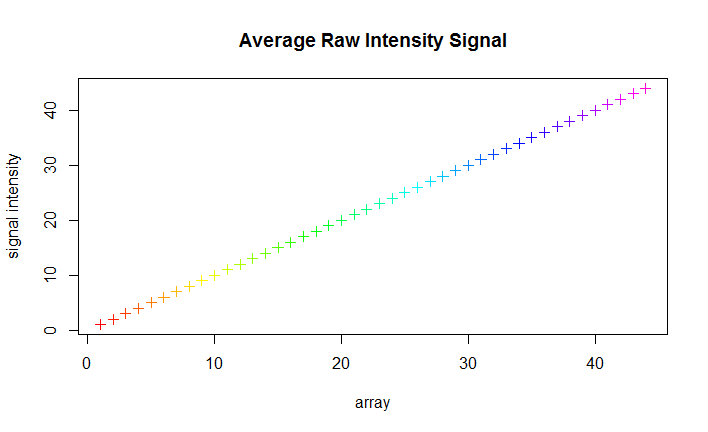
|  |
| --- |
| > plot(1:44,qc$pm\_mean,ylim=c(0,1000),xlab="array",ylab="signal intensity",main="Average Raw Intensity Signal",col=c(1:44),pch=3)  > legend(1,1000,colnames(exon),fill=c(1:44),cex=0.25) |



Examination of the average raw intensity signal doesn’t reveal any outliers.

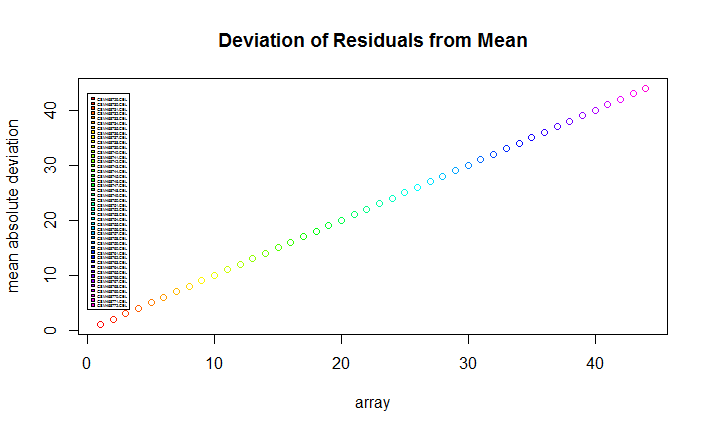
Let’s take a closer look

|  |
| --- |
| > plot(1:44,qc$pm\_mean,xlab="array",ylab="signal intensity",main="Average Raw Intensity Signal",col=c(1:44),pch=3) |



Let’s plot the deviations of residuals from mean:

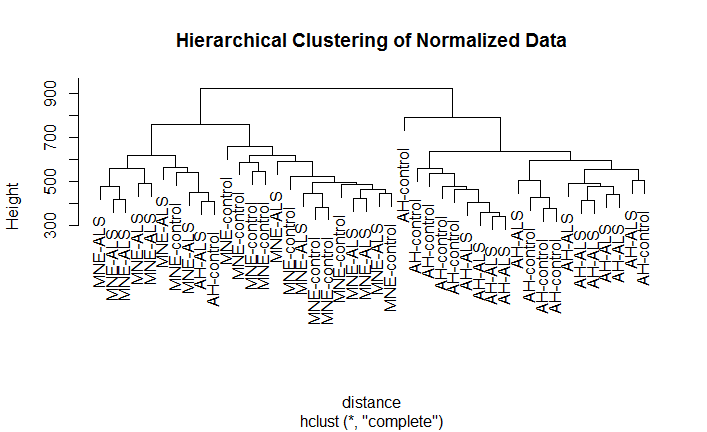
|  |
| --- |
| > plot(1:44,qc$all\_probeset\_mad\_residual\_mean,xlab="array",ylab="mean absolute deviation",main="Deviation of Residuals from Mean",col=c(1:44))  > legend(0,43,colnames(exon),fill=c(1:44),cex=0.25) |



Examination of the deviations of residuals from mean doesn’t reveal any outliers.

Finally, let’s plot a hierarchical clustering of the normalized data:

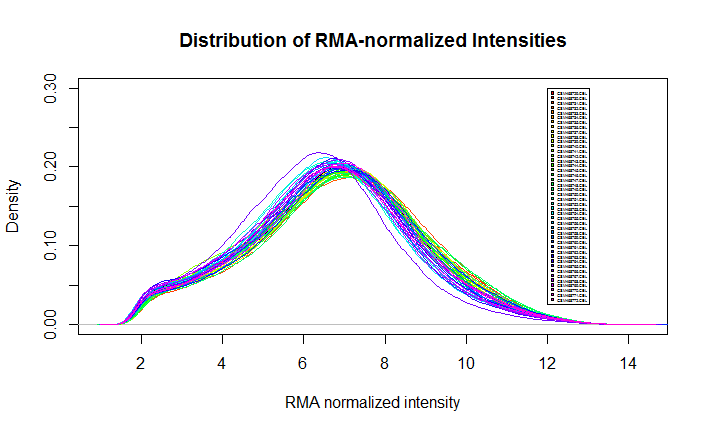
|  |
| --- |
| > cel\_files = read.delim("D:/Final Project/cel\_files.txt")  > grouping = cel\_files$group\_id  > dist = dist(t(exon))  > plot(hclust(dist),main="Hierarchical Clustering of Normalized Data",labels=grouping,xlab="distance") |



There does not seem to be any outlier within the arrays.

Next, we examine a distribution of normalized intensities:

|  |
| --- |
| > plot(density(exon[,1]),main="Distribution of RMA-normalized Intensities",xlab="RMA normalized intensity",ylim=c(0,0.3))  > for(i in 2:ncol(exon)) {lines(density(exon[,i]),col=i)}  > legend(12,0.3,colnames(exon),fill=c(1:44), cex=0.25) |



Everything checks out. Let’s filter the probesets.

First, we filter for undetected probesets:

|  |
| --- |
| > d.mne\_als = apply(dabg[,cel\_files$group\_id == "MNE-ALS "],1,function(x){length(which(x<0.05))})  > d.mne\_control = apply(dabg[,cel\_files$group\_id == "MNE-control "],1,function(x){length(which(x<0.05))})  > d.ah\_als = apply(dabg[,cel\_files$group\_id == "AH-ALS "],1,function(x){length(which(x<0.05))})  > d.ah\_control = apply(dabg[,cel\_files$group\_id == "AH-control "],1,function(x){length(which(x<0.05))})  > exon.filtered = exon[sort(union(union(union(which(d.ah\_als>=3), which(d.ah\_control>=3)),which(d.mne\_als>=3)),which(d.mne\_control>=3))),]  > dim(exon.filtered)  [1] 259063 44  > dim(exon)[1]-dim(exon.filtered)[1]  [1] 28266 |

28266 probesets were removed.

Next, filter for cross-hybridizing probesets:

|  |
| --- |
| > ann = read.csv("HuEx-1\_0-st-v2.na33.1.hg19.probeset.csv")  > ann.core = ann[match(row.names(exon.filtered),ann[,1]),]  > dim(ann)  [1] 1432143 39  > dim(ann.core)  [1] 259063 39  > dim(ann)[1]-dim(ann.core)[1]  [1] 1173080  > keep = which(ann.core$crosshyb\_type==1)  > ids = ann.core[keep,1]  > exon.fil2 = exon.filtered[match(ids,rownames(exon.filtered)),]  > dim(exon.fil2)  [1] 210909 44  > write.table(exon.fil2,"exon\_filtered.txt",sep="\t",quote=F,row.names=T) |

Finally, we filter for genes undetected in all of the groups:

|  |
| --- |
| > dim(dabg.core)  [1] 287329 44  > length(intersect(row.names(dabg.core),ann[,1]))  [1] 287329  > uniq2 = intersect(row.names(dabg.core),ann[,1])  > dabg.core2 = dabg.core[match(uniq2,row.names(dabg.core)),]  > dim(dabg.core2)  [1] 287329 44  > dabg.core2[,45] = ann[match(row.names(dabg.core2),ann$probeset\_id),7]  > gene.ids = unique(dabg.core2[,45])  > length(gene.ids)  [1] 18727  > gene.detection = matrix(nrow=length(unique(dabg.core2[,45])), ncol=44)  > rownames(gene.detection) = gene.ids  > colnames(gene.detection) = colnames(gene)  > for (i in 1:44) {gene.detection[,i] = tapply(dabg.core2[,i],dabg.core2[,45],function(x){length(which(x<0.05))/length(x)})}  > d.genes.ah\_als = apply(gene.detection[,cel\_files$group\_id == "AH-ALS "],1,function(x){length(which(x>=0.5))})  > d.genes.ah\_control = apply(gene.detection[,cel\_files$group\_id == "AH-control "],1,function(x){length(which(x>=0.5))})  > d.genes.mne\_control = apply(gene.detection[,cel\_files$group\_id == "MNE-control "],1,function(x){length(which(x>=0.5))})  > d.genes.mne\_als = apply(gene.detection[,cel\_files$group\_id == "MNE-ALS "],1,function(x){length(which(x>=0.5))})  > keep.genes = which((d.genes.ah\_als>=3)&(d.genes.ah\_control>=3)&(d.genes.mne\_als>=3)&(d.genes.mne\_control>=3))  > length(keep.genes)  [1] 16111  > keep.gene.ids = rownames(gene.detection)[keep.genes]  > length(intersect(rownames(gene),keep.gene.ids))  [1] 15408  > rmna = match(keep.gene.ids,rownames(gene))  > rmna = rmna[-which(is.na(rmna)==T)]  > gene.filtered = gene[rmna,]  > dim(gene.filtered)  [1] 15408 44  > write.table(gene.filtered,"gene\_filtered.txt",sep="\t",quote=F,row.names=T) |

After save the normalized and filtered data, we are ready to conduct our analysis.

We run filtered data through MiDAS and load into R for multiplicity correction.

|  |
| --- |
| apt-midas --cel-files .\cel\_files.txt -g .\gene\_filtered.txt -e .\exon\_filtered.txt -m .\HuEx-1\_0-st-v2.r2.dt1.hg18.core.mps -nol -n -o .\output\midas  > midas = read.table("./output/midas/midas.pvalues.txt",skip=16,sep="\t",header=T)  > midas = midas[order(midas$pvalue),]  > head(midas)  probeset\_list\_id probeset\_id pvalue  6051 2362900 2362892 0.000307  56212 2855966 2855963 0.000340  166037 3901319 3901296 0.000371  172465 3960315 3960302 0.000380  111277 3433980 3433929 0.000433  36453 2644440 2644418 0.000508  > summary(p.adjust(midas$pvalue,method="BH"))  Min. 1st Qu. Median Mean 3rd Qu. Max.  0.4989 0.5434 0.6514 0.6768 0.7883 1.0000  > summary(p.adjust(midas$pvalue,method="fdr"))  Min. 1st Qu. Median Mean 3rd Qu. Max.  0.4989 0.5434 0.6514 0.6768 0.7883 1.0000  > midas = cbind(midas, p.adjust(midas$pvalue, method="BH"))  > head(midas)  probeset\_list\_id probeset\_id pvalue p.adjust(midas$pvalue)  6051 2362900 2362892 0.000307 1  56212 2855966 2855963 0.000340 1  166037 3901319 3901296 0.000371 1  172465 3960315 3960302 0.000380 1  111277 3433980 3433929 0.000433 1  36453 2644440 2644418 0.000508 1  p.adjust(midas$pvalue, method = "BH")  6051 0.4988673  56212 0.4988673  166037 0.4988673  172465 0.4988673  111277 0.4988673  36453 0.4988673 |

It seems that the effects sizes aren’t large enough for classic multiplicity correction techniques like Benjamini-Hochberg and FDR. The lowest p-values are 0.4989 for each of the methods.

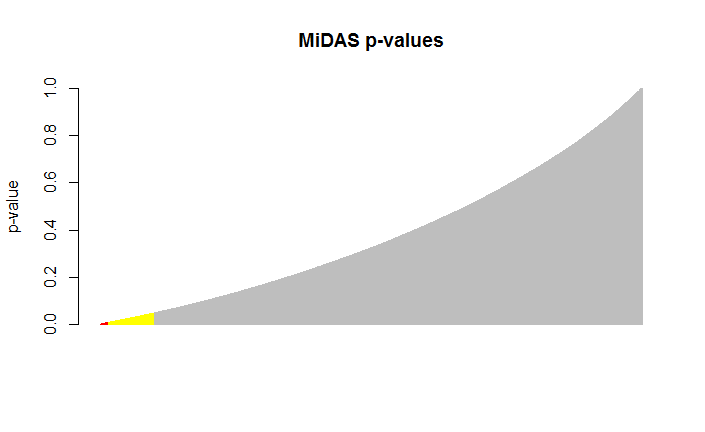
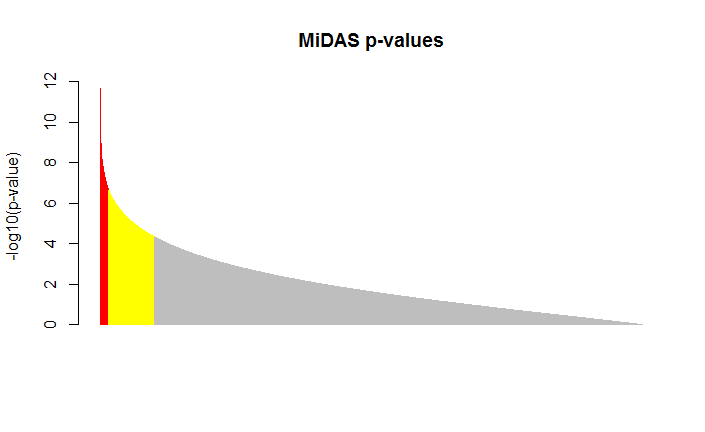
To make sure that the results after multiplicity correction are correct, we can run MiDAS on the unfiltered probesets. Then we load the data into R:

|  |
| --- |
| apt-midas --cel-files .\cel\_files2.txt -g .\output\gene\rma-sketch.summary.txt -e .\output\exon\rma-sketch.summary.txt -m .\HuEx-1\_0-st-v2.r2.dt1.hg18.core.mps -nol -n -f -o .\output\midas2  > midas = read.table("./output/midas2/midas.pvalues.txt",skip=16,sep="\t",header=T)  > midas = midas[order(midas$pvalue),]  > head(midas)  probeset\_list\_id probeset\_id pvalue  123779 3320171 3320169 0.000234  7666 2362900 2362892 0.000307  69957 2855966 2855963 0.000340  73059 2886396 2886174 0.000340  214676 3901319 3901296 0.000371  223559 3960315 3960302 0.000380  > summary(p.adjust(midas$pvalue,method="BH"))  Min. 1st Qu. Median Mean 3rd Qu. Max.  0.5177 0.5653 0.6738 0.7008 0.8158 1.0000  > summary(p.adjust(midas$pvalue,method="fdr"))  Min. 1st Qu. Median Mean 3rd Qu. Max.  0.5177 0.5653 0.6738 0.7008 0.8158 1.0000 |

The rest of the analysis is continued using the filtered dataset.

Let’s start the analysis by plotting the distribution of p-values from MiDAS.

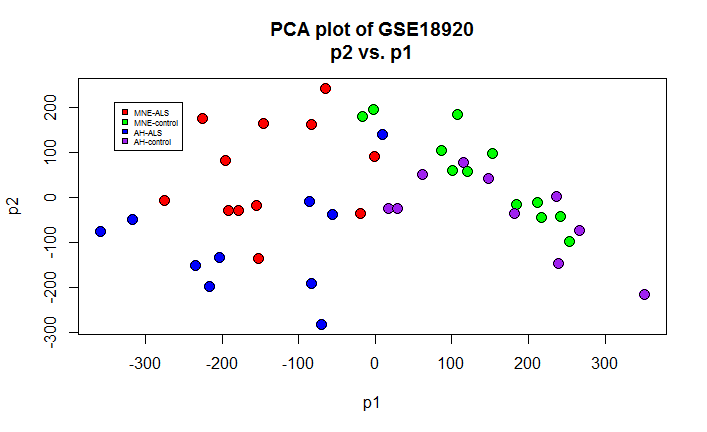
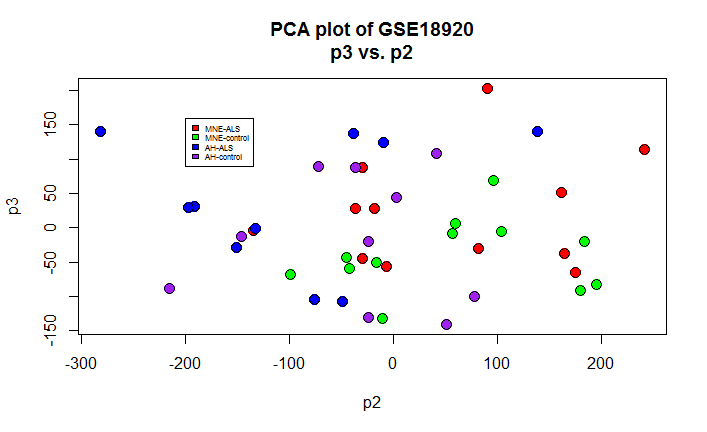
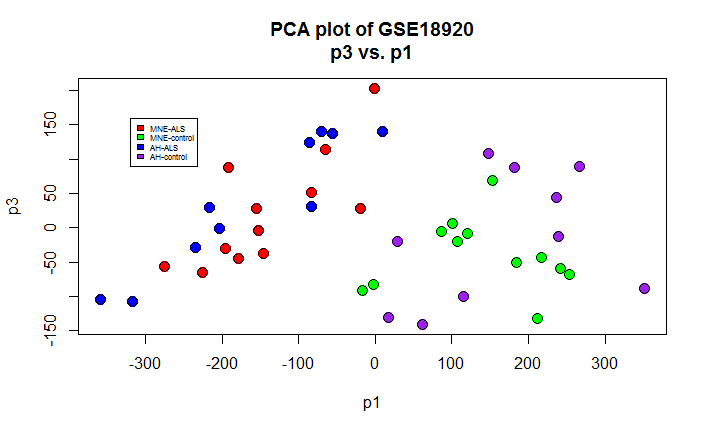
|  |
| --- |
| > pcolors = rep("gray", length(midas$pvalue))  > for (i in 1:length(midas$pvalue)) {if (midas$pvalue[i]<0.01) {pcolors[i]="red"} else if (midas$pvalue[i]<0.05) {pcolors[i]="yellow"}}  > plot(-log2(midas$pvalue),type='h', col=pcolors,axes=F,xlab="",ylab="-log10(p-value)",main="MiDAS p-values")  > axis(2)  > pcolors = rep("gray", length(midas$pvalue))  > for (i in 1:length(midas$pvalue)) {if (midas$pvalue[i]<0.01) {pcolors[i]="red"} else if (midas$pvalue[i]<0.05) {pcolors[i]="yellow"}}  > plot(midas$pvalue,type='h', col=pcolors,axes=F,xlab="",ylab="p-value",main="MiDAS p-values")  > axis(2)  > sum(midas$pvalue<0.05)  [1] 17963  > sum(midas$pvalue<0.01)  [1] 2723  > sum(midas$pvalue<0.001)  [1] 56 |

There are few 17963 p-values significant at the 0.05 level (yellow), 2723 at 0.01 (red), and 56 at 0.001.

We use PCA to visualize the difference between conditions:

|  |
| --- |
| > d.exon = exon.fil2  > dat.pca = prcomp(t(d.exon))  > dat.loadings = dat.pca$x[,1:3]  > plot(range(dat.loadings[,1]),range(dat.loadings[,2]),type="n",xlab='p1',ylab='p2',main='PCA plot of GSE18920\np2 vs. p1')  > points(dat.loadings[,1][cel\_files$group\_id == "MNE-ALS "], dat.loadings[,2][cel\_files$group\_id == "MNE-ALS "],col=1,bg='red',pch=21,cex=1.5)  > points(dat.loadings[,1][cel\_files$group\_id == "MNE-control "], dat.loadings[,2][cel\_files$group\_id == "MNE-control "],col=1,bg='blue',pch=21,cex=1.5)  > points(dat.loadings[,1][cel\_files$group\_id == "AH-ALS "], dat.loadings[,2][cel\_files$group\_id == "AH-ALS "],col=1,bg='green',pch=21,cex=1.5)  > points(dat.loadings[,1][cel\_files$group\_id == "AH-control "], dat.loadings[,2][cel\_files$group\_id == "AH-control "],col=1,bg='purple',pch=21,cex=1.5)  > legend(-340,210,c("MNE-ALS","MNE-control","AH-ALS","AH-control"),fill=c("red","green","blue","purple"),cex=0.5)  > plot(range(dat.loadings[,1]),range(dat.loadings[,3]),type="n",xlab='p1',ylab='p3',main='PCA plot of GSE18920\np3 vs. p1')  > points(dat.loadings[,1][cel\_files$group\_id == "MNE-ALS "], dat.loadings[,3][cel\_files$group\_id == "MNE-ALS "],col=1,bg='red',pch=21,cex=1.5)  > points(dat.loadings[,1][cel\_files$group\_id == "MNE-control "], dat.loadings[,3][cel\_files$group\_id == "MNE-control "],col=1,bg='blue',pch=21,cex=1.5)  > points(dat.loadings[,1][cel\_files$group\_id == "AH-ALS "], dat.loadings[,3][cel\_files$group\_id == "AH-ALS "],col=1,bg='green',pch=21,cex=1.5)  > points(dat.loadings[,1][cel\_files$group\_id == "AH-control "], dat.loadings[,3][cel\_files$group\_id == "AH-control "],col=1,bg='purple',pch=21,cex=1.5)  > legend(-320,160,c("MNE-ALS","MNE-control","AH-ALS","AH-control"),fill=c("red","green","blue","purple"),cex=0.5)  > plot(range(dat.loadings[,2]),range(dat.loadings[,3]),type="n",xlab='p2',ylab='p3',main='PCA plot of GSE18920\np3 vs. p2')  > points(dat.loadings[,2][cel\_files$group\_id == "MNE-ALS "], dat.loadings[,3][cel\_files$group\_id == "MNE-ALS "],col=1,bg='red',pch=21,cex=1.5)  > points(dat.loadings[,2][cel\_files$group\_id == "MNE-control "], dat.loadings[,3][cel\_files$group\_id == "MNE-control "],col=1,bg='blue',pch=21,cex=1.5)  > points(dat.loadings[,2][cel\_files$group\_id == "AH-ALS "], dat.loadings[,3][cel\_files$group\_id == "AH-ALS "],col=1,bg='green',pch=21,cex=1.5)  > points(dat.loadings[,2][cel\_files$group\_id == "AH-control "], dat.loadings[,3][cel\_files$group\_id == "AH-control "],col=1,bg='purple',pch=21,cex=1.5)  > legend(-200,160,c("MNE-ALS","MNE-control","AH-ALS","AH-control"),fill=c("red","green","blue","purple"),cex=0.5) |

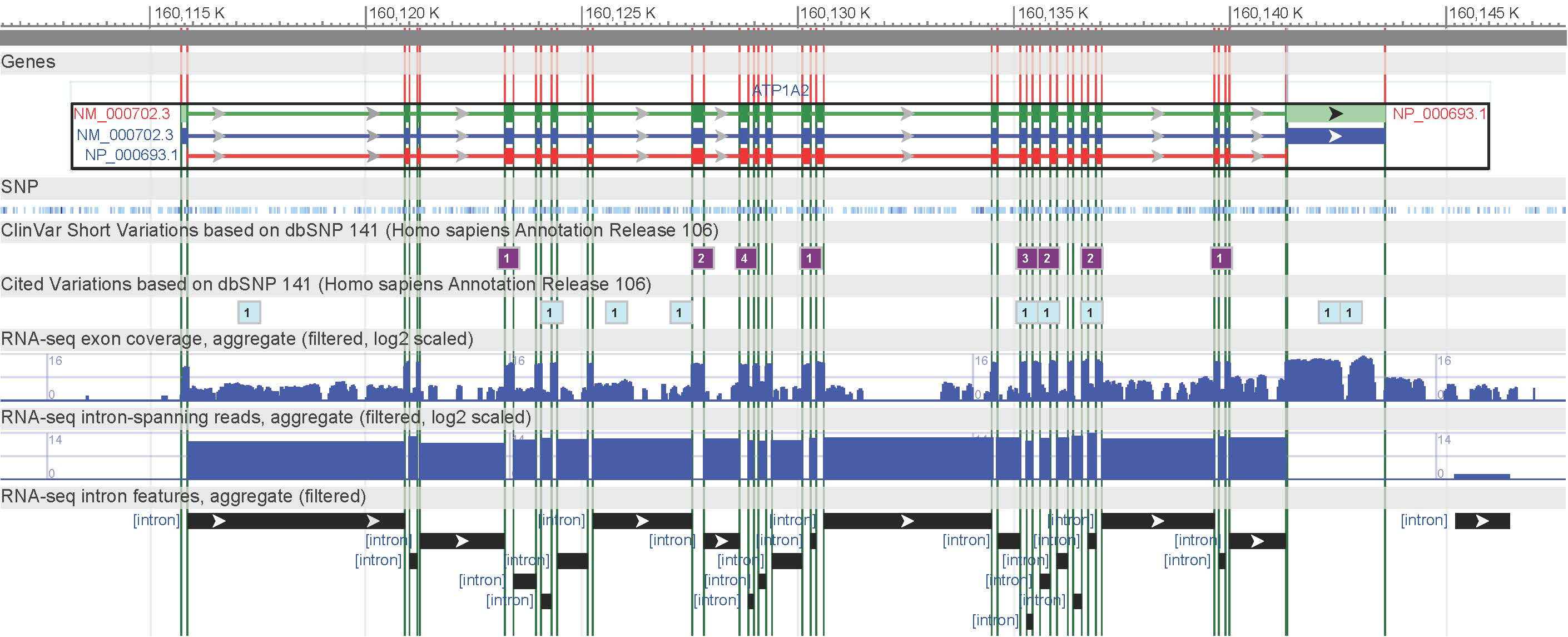
The first two plots (p2 vs. p1 and p3 vs. p1) separate the ALS and control classes well. There is also decent separation between the subtypes of ALS and control.

Let’s look at top significant probes.

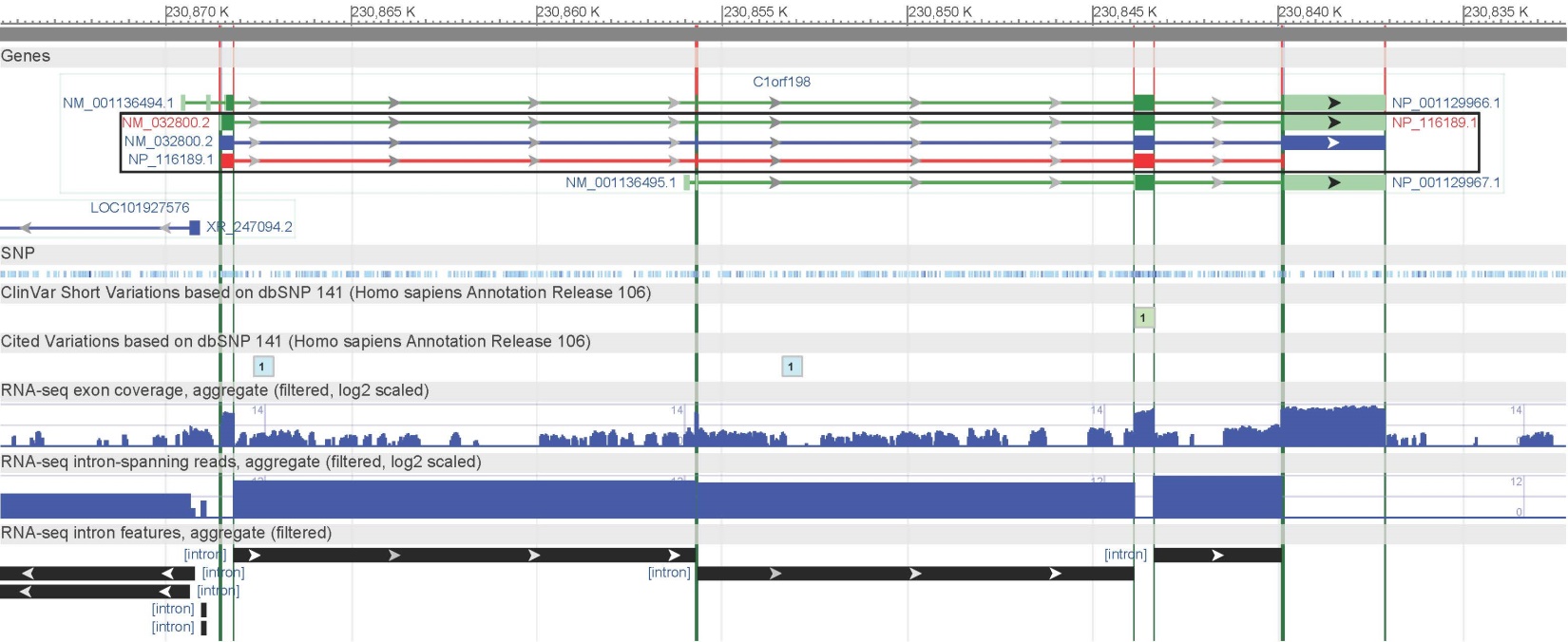
|  |
| --- |
| > sig = head(midas, 20)  > sig.ann = ann.core[which(ann.core$probeset\_id%in%pl),]  > write.table(sig.ann,"sig.txt",sep="\t",quote=F,row.names=F)  probeset\_id seqname strand start stop probe\_count transcript\_cluster\_id exon\_id psr\_id gene\_assignment mrna\_assignment crosshyb\_type number\_independent\_probes number\_cross\_hyb\_probes number\_nonoverlapping\_probes level bounded noBoundedEvidence has\_cds fl mrna est vegaGene vegaPseudoGene ensGene sgpGene exoniphy twinscan geneid genscan genscanSubopt mouse\_fl mouse\_mrna rat\_fl rat\_mrna microRNAregistry rnaGene mitomap probeset\_type  2362900 chr1 + 160090697 160090724 4 2362892 29085 39175 NM\_000702 // ATP1A2 /// ENST00000361216 // ATP1A2 /// BC052271 // ATP1A2 /// ENST00000472488 // ATP1A2 /// ENST00000392233 // ATP1A2 NM\_000702 // chr1 // 100 // 4 // 4 // 0 /// ENST00000361216 // chr1 // 100 // 4 // 4 // 0 /// BC052271 // chr1 // 100 // 4 // 4 // 0 /// ENST00000472488 // chr1 // 100 // 4 // 4 // 0 /// ENST00000392233 // chr1 // 100 // 4 // 4 // 0 1 1 0 1 core 0 0 1 3 4 6 0 0 1 1 1 1 1 1 0 2 2 2 0 0 0 0 main  2460352 chr1 - 231003962 231003997 4 2460325 88858 118536 ENST00000366663 // C1orf198 /// ENST00000470540 // C1orf198 /// NM\_032800 // C1orf198 /// NM\_001136494 // C1orf198 /// BC066649 // C1orf198 ENST00000366663 // chr1 // 100 // 4 // 4 // 0 /// ENST00000470540 // chr1 // 100 // 4 // 4 // 0 /// NM\_032800 // chr1 // 100 // 4 // 4 // 0 /// NM\_001136494 // chr1 // 100 // 4 // 4 // 0 /// BC066649 // chr1 // 100 // 4 // 4 // 0 1 1 0 1 core 0 0 1 2 3 18 0 0 1 1 0 0 0 0 0 2 2 0 0 0 0 0 main  2644440 chr3 + 137749907 137749937 4 2644418 204891 267478 NM\_016369 // CLDN18 /// NM\_001002026 // CLDN18 /// ENST00000343735 // CLDN18 /// ENST00000183605 // CLDN18 /// BC146668 // CLDN18 /// ENST00000536138 // CLDN18 /// ENST00000479660 // CLDN18 NM\_016369 // chr3 // 100 // 4 // 4 // 0 /// NM\_001002026 // chr3 // 100 // 4 // 4 // 0 /// ENST00000343735 // chr3 // 100 // 4 // 4 // 0 /// ENST00000183605 // chr3 // 100 // 4 // 4 // 0 /// BC146668 // chr3 // 100 // 4 // 4 // 0 /// ENST00000536138 // chr3 // 100 // 4 // 4 // 0 /// ENST00000479660 // chr3 // 100 // 4 // 4 // 0 1 1 0 1 core 0 0 1 6 2 1 0 0 2 1 0 1 1 1 1 5 1 0 0 0 0 0 main  2710618 chr3 - 190039774 190039971 4 2710599 246445 320594 NM\_021101 // CLDN1 /// ENST00000295522 // CLDN1 /// AF114837 // CLDN1 /// ENST00000545382 // CLDN1 NM\_021101 // chr3 // 100 // 4 // 4 // 0 /// ENST00000295522 // chr3 // 100 // 4 // 4 // 0 /// AF114837 // chr3 // 100 // 4 // 4 // 0 /// ENST00000545382 // chr3 // 100 // 3 // 3 // 0 1 4 0 3 core 0 0 1 8 0 9 0 0 1 1 1 1 1 1 1 3 7 3 0 0 0 0 main  2855966 chr5 - 45262452 45262477 2 2855963 337651 436360 NM\_021072 // HCN1 /// ENST00000303230 // HCN1 /// AF488549 // HCN1 NM\_021072 // chr5 // 100 // 2 // 2 // 0 /// ENST00000303230 // chr5 // 100 // 2 // 2 // 0 /// AF488549 // chr5 // 100 // 2 // 2 // 0 1 1 0 1 core 0 0 1 2 1 0 0 0 1 0 0 0 0 0 0 2 2 2 0 0 0 0 main  2933421 chr6 + 158454487 158454523 4 2933392 385499 498200 --- --- 1 2 0 1 core 0 0 1 4 3 3 1 0 1 1 1 1 1 1 1 4 4 4 0 0 0 0 main  3071701 chr7 - 128032362 128032405 4 3071700 471378 610117 NM\_000883 // IMPDH1 /// NM\_183243 // IMPDH1 /// NM\_001102605 // IMPDH1 /// NM\_001142573 // IMPDH1 /// NM\_001142574 // IMPDH1 /// NM\_001142575 // IMPDH1 /// ENST00000338791 // IMPDH1 /// ENST00000354269 // IMPDH1 /// ENST00000343214 // IMPDH1 /// ENST00000348127 // IMPDH1 /// ENST00000419067 // IMPDH1 /// ENST00000496200 // IMPDH1 /// BC033622 // IMPDH1 /// ENST00000469328 // IMPDH1 /// ENST00000484496 // IMPDH1 /// ENST00000378717 // IMPDH1 NM\_000883 // chr7 // 100 // 4 // 4 // 0 /// NM\_183243 // chr7 // 100 // 4 // 4 // 0 /// NM\_001102605 // chr7 // 100 // 4 // 4 // 0 /// NM\_001142573 // chr7 // 100 // 4 // 4 // 0 /// NM\_001142574 // chr7 // 100 // 4 // 4 // 0 /// NM\_001142575 // chr7 // 100 // 4 // 4 // 0 /// ENST00000338791 // chr7 // 100 // 4 // 4 // 0 /// ENST00000354269 // chr7 // 100 // 4 // 4 // 0 /// ENST00000343214 // chr7 // 100 // 4 // 4 // 0 /// ENST00000348127 // chr7 // 100 // 4 // 4 // 0 /// ENST00000419067 // chr7 // 100 // 4 // 4 // 0 /// ENST00000496200 // chr7 // 100 // 4 // 4 // 0 /// BC033622 // chr7 // 100 // 4 // 4 // 0 /// ENST00000469328 // chr7 // 100 // 4 // 4 // 0 /// ENST00000484496 // chr7 // 100 // 4 // 4 // 0 /// ENST00000378717 // chr7 // 100 // 4 // 4 // 0 1 2 0 1 core 0 0 0 4 4 45 1 0 2 0 0 0 0 0 0 2 2 0 0 0 0 0 main  3359463 chr11 - 2949729 2949895 4 3359461 650378 843108 NM\_003311 // PHLDA2 /// ENST00000314222 // PHLDA2 /// AF019953 // PHLDA2 /// AF001294 // PHLDA2 /// AF035444 // PHLDA2 /// BC005034 // PHLDA2 /// AK223027 // PHLDA2 NM\_003311 // chr11 // 100 // 4 // 4 // 0 /// ENST00000314222 // chr11 // 100 // 4 // 4 // 0 /// AF019953 // chr11 // 100 // 4 // 4 // 0 /// AF001294 // chr11 // 100 // 4 // 4 // 0 /// AF035444 // chr11 // 100 // 4 // 4 // 0 /// BC005034 // chr11 // 100 // 4 // 4 // 0 /// AK223027 // chr11 // 100 // 4 // 4 // 0 1 4 0 4 core 0 0 0 5 0 45 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 main  3394303 chr11 - 119187768 119187799 4 3394264 671443 871515 NM\_006500 // MCAM /// ENST00000264036 // MCAM /// BC056418 // MCAM /// AK291571 // MCAM /// AF089868 // MCAM /// M29277 // MCAM /// M28882 // MCAM /// AK128335 // MCAM /// ENST00000528533 // MCAM /// ENST00000527913 // MCAM /// ENST00000528502 // MCAM NM\_006500 // chr11 // 100 // 4 // 4 // 0 /// ENST00000264036 // chr11 // 100 // 4 // 4 // 0 /// BC056418 // chr11 // 100 // 4 // 4 // 0 /// AK291571 // chr11 // 100 // 4 // 4 // 0 /// AF089868 // chr11 // 100 // 4 // 4 // 0 /// M29277 // chr11 // 100 // 4 // 4 // 0 /// M28882 // chr11 // 100 // 4 // 4 // 0 /// AK128335 // chr11 // 100 // 4 // 4 // 0 /// ENST00000528533 // chr11 // 100 // 4 // 4 // 0 /// ENST00000527913 // chr11 // 100 // 4 // 4 // 0 /// ENST00000528502 // chr11 // 100 // 4 // 4 // 0 1 1 0 1 core 0 0 1 5 1 30 0 0 2 0 0 1 1 1 0 4 0 0 0 0 0 0 main  3396254 chr11 - 124789582 124789607 2 3396249 672559 873091 NM\_001037558 // HEPN1 /// NM\_152722 // HEPACAM /// ENST00000298251 // HEPN1 /// ENST00000298251 // HEPACAM /// ENST00000408930 // HEPN1 /// AK122595 // HEPACAM NM\_001037558 // chr11 // 100 // 2 // 2 // 0 /// NM\_152722 // chr11 // 100 // 2 // 2 // 0 /// ENST00000298251 // chr11 // 100 // 2 // 2 // 0 /// ENST00000408930 // chr11 // 100 // 2 // 2 // 0 /// AK122595 // chr11 // 100 // 2 // 2 // 0 1 1 0 1 core 0 0 0 2 3 8 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 main  3433980 chr12 + 119583188 119583221 4 3433929 696009 903687 NM\_194286 // SRRM4 /// ENST00000267260 // SRRM4 /// BC152471 // SRRM4 NM\_194286 // chr12 // 100 // 4 // 4 // 0 /// ENST00000267260 // chr12 // 100 // 4 // 4 // 0 /// BC152471 // chr12 // 100 // 4 // 4 // 0 1 1 0 1 core 0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 main  3458724 chr12 - 58025701 58025893 4 3458700 710903 924174 NM\_001478 // B4GALNT1 /// ENST00000341156 // B4GALNT1 /// BC029828 // B4GALNT1 /// M83651 // B4GALNT1 /// AK293432 // B4GALNT1 /// AK299845 // B4GALNT1 /// AK289690 // B4GALNT1 /// AK302503 // B4GALNT1 /// AB209460 // B4GALNT1 /// ENST00000553142 // B4GALNT1 /// ENST00000552798 // B4GALNT1 /// ENST00000449184 // B4GALNT1 /// ENST00000548888 // B4GALNT1 /// ENST00000551925 // B4GALNT1 /// ENST00000418555 // B4GALNT1 /// ENST00000550764 // B4GALNT1 /// ENST00000552350 // B4GALNT1 NM\_001478 // chr12 // 100 // 4 // 4 // 0 /// ENST00000341156 // chr12 // 100 // 4 // 4 // 0 /// BC029828 // chr12 // 100 // 4 // 4 // 0 /// M83651 // chr12 // 100 // 4 // 4 // 0 /// AK293432 // chr12 // 100 // 4 // 4 // 0 /// AK299845 // chr12 // 100 // 4 // 4 // 0 /// AK289690 // chr12 // 100 // 4 // 4 // 0 /// AK302503 // chr12 // 75 // 3 // 4 // 0 /// AB209460 // chr12 // 100 // 4 // 4 // 0 /// ENST00000553142 // chr12 // 100 // 4 // 4 // 0 /// ENST00000552798 // chr12 // 100 // 4 // 4 // 0 /// ENST00000449184 // chr12 // 100 // 4 // 4 // 0 /// ENST00000548888 // chr12 // 100 // 4 // 4 // 0 /// ENST00000551925 // chr12 // 100 // 4 // 4 // 0 /// ENST00000418555 // chr12 // 100 // 4 // 4 // 0 /// ENST00000550764 // chr12 // 100 // 4 // 4 // 0 /// ENST00000552350 // chr12 // 100 // 4 // 4 // 0 /// GENSCAN00000002316 // chr12 // 100 // 4 // 4 // 0 1 4 0 3 core 0 0 1 3 0 8 0 0 2 1 1 1 1 1 0 6 3 2 0 0 0 0 main  3643344 chr16 + 767074 767341 4 3643333 825367 1074071 NM\_024042 // METRN /// ENST00000568223 // METRN /// BC000662 // METRN /// ENST00000564661 // METRN /// ENST00000570132 // METRN NM\_024042 // chr16 // 100 // 4 // 4 // 0 /// ENST00000568223 // chr16 // 100 // 4 // 4 // 0 /// BC000662 // chr16 // 100 // 4 // 4 // 0 /// ENST00000564661 // chr16 // 100 // 4 // 4 // 0 /// ENST00000570132 // chr16 // 100 // 2 // 2 // 0 1 4 0 4 core 0 0 1 2 0 48 0 0 1 1 1 1 1 1 0 1 3 0 0 0 0 0 main  3748036 chr17 - 17754219 17754262 4 3748026 888255 1160299 NM\_001033551 // TOM1L2 /// NM\_001082968 // TOM1L2 /// ENST00000379504 // TOM1L2 /// ENST00000318094 // TOM1L2 /// AF467441 // TOM1L2 /// ENST00000478943 // TOM1L2 /// ENST00000395739 // TOM1L2 /// ENST00000535933 // TOM1L2 /// ENST00000486413 // TOM1L2 NM\_001033551 // chr17 // 100 // 4 // 4 // 0 /// NM\_001082968 // chr17 // 100 // 4 // 4 // 0 /// ENST00000379504 // chr17 // 100 // 4 // 4 // 0 /// ENST00000318094 // chr17 // 100 // 4 // 4 // 0 /// AF467441 // chr17 // 100 // 4 // 4 // 0 /// ENST00000478943 // chr17 // 100 // 4 // 4 // 0 /// ENST00000395739 // chr17 // 100 // 4 // 4 // 0 /// ENST00000535933 // chr17 // 100 // 4 // 4 // 0 /// ENST00000486413 // chr17 // 100 // 4 // 4 // 0 1 2 0 1 core 0 0 1 3 2 6 0 0 3 1 0 1 1 1 1 2 1 0 0 0 0 0 main  3901319 chr20 - 23618392 23618420 4 3901296 980076 1286931 ENST00000398411 // CST3 /// ENST00000376925 // CST3 /// ENST00000398409 // CST3 /// NM\_000099 // CST3 /// BT006839 // CST3 ENST00000398411 // chr20 // 100 // 4 // 4 // 0 /// ENST00000376925 // chr20 // 100 // 4 // 4 // 0 /// ENST00000398409 // chr20 // 100 // 4 // 4 // 0 /// NM\_000099 // chr20 // 25 // 1 // 4 // 0 /// BT006839 // chr20 // 25 // 1 // 4 // 0 1 1 0 1 core 0 0 1 5 1 295 2 0 1 1 0 1 1 1 0 6 3 0 1 0 0 0 main  3910396 chr20 - 52675220 52675250 4 3910360 985633 1294343 NM\_003657 // BCAS1 /// ENST00000395961 // BCAS1 /// BC126346 // BCAS1 /// ENST00000371440 // BCAS1 NM\_003657 // chr20 // 100 // 4 // 4 // 0 /// ENST00000395961 // chr20 // 100 // 4 // 4 // 0 /// BC126346 // chr20 // 100 // 4 // 4 // 0 /// ENST00000371440 // chr20 // 100 // 4 // 4 // 0 1 1 0 1 core 0 0 1 2 1 3 0 0 1 1 0 0 1 1 1 2 1 0 0 0 0 0 main  3913893 chr20 - 62119374 62119498 4 3913892 987729 1297112 NM\_001958 // EEF1A2 /// ENST00000298049 // EEF1A2 /// ENST00000217182 // EEF1A2 /// BC110409 // EEF1A2 NM\_001958 // chr20 // 100 // 4 // 4 // 0 /// ENST00000298049 // chr20 // 100 // 4 // 4 // 0 /// ENST00000217182 // chr20 // 100 // 4 // 4 // 0 /// BC110409 // chr20 // 100 // 4 // 4 // 0 1 4 0 2 core 0 0 0 2 3 54 2 0 1 0 0 0 0 0 0 3 0 0 0 0 0 0 main  3939890 chr22 + 24581988 24582126 4 3939875 1003518 1318074 NM\_019601 // SUSD2 /// ENST00000358321 // SUSD2 /// AK126105 // SUSD2 /// ENST00000463101 // SUSD2 NM\_019601 // chr22 // 100 // 4 // 4 // 0 /// ENST00000358321 // chr22 // 100 // 4 // 4 // 0 /// AK126105 // chr22 // 100 // 4 // 4 // 0 /// ENST00000463101 // chr22 // 100 // 4 // 4 // 0 1 4 0 3 core 0 0 1 2 2 17 1 0 3 1 1 1 1 1 0 2 8 0 0 0 0 0 main  3939899 chr22 + 24583540 24583721 4 3939875 1003519 1318083 NM\_019601 // SUSD2 /// ENST00000358321 // SUSD2 /// AK126105 // SUSD2 /// ENST00000463101 // SUSD2 NM\_019601 // chr22 // 100 // 4 // 4 // 0 /// ENST00000358321 // chr22 // 100 // 4 // 4 // 0 /// AK126105 // chr22 // 100 // 4 // 4 // 0 /// ENST00000463101 // chr22 // 100 // 4 // 4 // 0 1 4 0 4 core 0 0 1 2 2 7 1 0 3 1 1 1 1 1 0 2 8 0 0 0 0 0 main  3960315 chr22 - 38379844 38379875 4 3960302 1015777 1334938 NM\_006941 // SOX10 /// ENST00000396884 // SOX10 /// ENST00000360880 // SOX10 /// BC007595 // SOX10 /// ENST00000416937 // SOX10 /// ENST00000427770 // SOX10 NM\_006941 // chr22 // 100 // 4 // 4 // 0 /// ENST00000396884 // chr22 // 100 // 4 // 4 // 0 /// ENST00000360880 // chr22 // 100 // 4 // 4 // 0 /// BC007595 // chr22 // 100 // 4 // 4 // 0 /// ENST00000416937 // chr22 // 100 // 4 // 4 // 0 /// ENST00000427770 // chr22 // 100 // 4 // 4 // 0 1 1 0 1 core 0 0 0 3 2 11 1 0 1 0 1 0 0 0 0 3 5 2 1 0 0 0 main |

First, we check GenBank for the top 5 significant genes.

NM\_000702 // ATP1A2 /// ENST00000361216 // ATP1A2 /// BC052271 // ATP1A2 /// ENST00000472488 // ATP1A2 /// ENST00000392233 // ATP1A2



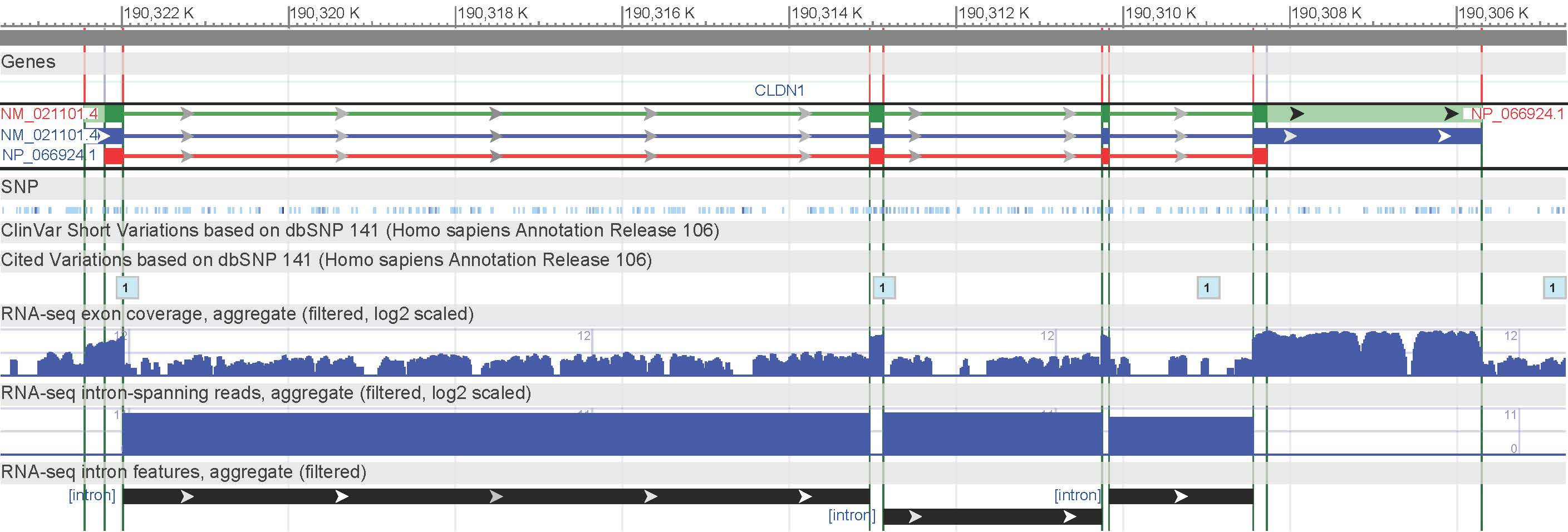
ENST00000366663 // C1orf198 /// ENST00000470540 // C1orf198 /// NM\_032800 // C1orf198 /// NM\_001136494 // C1orf198 /// BC066649 // C1orf198



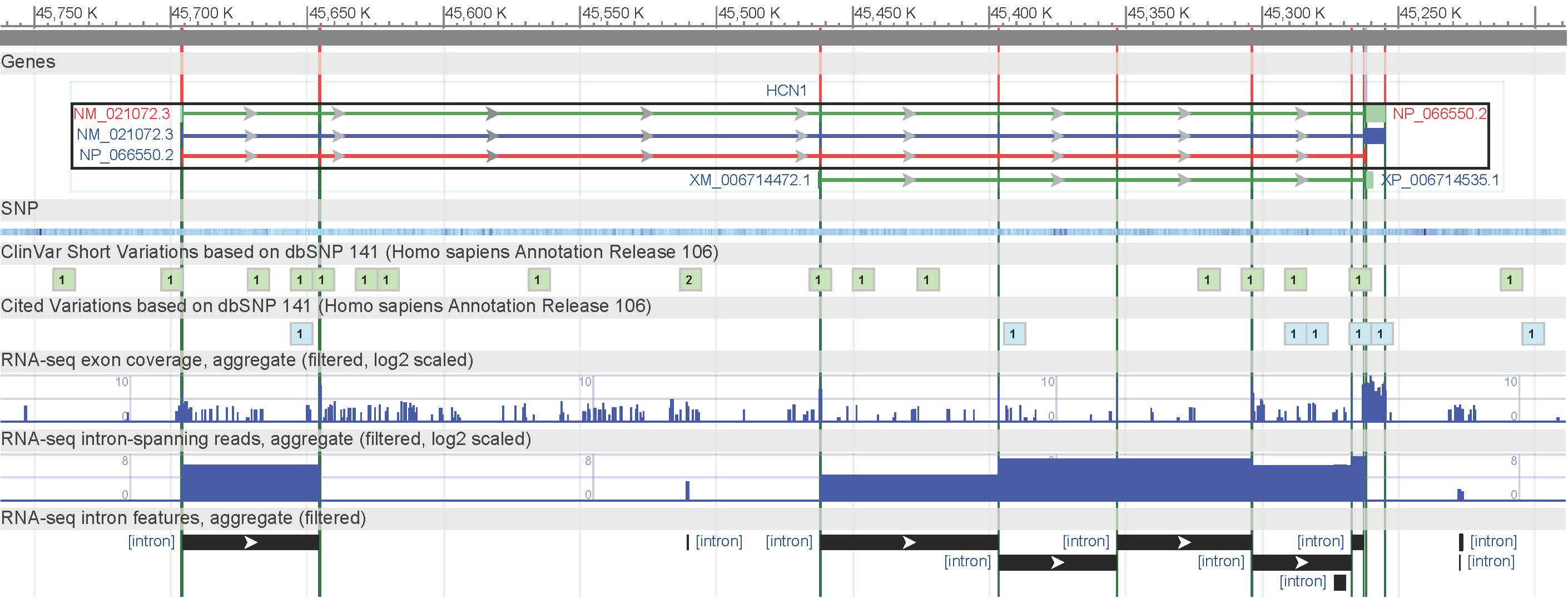
NM\_016369 // CLDN18 /// NM\_001002026 // CLDN18 /// ENST00000343735 // CLDN18 /// ENST00000183605 // CLDN18 /// BC146668 // CLDN18 /// ENST00000536138 // CLDN18 /// ENST00000479660 // CLDN18



NM\_021101 // CLDN1 /// ENST00000295522 // CLDN1 /// AF114837 // CLDN1 /// ENST00000545382 // CLDN1



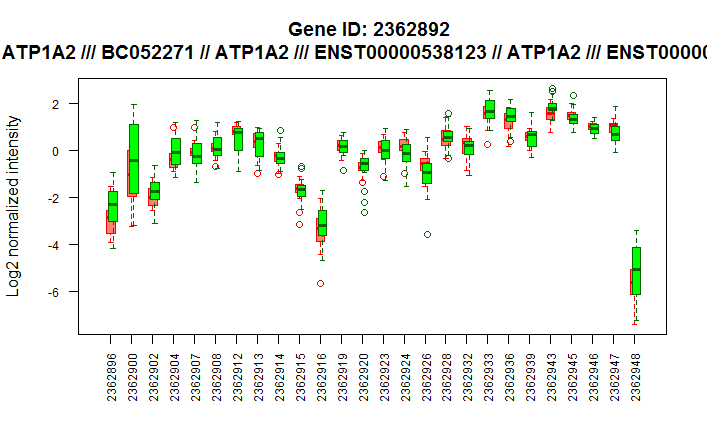
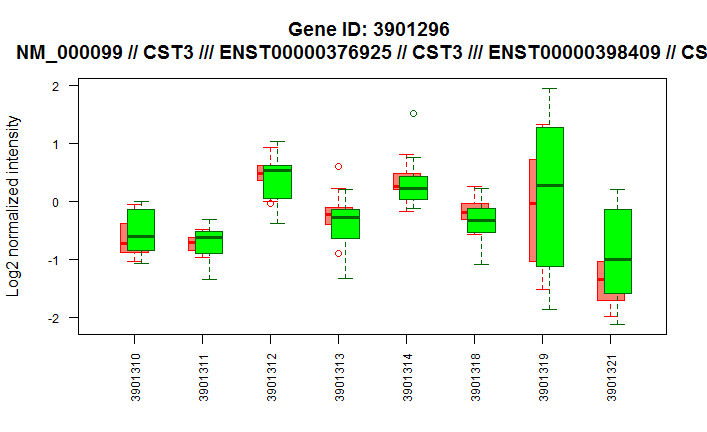
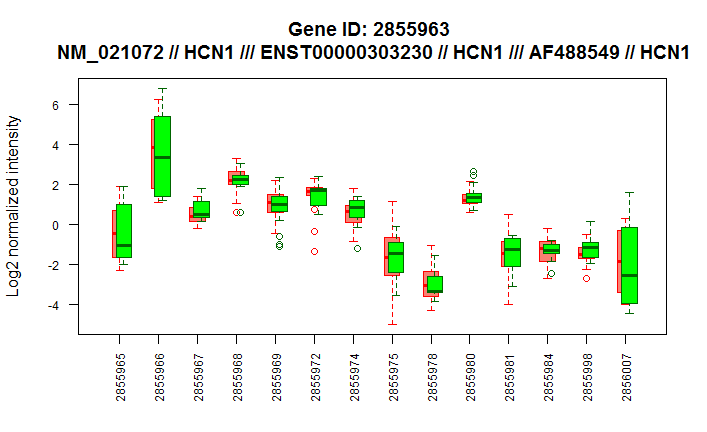
NM\_021072 // HCN1 /// ENST00000303230 // HCN1 /// AF488549 // HCN1



There is evidence of splicing in each of these genes.

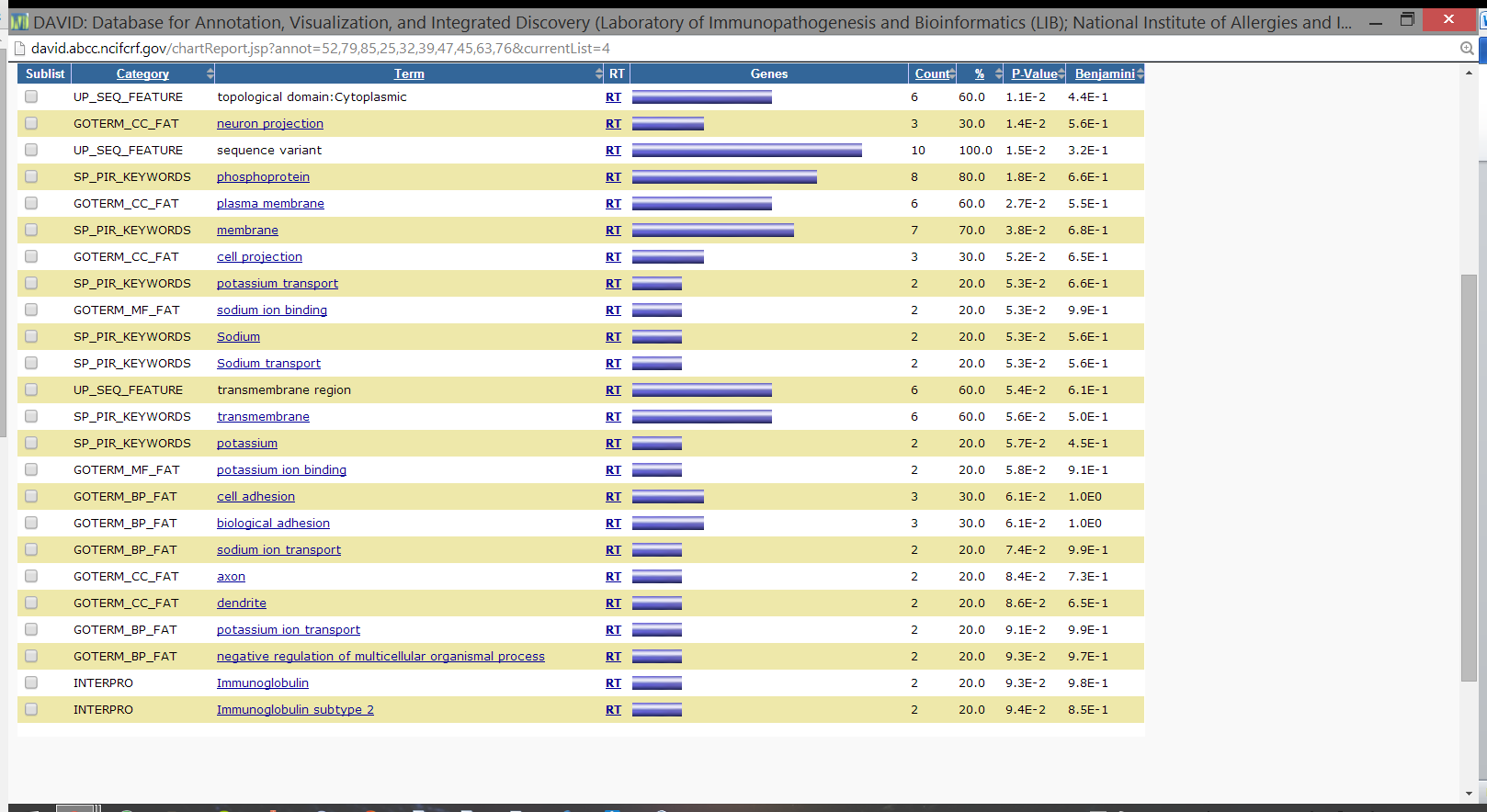
Let’s visualize the splicing in the top 3 genes between conditions.

|  |
| --- |
| > i = as.character(sig[1,2])  > map = ann  > ex <- dimnames(map[map$transcript\_cluster\_id %in% i,])[[1]]  > d.exon <- e[ex,]  > d.gene <- g[i,]  > d.exon = d.exon[-which(row.names(d.exon)>="NA"),]  > plot.exons <- function(exonx,genex,rx,ti) {  + rr <- rx  + rx <- rep(rx,nrow(exonx))  + rx[rx==1] <- "A"  + rx[rx==0] <- "B"  + rx <- as.factor(rx)  + ni <- t(t(exonx)-genex)  + exonx <- as.data.frame(t(ni))  + ex.stack <- stack(exonx)  + d <- data.frame(ex.stack,rx)  + names(d) <- c("exon\_values","exon\_id","class")  +  + d$exon\_id <- as.factor(d$exon\_id)  + d$class <- as.factor(d$class)  + genex.title <- as.character(map[match(ti,as.character(map$transcript\_cluster\_id)),"gene\_assignment"])  + plot(c(.5,(ncol(exonx)+.5)),range(d[,1]),type="n",axes=F,xlab="",ylab="")  + boxplot(exon\_values~exon\_id,add=T,subset=d$class=="A",d,col="salmon",border='red',cex.axis=.75,las=2,ylab='Log2 normalized intensity',main=paste("Gene ID:",ti,"\n",genex.title),boxwex=0.4)  + boxplot(exon\_values~exon\_id,subset=d$class=="B",d,add=T,col="green",border='darkgreen',axes=F,boxwex=0.4, at=c(1:ncol(exonx))+0.1)  + }  > factor <- read.delim("D:/Final Project/factor.txt")  > plot.exons(exonx=d.exon,genex=as.numeric(d.gene),rx=factor$group\_id,ti=i)  > i = as.character(sig[2,2])  > ex <- dimnames(map[map$transcript\_cluster\_id %in% i,])[[1]]  > d.exon <- e[ex,]  > d.gene <- g[i,]  > plot.exons(exonx=d.exon,genex=as.numeric(d.gene),rx=factor$group\_id,ti=i)  > i = as.character(sig[3,2])  > ex <- dimnames(map[map$transcript\_cluster\_id %in% i,])[[1]]  > d.exon <- e[ex,]  > d.gene <- g[i,]  > plot.exons(exonx=d.exon,genex=as.numeric(d.gene),rx=factor$group\_id,ti=i) |

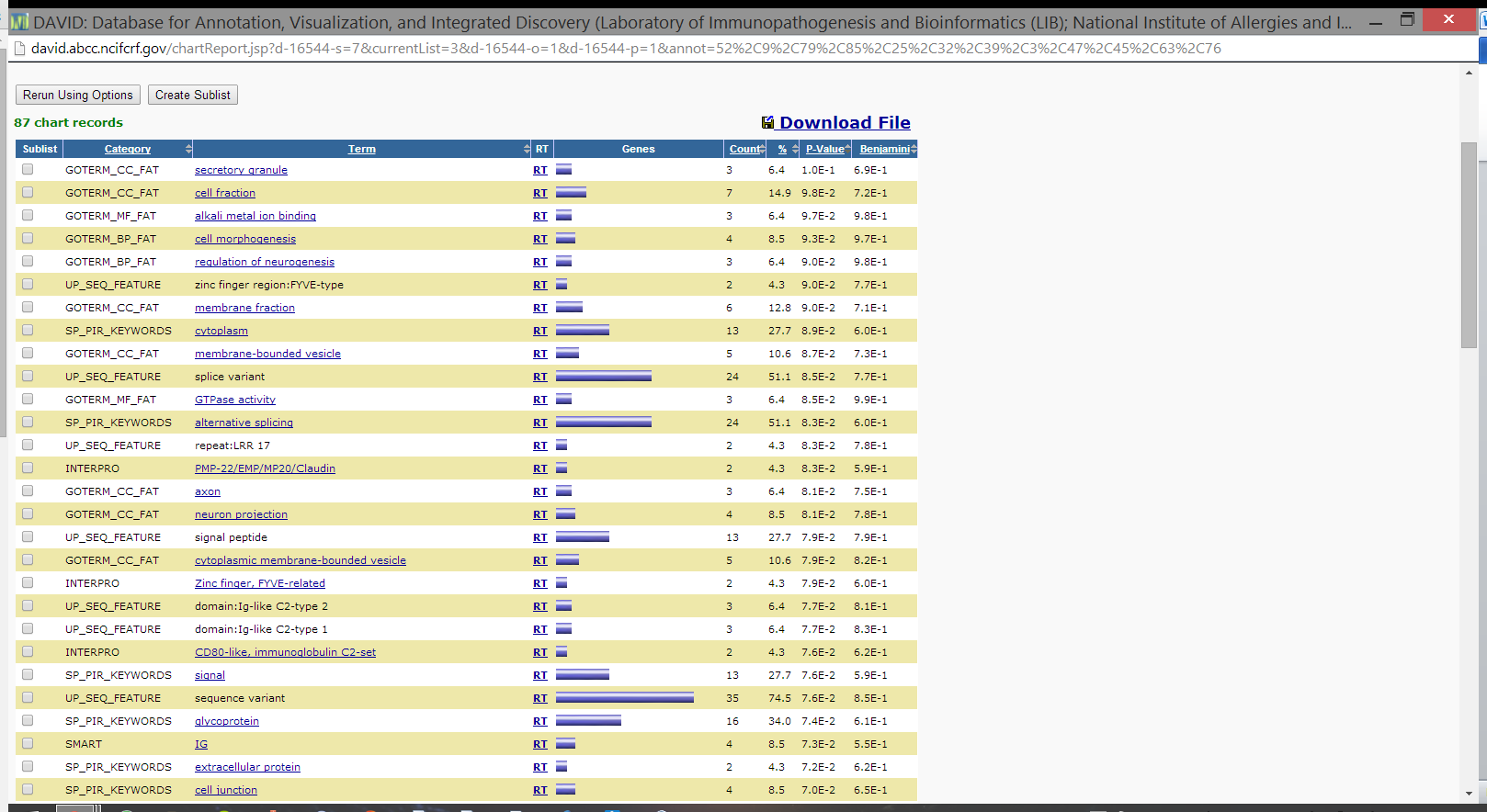
Red is for ALS, green for controls.

Here is the annotation table for top 10 genes from DAVID.



There are hits related to cell adhesion, neuronal membrane, and ion transport. This is in accordance with the findings from the paper.

Feeding the top 57 genes (p<0.001) into DAVID produces many hits tied to neurological activity and function. There are also hits to ALS specifically, and several other neurological diseases. There are many hits related to membrane function and cell adhesion.



**References:**

* Benson, Dennis A., et al. "GenBank." Nucleic acids research 27.1 (1999): 12-17.
* Huang, Da Wei, Brad T. Sherman, and Richard A. Lempicki. "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources." Nature protocols 4.1 (2008): 44-57.
* Lagier-Tourenne, Clotilde, and Don W. Cleveland. "Rethinking als: The fus about tdp-43." Cell 136.6 (2009): 1001-1004.
* Lockstone, Helen E. "Exon array data analysis using Affymetrix power tools and R statistical software." Briefings in bioinformatics 12.6 (2011): 634-644.
* Rabin, Stuart J., et al. "Sporadic ALS has compartment-specific aberrant exon splicing and altered cell–matrix adhesion biology." Human molecular genetics 19.2 (2010): 313-328.
* Rowland, Lewis P., and Neil A. Shneider. "Amyotrophic lateral sclerosis." New England Journal of Medicine 344.22 (2001): 1688-1700.