Anti-seed PNAs targeting multiple oncomiRs for brain tumor therapy

Vijender Singh, CBC, UCHC

‘2023-02-06

```
rm(list=ls())
gc()
```

```
##          used (Mb) gc trigger (Mb) limit (Mb) max used (Mb)
## Ncells 479336 25.6    1034831 55.3         NA   669400 35.8
## Vcells 906052  7.0    8388608 64.0      16384  1851692 14.2
```

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE)


# Load Packages
suppressPackageStartupMessages(library("dplyr",quietly=TRUE))
suppressPackageStartupMessages(library("tidyverse",quietly=TRUE))
suppressPackageStartupMessages(library("genefilter",quietly=TRUE))
suppressPackageStartupMessages(library("grDevices",quietly=TRUE))
suppressPackageStartupMessages(library("ggrepel",quietly=TRUE))
suppressPackageStartupMessages(library("pheatmap",quietly=TRUE))
suppressPackageStartupMessages(library("RColorBrewer",quietly=TRUE))
suppressPackageStartupMessages(library("gplots",quietly=TRUE))
suppressPackageStartupMessages(library("RColorBrewer", quietly=TRUE))
suppressPackageStartupMessages(library("ggplot2", quietly=TRUE))
suppressPackageStartupMessages(library("labeling", quietly=TRUE))
suppressPackageStartupMessages(library("DESeq2",quietly=TRUE))
suppressPackageStartupMessages(library("skimr",quietly=TRUE))
suppressPackageStartupMessages(library("here",quietly=TRUE))
suppressPackageStartupMessages(library("data.table",quietly=TRUE))
suppressPackageStartupMessages(library(fgsea,quietly=TRUE))
suppressPackageStartupMessages(library(org.Hs.eg.db,quietly=TRUE))
suppressPackageStartupMessages(library(msigdbr,quietly=TRUE))
suppressPackageStartupMessages(library(clusterProfiler,quietly=TRUE))
suppressPackageStartupMessages(library(enrichplot,quietly=TRUE))
suppressPackageStartupMessages(library(DOSE,quietly=TRUE))
#Project Directory
projectDir=setwd(here("."))
print(paste0("Project Directory :", projectDir ))
```

```
## [1] "Project Directory :/Users/astral/Documents/presentation/bahal"
```

RNAseq analysis : All Samples

Data Import

Import Counts Matrix and Sample Metadata The data is in tab-separated test (.txt) format. Data is imported using `fred` function from `data.table` package. The sample metadat is vaialable in `sampleTable.csv` file and is read in using `read.csv` function.

```
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

Count matrix Stats: Features(genes) : 35741 Number of samples : 11

Sample Meatadata sample condition Cells Control-1 Control-1 control U87 Control-2 Control-2 control U87 Control-3 Control-3 control U87 PNA-10b-1 PNA-10b-1 PNA.10b U87 PNA-10b-2 PNA-10b-2 PNA.10b U87 PNA-10b-3 PNA-10b-3 PNA.10b U87 PNA-21-1 PNA-21-1 PNA.21 U87 PNA-21-2 PNA-21-2 PNA.21 U87 PNA-21-3 PNA-21-3 PNA.21 U87 PNA-10b+21-1 PNA-10b+21-1 PNA10b.21 U87 PNA-10b+21-2 PNA-10b+21-2 PNA10b.21 U87 PNA-10b+21-3 PNA-10b+21-3 PNA10b.21 U87 treat.info Incubation.time Control-1 mock 72h Control-2 mock 72h Control-3 mock 72h PNA-10b-1 sgPNA-10b/BNP (150 nM sgPNA-10b) 72h PNA-10b-2 sgPNA-10b/BNP (150 nM sgPNA-10b) 72h PNA-10b-3 sgPNA-10b/BNP (150 nM sgPNA-10b) 72h PNA-21-1 sgPNA-21/BNP (150 nM sgPNA-21) 72h PNA-21-2 sgPNA-21/BNP (150 nM sgPNA-21) 72h PNA-21-3 sgPNA-21/BNP (150 nM sgPNA-21) 72h PNA-10b+21-1 sgPNA/BNP (150 nM sgPNA-21, 150 nM sgPNA-10b) 72h PNA-10b+21-2 sgPNA/BNP (150 nM sgPNA-21, 150 nM sgPNA-10b) 72h PNA-10b+21-3 sgPNA/BNP (150 nM sgPNA-21, 150 nM sgPNA-10b) 72h

Reference file gene_name ENSG00000000003 TSPAN6 ENSG00000000419 DPM1 ENSG00000000457 SCYL3 ENSG00000000460 C1orf112 ENSG00000000938 FGR ENSG00000000971 CFH description ENSG00000000003 tetraspanin 6 ENSG00000000419 dolichyl-phosphate mannosyltransferase subunit 1, catalytic ENSG00000000457 SCY1 like pseudokinase 3 ENSG00000000460 FIGNL1 interacting regulator of recombination and mitosis ENSG00000000938 FGR proto-oncogene, Src family tyrosine kinase ENSG00000000971 complement factor H ensembl entrez GENENAME GENENAME_entrezid ENSG00000000003 ENSG00000000003 7105 TSPAN6 TSPAN6 ENSG00000000419 ENSG00000000419 8813 DPM1 DPM1 ENSG00000000457 ENSG00000000457 57147 SCYL3 SCYL3 ENSG00000000460 ENSG00000000460 55732 FIRRM FIRRM ENSG00000000938 ENSG00000000938 2268 FGR FGR ENSG00000000971 ENSG00000000971 3075 CFH CFH

Filtering In this step we will filter out genes with very low counts across the samples. DESeq2 performs independent filtering but reducing the dimension of input dataset can speed up the process in large datasets.

```
genesInAssay<-dim(counts)[1]

# Filtering genes with some parameters.
minimumCountpergene=10
MinSampleWithminimumgeneCounts=4
counts<-counts[rowSums(data.frame(counts>minimumCountpergene))>MinSampleWithminimumgeneCounts,]

cat(c("\n Total genes before filtering : ",genesInAssay),sep=" ",append=TRUE)
cat(c("\n Minimum Counts per gene : ",minimumCountpergene),sep=" ",append = TRUE)
cat(c("\n Minimum Samlpes with Minimum Counts per gene : ",MinSampleWithminimumgeneCounts),sep=" ",appe
cat(c("\n Total genes after filtering : ",dim(counts)[1]),sep=" ",append=TRUE)
```

```
##
##  Total genes before filtering :  35741
##  Minimum Counts per gene :  10
```

```
##  Minimum Samlpes with Minimum Counts per gene :  4
##  Total genes after filtering :  15036
```

Counts QC

In this step we will perform few QC steps to check data completeness and will also try to identify if any sample is behaving odd from the raw counts perspective.

QC1. Sample checks Test that all the input samples in sampleTable have their corresponding counts in the `counts` object.

```
if (sum(colnames(counts) %in% sampleTable$sample) == length(sampleTable$sample)){
  message("Good News !!! Samples in count matrix matches with that of in sampleTable")
}
```

```
## Good News !!! Samples in count matrix matches with that of in sampleTable
```

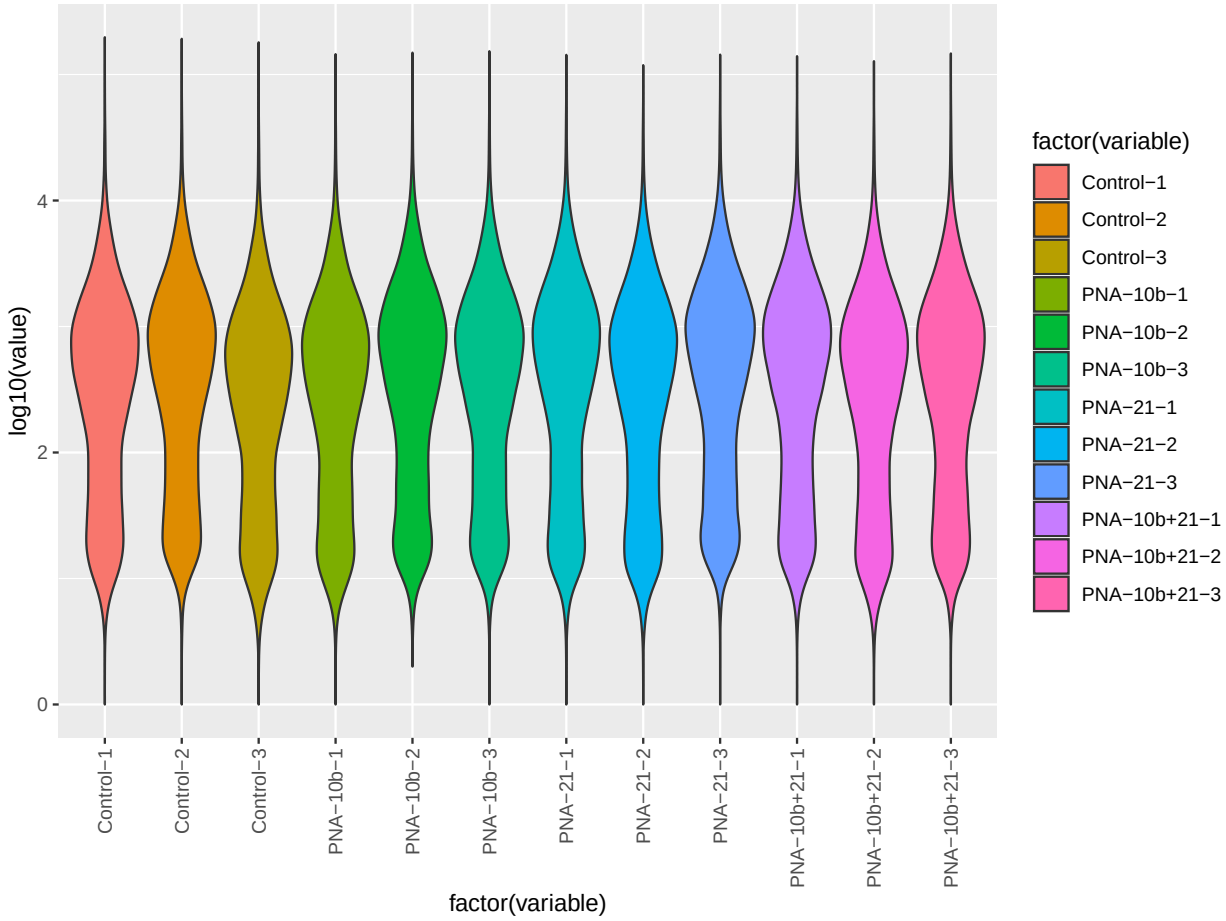
```
cat("\n\n Matching order of samples between counts and sampleTable \n")
counts<-counts[sampleTable$sample]

cat("\n Is the sample orders identical between counts matrix and sampletable \n")
identical(colnames(counts),sampleTable$sample)

cat("\n\n Convert metadata in categorical data whereever applicable\n")
sampleTable$condition<-as.factor(sampleTable$condition)

str(sampleTable)
```

```
##
##
##  Matching order of samples between counts and sampleTable
##
##  Is the sample orders identical between counts matrix and sampletable
## [1] TRUE
##
##
##  Convert metadata in categorical data whereever applicable
## 'data.frame':    12 obs. of  5 variables:
##  $ sample         : chr  "Control-1" "Control-2" "Control-3" "PNA-10b-1" ...
##  $ condition      : Factor w/ 4 levels "control","PNA.10b",..: 1 1 1 2 2 2 3 3 3 4 ...
##  $ Cells          : chr  "U87" "U87" "U87" "U87" ...
##  $ treat.info     : chr  "mock" "mock" "mock" "sgPNA-10b/BNP (150 nM sgPNA-10b)" ...
##  $ Incubation.time: chr  "72h" "72h" "72h" "72h" ...
```

QC2. QC of counts distribution per sample.

Sample: Count distribution Violin plot

Violin Count Plot



Quality check of samples to identify outliers or odd behaving samples.

```
df_qc<-skim(counts)
head(df_qc)
```

Table 1: Data summary

Name	counts
Number of rows	15036
Number of columns	12
Column type frequency:	
numeric	6
Group variables	None

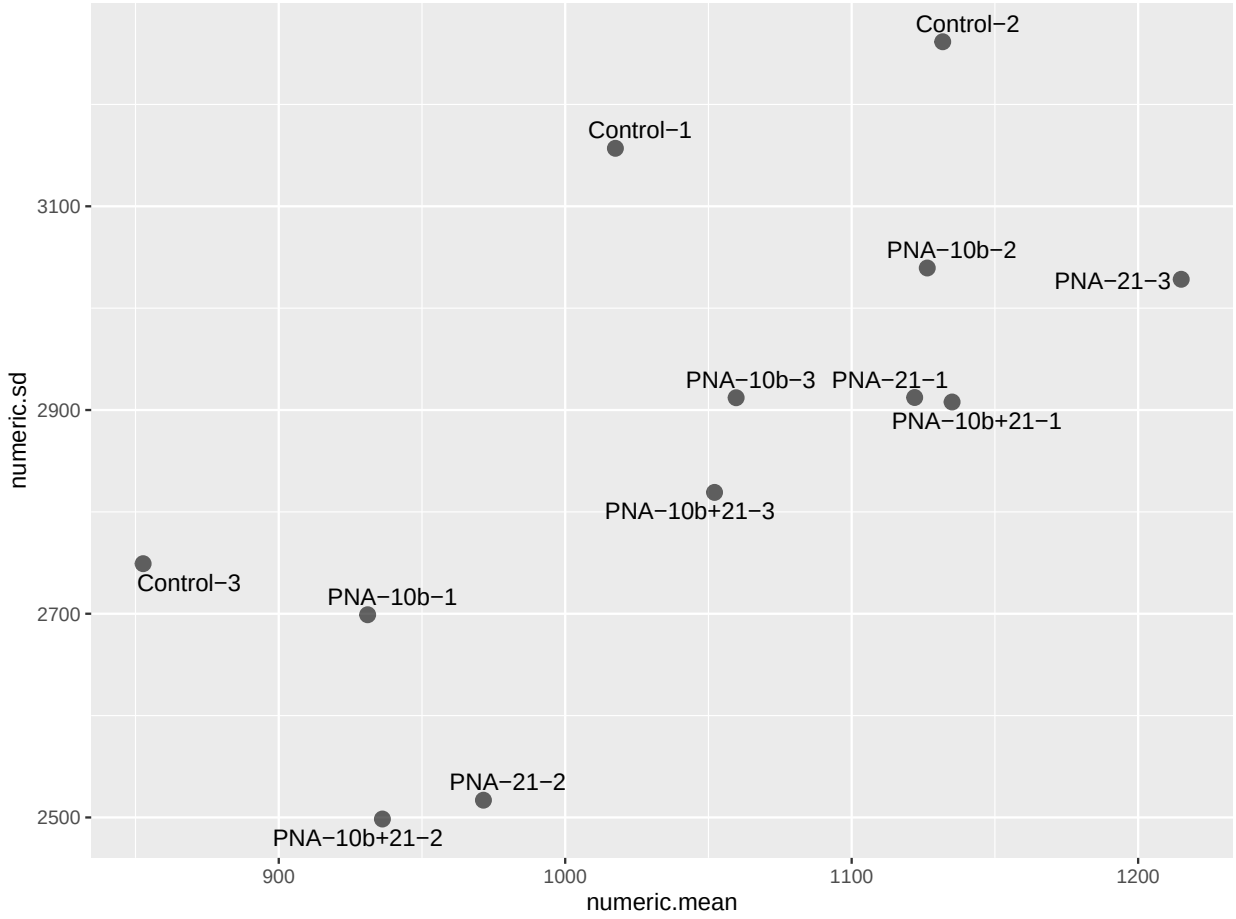
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Control-1	0	1	1017.52	3156.94	1	60	347	1009.00	197719	
Control-2	0	1	1131.79	3261.55	0	70	395	1155.00	192146	
Control-3	0	1	852.65	2749.13	0	50	283	836.00	180138	
PNA-10b-1	0	1	931.03	2699.00	0	54	318	937.00	145053	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
PNA-10b-2	0	1	1126.39	3039.44	0	66	393	1157.25	149187	
PNA-10b-3	0	1	1059.67	2912.07	0	65	376	1092.00	153101	

Look at the distribution of mean and std.deviation of counts across all samples. #### Sample Counts : Mean and Standard Deviation {.tabset}

Counts Mean and Standard deviation plot



DESeq2

Create DESeq2 Object In this analysis `control` samples are used as refrence condition for statistical analysis.

```
# reassuring that order of samples in rows of sampleTable are identical with the sample order in count
counts<-counts[sampleTable$sample]

dds <- DESeqDataSetFromMatrix(countData = as.matrix(counts),
                              colData=sampleTable,
                              design = ~condition)

# Setting "control" as baseline or reference
dds$condition<-relevel(dds$condition, ref=ref_level)
```

```
dds
```

```
## class: DESeqDataSet
## dim: 15036 12
## metadata(1): version
## assays(1): counts
## rownames(15036): ENSG00000000003 ENSG00000000419 ... ENSG00000288380
##   ENSG00000288398
## rowData names(0):
## colnames(12): Control-1 Control-2 ... PNA-10b+21-2 PNA-10b+21-3
## colData names(5): sample condition Cells treat.info Incubation.time
```

```
dds<-DESeq(dds)
```

Executing DESeq2

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

Accessing results

```
resultsNames(dds)
```

```
## [1] "Intercept"                      "condition_PNA.10b_vs_control"
## [3] "condition_PNA.21_vs_control"    "condition_PNA10b.21_vs_control"
```

```
## Access and saving normalised counts
normcounts<-as.data.frame(counts(dds,normalized =TRUE))

normAnnot<-merge(normcounts,ref, by=0, all.x=T)
normAnnot = data.frame(lapply(normAnnot, as.character), stringsAsFactors=FALSE)
write.csv(normAnnot,file="NormalisedCounts.csv")
```

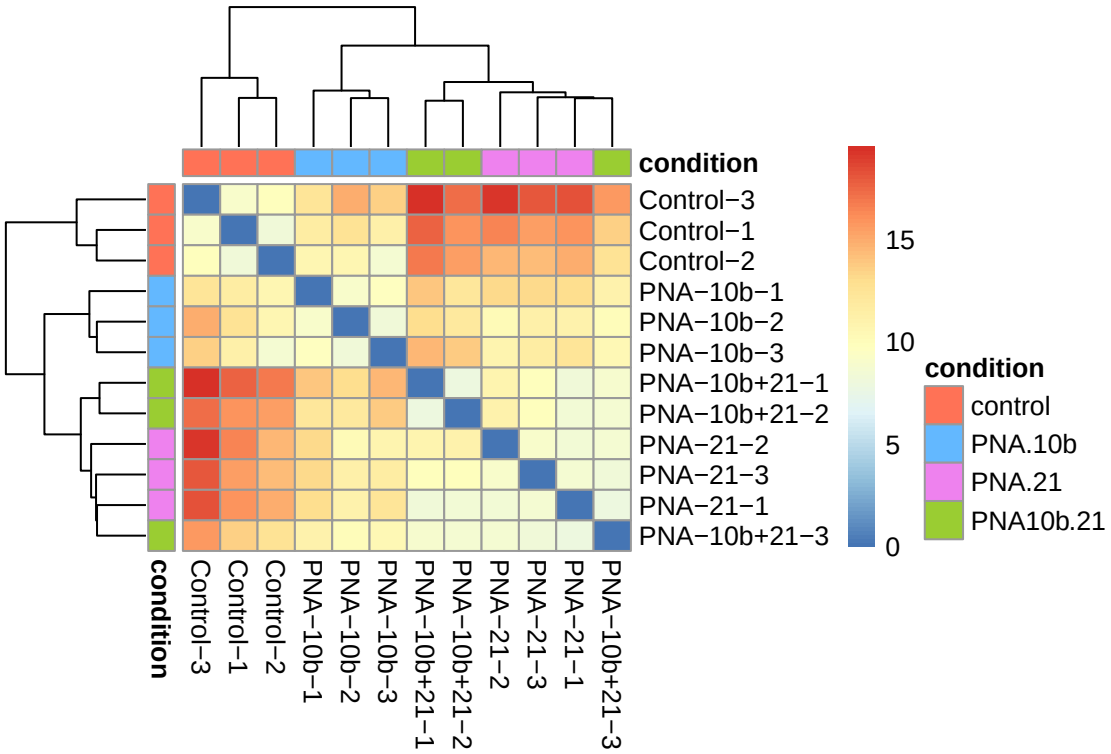
Sample QC

Sample to sample Distance Matrix

```
rld <- rlogTransformation(dds, blind=T)
vsd <- varianceStabilizingTransformation(dds, blind=T)

sampleDists <- dist(t(assay(rld)))
sampleDistMatrix <- as.matrix(sampleDists)

sampleDistMatrix_meta<-merge(sampleDistMatrix,sampleTable, by=0, sort=FALSE) %>% column_to_rownames(., v

head(sampleDistMatrix_meta)
```

```
##           Control-1 Control-2 Control-3 PNA-10b-1 PNA-10b-2 PNA-10b-3 PNA-21-1
## Control-1  0.000000  8.236115  8.973778 11.635699 12.670925 11.201484 15.80339
```

```
## Control-2  8.236115  0.000000  9.925959 10.582452 10.683862  8.744257 14.98897
## Control-3  8.973778  9.925959  0.000000 12.316396 14.912406 13.632284 18.27778
## PNA-10b-1 11.635699 10.582452 12.316396  0.000000  8.987803  9.697771 13.04147
## PNA-10b-2 12.670925 10.683862 14.912406  8.987803  0.000000  8.269350 11.07722
## PNA-10b-3 11.201484  8.744257 13.632284  9.697771  8.269350  0.000000 12.47302
##           PNA-21-2 PNA-21-3 PNA-10b+21-1 PNA-10b+21-2 PNA-10b+21-3    sample
## Control-1 16.44619 15.43151     17.61706     15.83128     13.62169 Control-1
## Control-2 14.57457 14.25682     16.91665     15.41219     12.60708 Control-2
## Control-3 19.14159 18.03891     19.49751     17.31771     15.69635 Control-3
## PNA-10b-1 13.18434 13.24462     14.01948     12.25626     11.07634 PNA-10b-1
## PNA-10b-2 10.24573 11.23835     12.88258     11.95172     10.00508 PNA-10b-2
## PNA-10b-3 10.90777 11.64915     14.53552     13.77476     10.52523 PNA-10b-3
##           condition Cells                          treat.info Incubation.time
## Control-1   control   U87                                mock             72h
## Control-2   control   U87                                mock             72h
## Control-3   control   U87                                mock             72h
## PNA-10b-1   PNA.10b   U87 sgPNA-10b/BNP (150 nM sgPNA-10b)             72h
## PNA-10b-2   PNA.10b   U87 sgPNA-10b/BNP (150 nM sgPNA-10b)             72h
## PNA-10b-3   PNA.10b   U87 sgPNA-10b/BNP (150 nM sgPNA-10b)             72h
```



Principal component Analysis

Principal Component Analysis (PCA) is a widely used statistical method in the analysis of differential gene expression data. It helps in understanding the underlying structure of high-dimensional data, like gene expression profiles, by reducing its complexity. PCA achieves this by transforming the original variables into a new set of uncorrelated variables called principal components, which are ordered so that the first few retain most of the variation present in all of the original variables.

In the context of differential gene expression, PCA is particularly useful for several reasons:

1. Data Visualization: PCA allows for the visualization of complex gene expression data in two or three

dimensions. This can help in identifying patterns, trends, or clusters in the data that might indicate similarities or differences in gene expression under various conditions or among different samples.

2. Noise Reduction: By focusing on principal components that capture the most variance, PCA can help filter out noise and reduce the dimensionality of the data, making further analysis more tractable.
3. Identification of Important Genes: The loading scores of the principal components can be used to identify genes that contribute most to the variance in the data. These genes can be candidates for further study as they might play significant roles in the biological processes or conditions being studied.
4. Batch Effect Correction: PCA can help in identifying and correcting for batch effects, which are technical non-biological differences between batches of samples that can affect gene expression levels.
5. Comparative Analysis: By comparing the PCA results of different conditions or experiments, researchers can identify shifts in the principal component space, indicating changes in gene expression profiles associated with different biological states or treatments.

```
rv <- rowVars(assay(rld))
select <- order(rv, decreasing=T)[seq_len(min(500,length(rv)))]
pc <- prcomp(t(assay(vsd)[select,]))
scores <- data.frame(pc$x)
scores<-cbind(scores,sampleTable)
summary(pc)
```

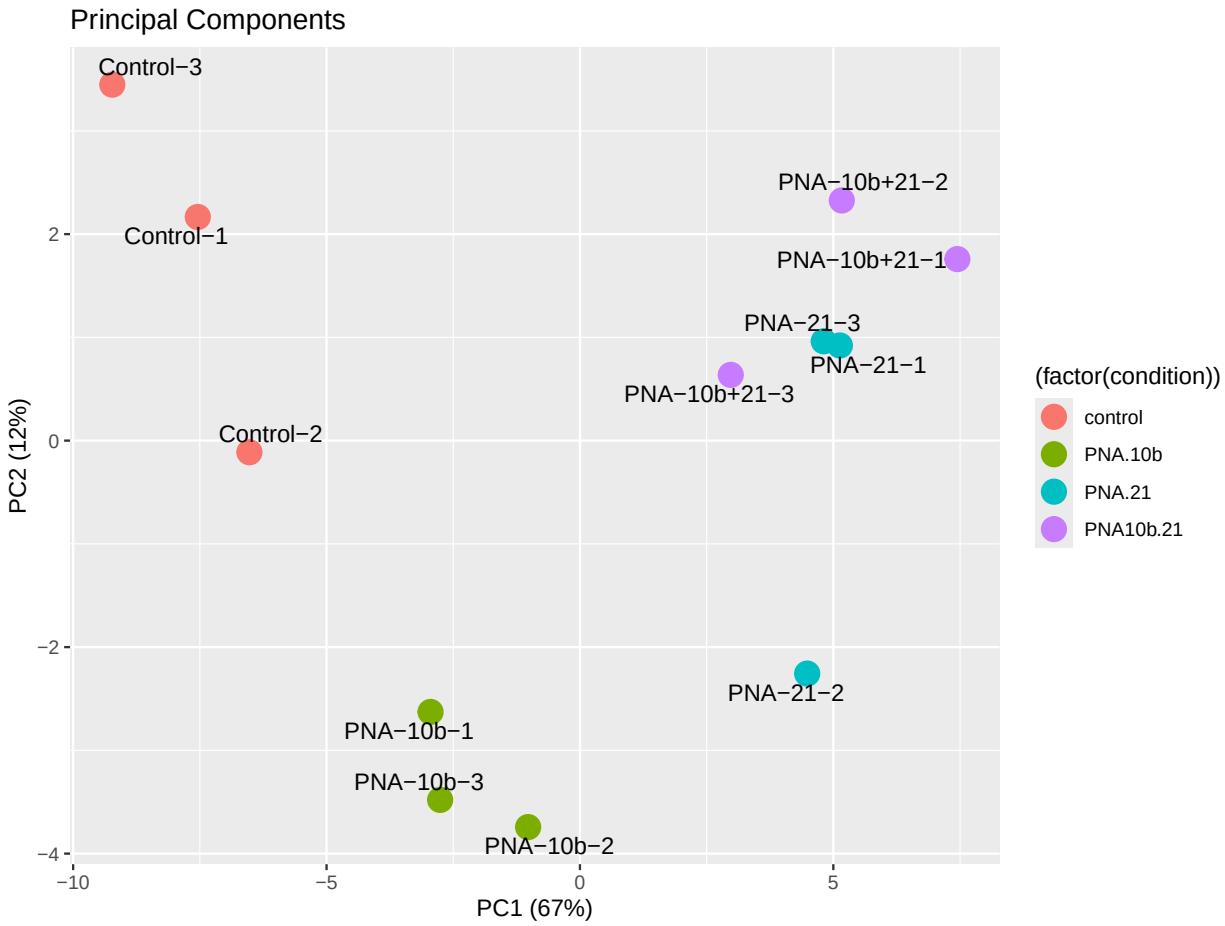
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     5.7403 2.4383 1.69291 1.35859 1.16329 1.04655 0.95398
## Proportion of Variance 0.6719 0.1212 0.05844 0.03763 0.02759 0.02233 0.01856
## Cumulative Proportion  0.6719 0.7931 0.85152 0.88915 0.91674 0.93907 0.95763
##                           PC8     PC9    PC10    PC11      PC12
## Standard deviation     0.80214 0.74085 0.68803 0.64212 2.553e-14
## Proportion of Variance 0.01312 0.01119 0.00965 0.00841 0.000e+00
## Cumulative Proportion  0.97075 0.98194 0.99159 1.00000 1.000e+00
```

```
pc1prcnt=as.character(as.integer(summary(pc)$importance[2,1]*100))
pc2prcnt=as.character(as.integer(summary(pc)$importance[2,2]*100))
pc3prcnt=as.character(as.integer(summary(pc)$importance[2,3]*100))
pc4prcnt=as.character(as.integer(summary(pc)$importance[2,4]*100))
pc5prcnt=as.character(as.integer(summary(pc)$importance[2,5]*100))
pc6prcnt=as.character(as.integer(summary(pc)$importance[2,6]*100))

names<-rownames(scores)

pc12 <-ggplot(scores, aes(x = PC1, y = PC2, label=names,col = (factor(condition))))+
  geom_point(size = 5)+
  geom_text_repel(label=names,col="black")+
  ggtitle("Principal Components")+
  xlab(paste0("PC1 (",pc1prcnt,"%)")) +
  ylab(paste0("PC2 (",pc2prcnt,"%)"))

pc34 <-ggplot(scores, aes(x = PC3, y = PC4, label=names,col = (factor(condition))))+
  geom_point(size = 5)+
  geom_text_repel(label=names,col="black")+
  ggtitle("Principal Components")+
  xlab(paste0("PC3 (",pc3prcnt,"%)")) +
  ylab(paste0("PC4 (",pc4prcnt,"%)"))
```

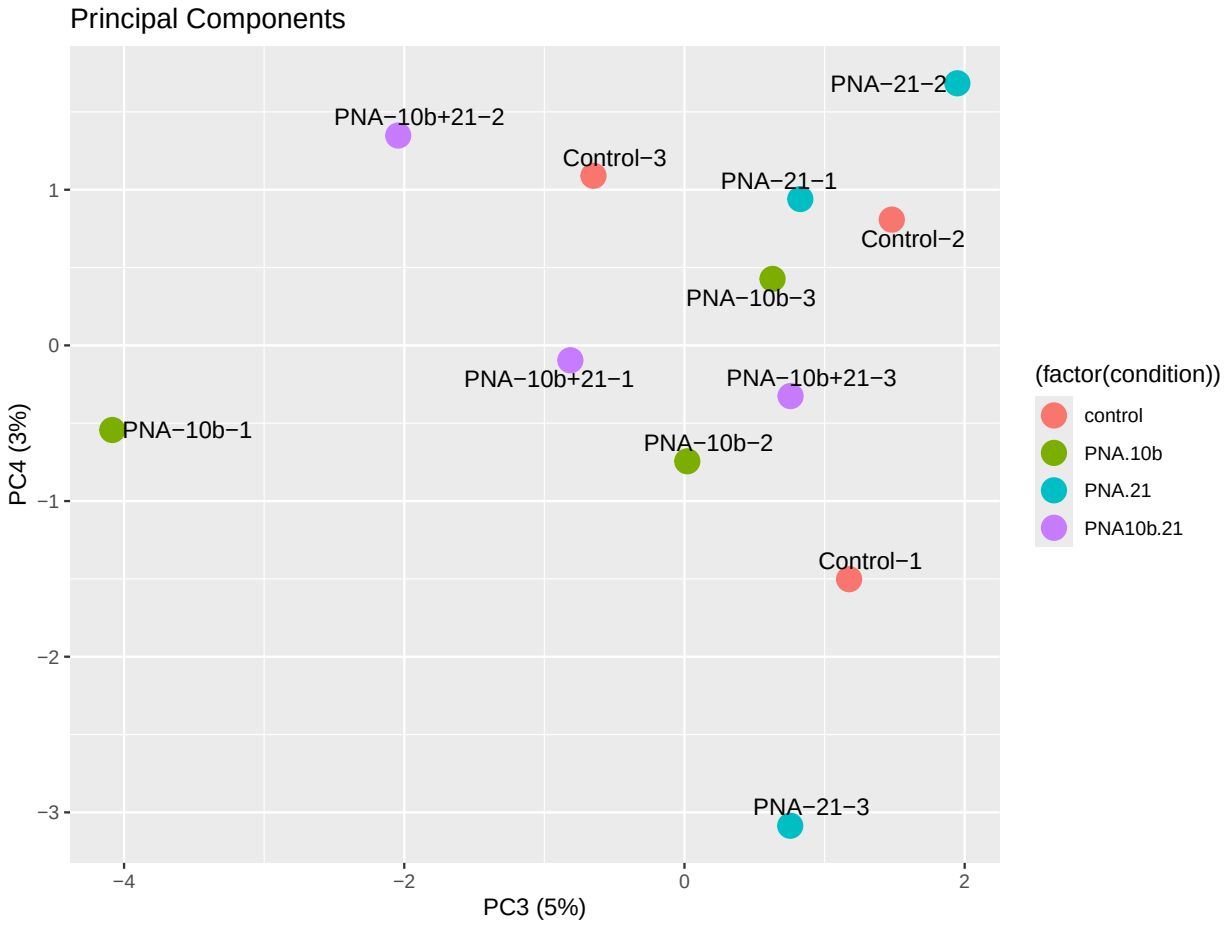
```
pc56 <-ggplot(scores, aes(x = PC5, y = PC6, label=names,col = (factor(condition))))+
  geom_point(size = 5)+
  geom_text_repel(label=names,col="black")+
  ggtitle("Principal Components")+
  xlab(paste0("PC5 (",pc5prcnt,"%)")) +
  ylab(paste0("PC6 (",pc6prcnt,"%)"))
```

PCA Plots

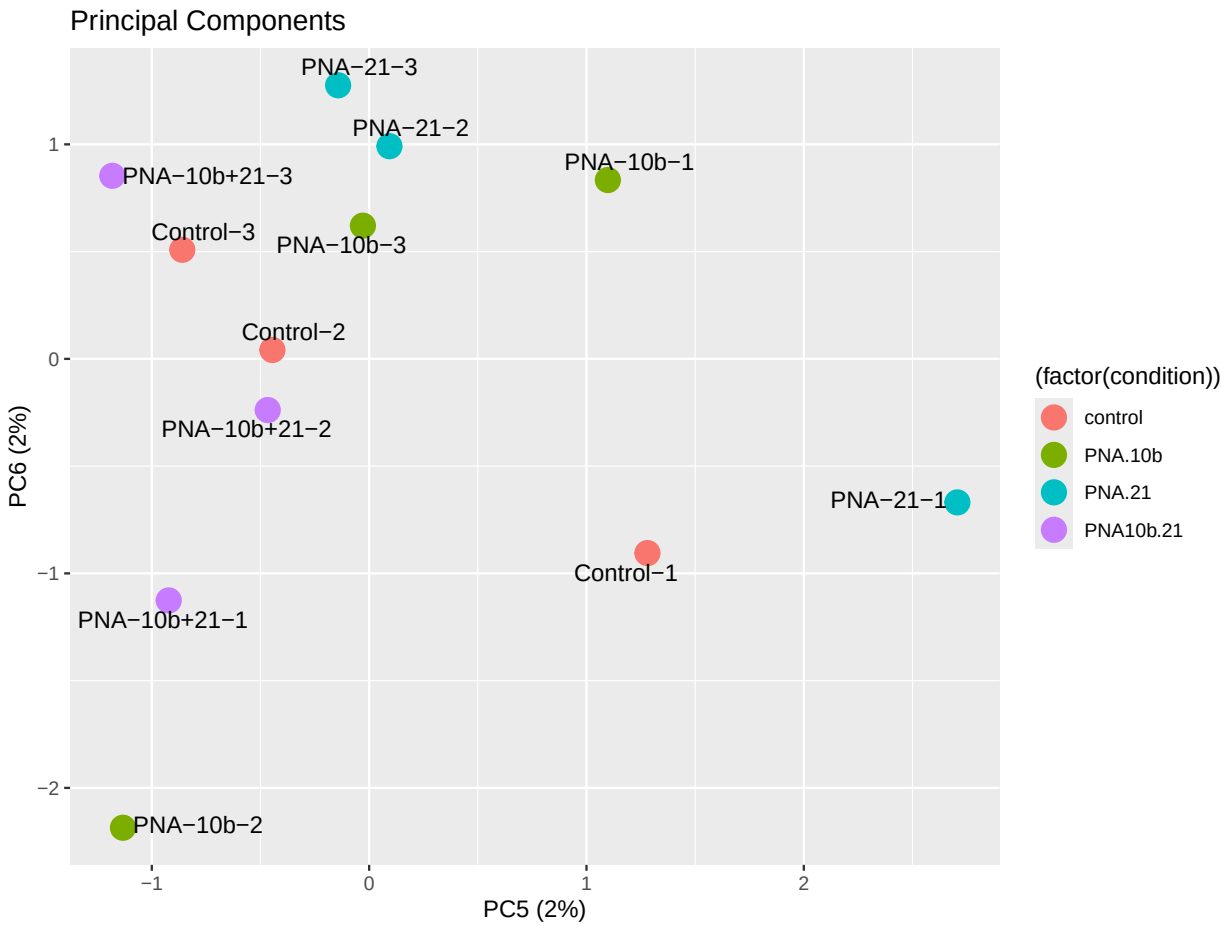
Plot PC12



Plot PC34


Principal Components
PNA−21−2
PNA−10b+21−2
Control−3
PNA−21−1
Control−2
PNA−10b−3
PNA−10b+21−1
PNA−10b+21−3
PNA−10b−1
PNA−10b−2
Control−1
PNA−21−3
PC4 (3%)
PC3 (5%)
1
0
−1
−2
−3
−4
−2
0
2
(factor(condition))
control
PNA.10b
PNA.21
PNA10b.21

Plot PC56



MA_Volcano.Plots

```
comparison_matrix<-resultsNames(dds)[-1]

source("../../DataAnalysis/Rscripts/volcanoCode_updatedApr2024.R")

resout<-list()
shrink_plots_list <- list()
volcano_plot_list <- list()
ma_plot_list <- list()

padj_threshold=0.05

lf_raw_shrink_plot<-function(raw_res,shrink_res,padj_threshold=0.05){
  plotdf<-data.frame("lfraw"=raw_res$log2FoldChange,"lfshrink"=shrink_res$log2FoldChange)
  plotdf["significant"]<-as.factor(ifelse(raw_res$padj<padj_threshold & abs(shrink_res$log2FoldChange)>0
  ggplot(plotdf,aes(lfraw,lfshrink,col=significant)) + geom_point()+
    ggtitle(paste0("Raw.Log2FC vs Shrinked.Log2FC","_",AnalysisID))
}

for (i in 1:3){
  AnalysisID <- paste0(comparison_matrix[i],"_14March2024")
  raw_res <- results(dds, name = comparison_matrix[i])
```

```
  shrink_res <- lfcShrink(dds,type="ashr",coef = comparison_matrix[i],quiet=TRUE)
  raw_res_anno <- merge(as.data.frame(raw_res) ,ref, by=0, all.x=TRUE, sort=FALSE)
  shrink_res_anno <- merge(as.data.frame(shrink_res),ref, by=0, all.x=TRUE,sort=FALSE)
  resout[[paste0(AnalysisID,"_raw")]] <- raw_res_anno
  resout[[paste0(AnalysisID,"_shrink")]] <- shrink_res_anno
  plotdf<-data.frame("lfraw"=raw_res$log2FoldChange,"lfshrink"=shrink_res$log2FoldChange)
  plotdf["significant"]<-as.factor(ifelse(raw_res$padj<padj_threshold &   abs(shrink_res$log2FoldChange

  raw_shrink_plots<- ggplot(plotdf,aes(lfraw,lfshrink,col=significant)) + geom_point()+
    ggtitle(paste0("Raw.Log2FC vs Shrinked.Log2FC","_",AnalysisID))


   Vol_plot<-volcanoPlot(shrink_res_anno,
            geneName.col="gene_name",
            plot.title=paste0("Shrink_Volcano Plot :",AnalysisID),
                              lfc_cut_off=0.58)

   volcano_plot_list[[AnalysisID]] <- Vol_plot

   raw_Vol_plot<-volcanoPlot(raw_res_anno,
            geneName.col="gene_name",
            plot.title=paste0("Raw_Volcano Plot :",AnalysisID),
                              lfc_cut_off=0.58)

   #raw_ma <- plotMA(raw_res, ylim=c(-2,2))
   #shrink_ma<- plotMA(shrink_res, ylim=c(-2,2))

   ma_plot_list[[paste0("RAW_",AnalysisID)]]<- raw_res

   ma_plot_list[[paste0("Shrink_",AnalysisID)]] <- shrink_res


   volcano_plot_list[[paste0("RAW_",AnalysisID)]] <- raw_Vol_plot

   shrink_plots_list[[paste0("Shrink_",AnalysisID)]] <- raw_shrink_plots

    raw_res_anno = data.frame(lapply(raw_res_anno, as.character), stringsAsFactors=FALSE)
   shrink_res_anno = data.frame(lapply(shrink_res_anno, as.character), stringsAsFactors=FALSE)
   write.csv(raw_res_anno, file=paste0(AnalysisID ,"_RawDE.csv"))
   write.csv(shrink_res_anno,file=paste0(AnalysisID ,"_ShrinkDE.csv"))
}
```

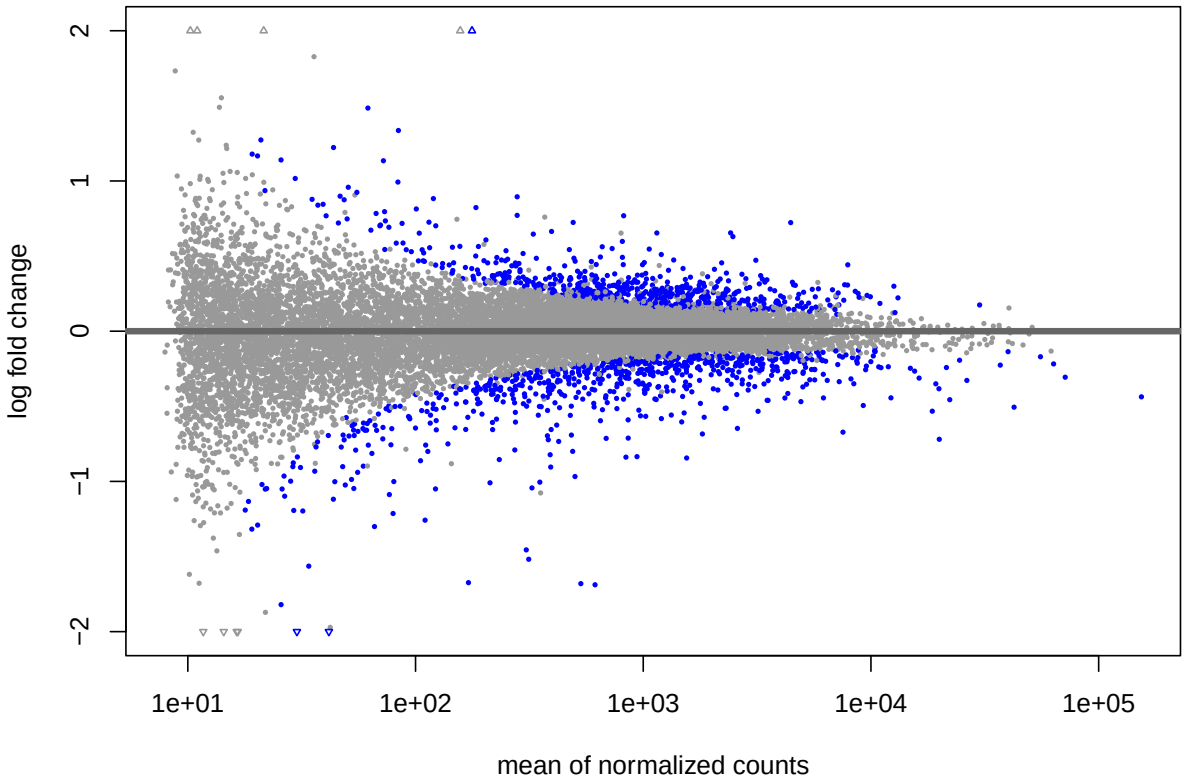
The lfcShrink function in DESeq2 is designed to perform shrinkage estimation of the log2 fold changes (log2FC) in differential expression analysis. DESeq2 is a popular R package used for analyzing count-based data, like RNA-Seq or other forms of genomic data that come in the form of read counts. The main goal of DESeq2 is to find differentially expressed genes between experimental conditions by modeling count data using negative binomial distributions.

The lfcShrink method is particularly useful because it addresses a common issue in differential expression analysis: the estimation of log2 fold changes, especially for genes with low counts or high variance, can be noisy, leading to overestimation of the true effect sizes. Shrinkage estimators pull the estimated log2 fold changes towards zero (or towards a specified prior if different from zero), reducing the influence of noisy estimates on downstream analysis and interpretation. This process tends to improve the stability and reliability of the log2 fold change estimates.

A plot that can be useful to exploring our results is the MA plot. The MA plot shows the mean of the normalized counts versus the log2 foldchanges for all genes tested. The genes that are significantly DE are colored to be easily identified. This is also a great way to illustrate the effect of LFC shrinkage. The DESeq2 package offers a simple function to generate an MA plot. That is, many of the low expressors exhibit very high fold changes. After shrinkage, we see the fold changes are much smaller estimates.

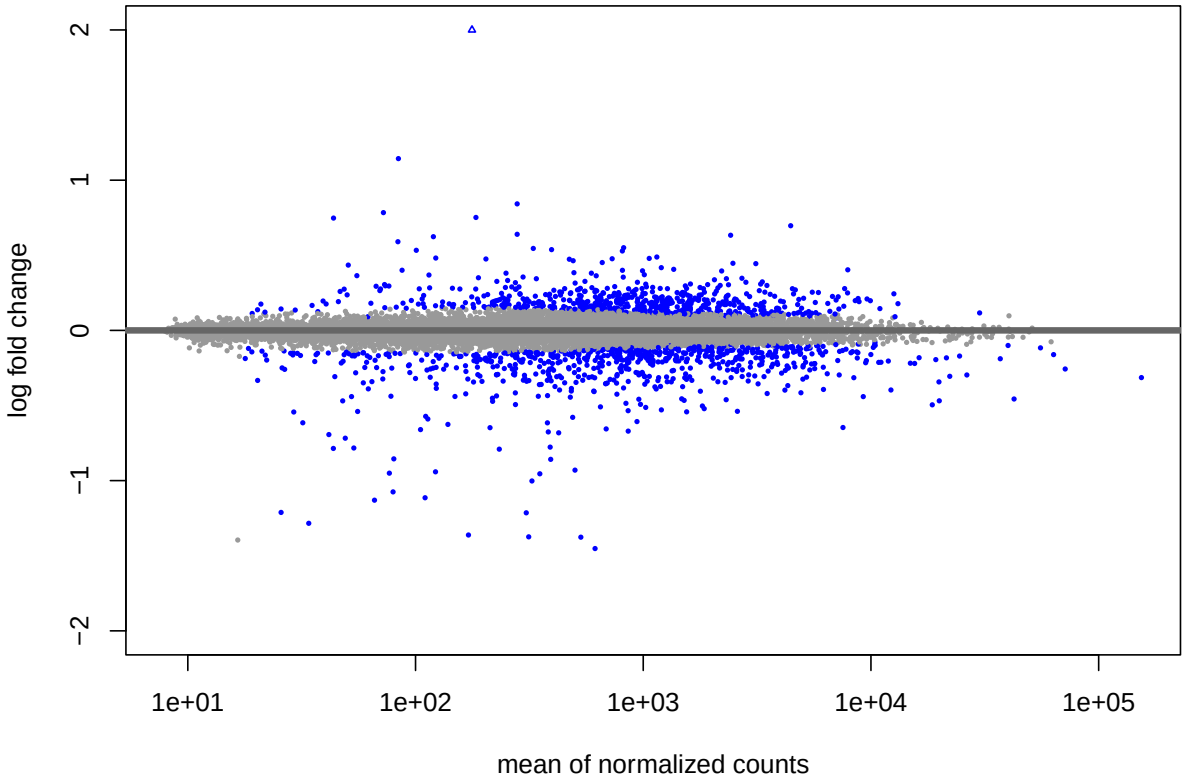
MA plot Raw and Shrink results

Plot : RAW_condition_PNA.10b_vs_control_14March2024



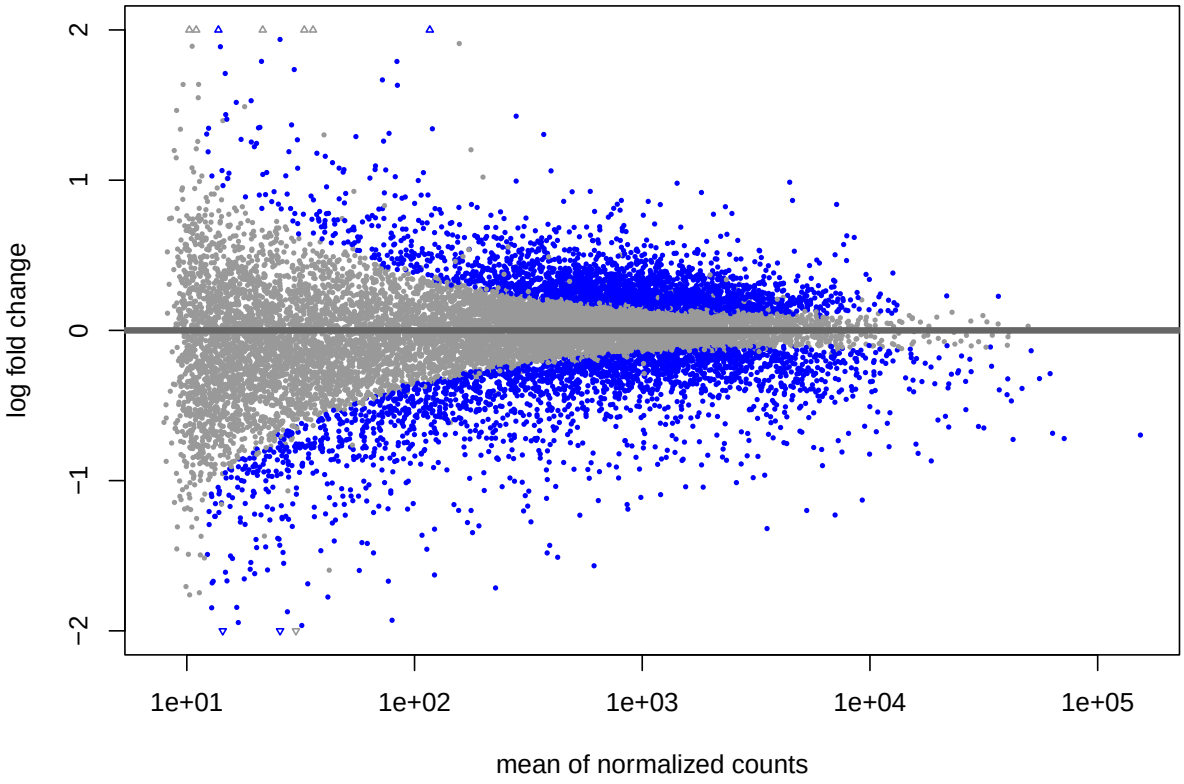
NULL

Plot : Shrink_condition_PNA.10b_vs_control_14March2024



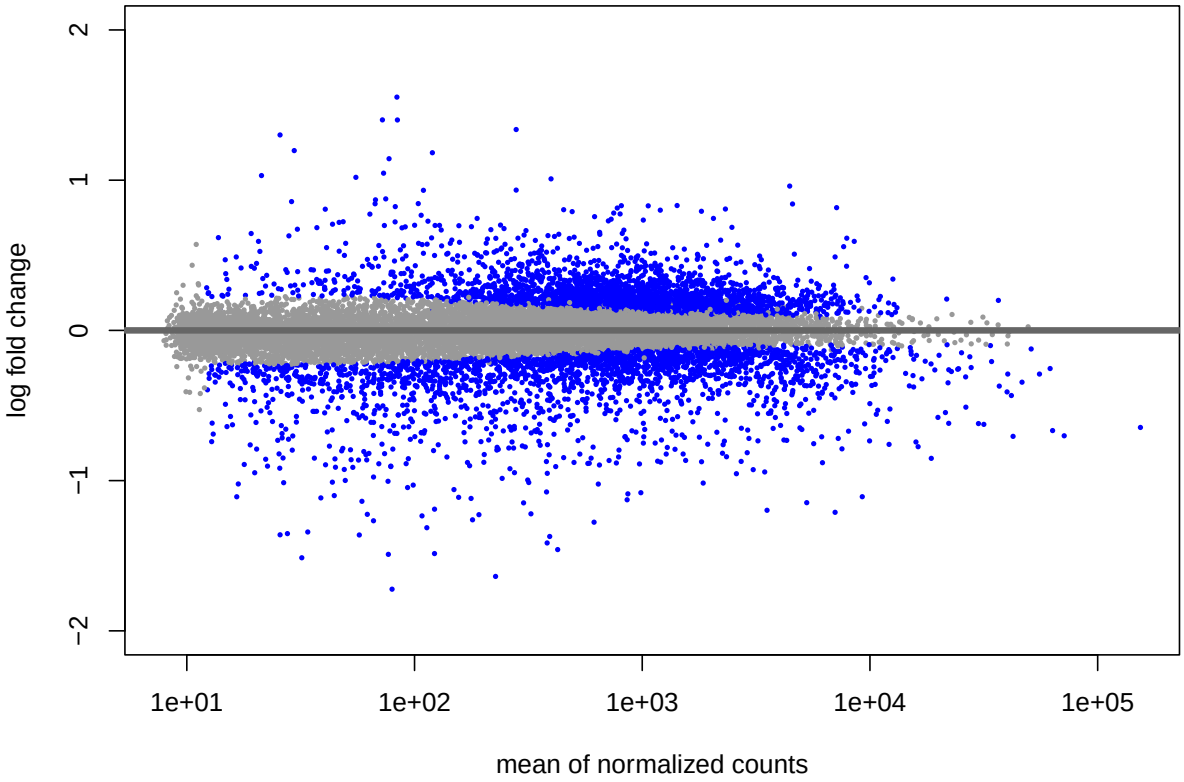
NULL

Plot : RAW_condition_PNA.21_vs_control_14March2024



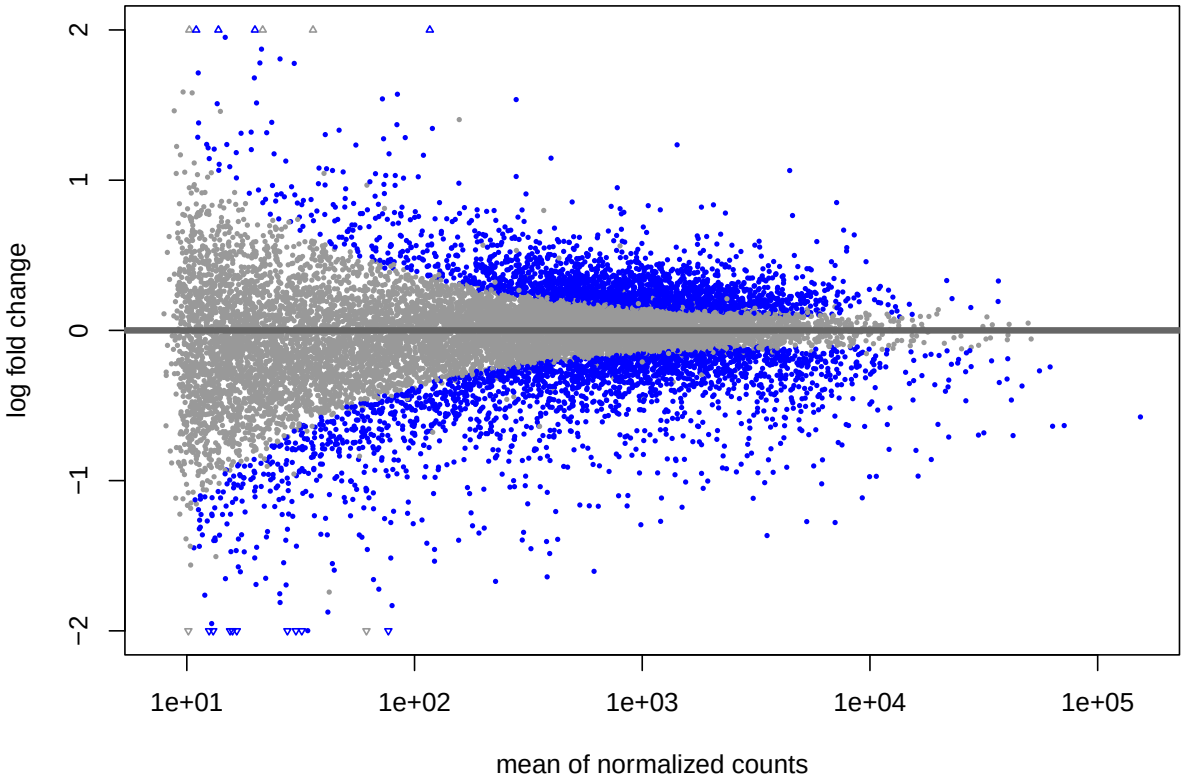
NULL

Plot : Shrink_condition_PNA.21_vs_control_14March2024



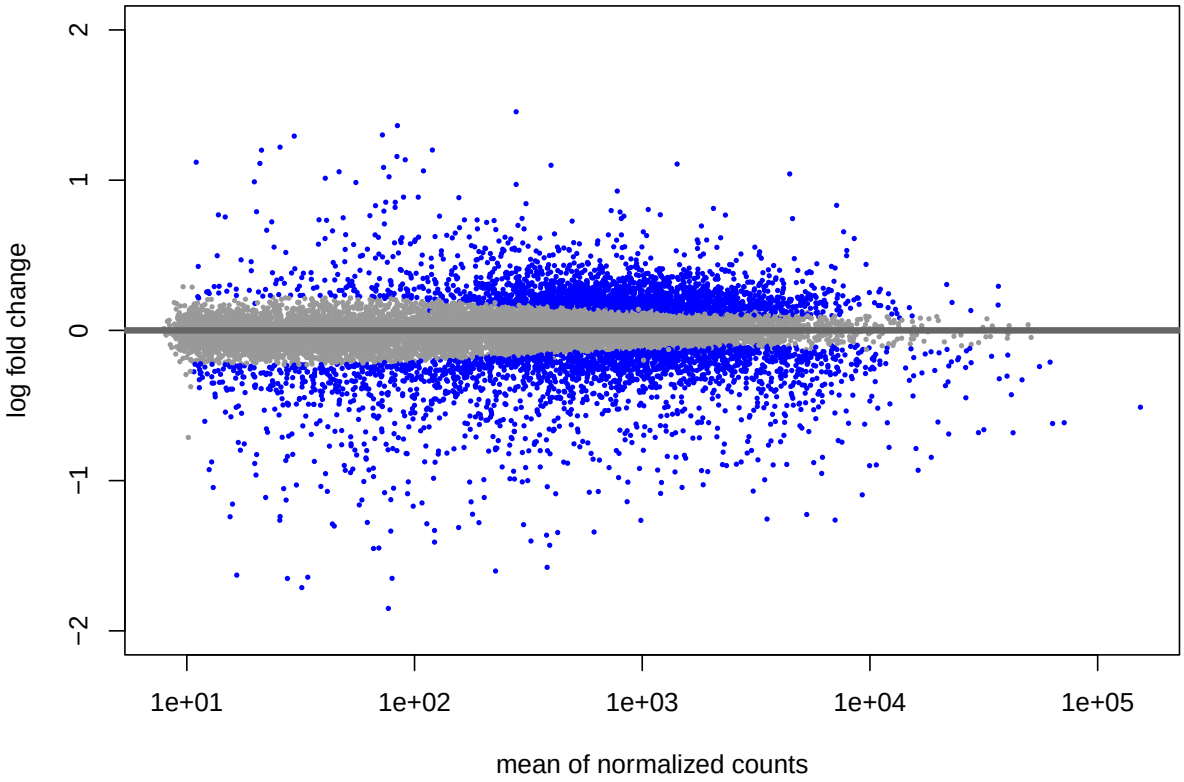
NULL

Plot : RAW_condition_PNA10b.21_vs_control_14March2024



NULL

Plot : Shrink_condition_PNA10b.21_vs_control_14March2024

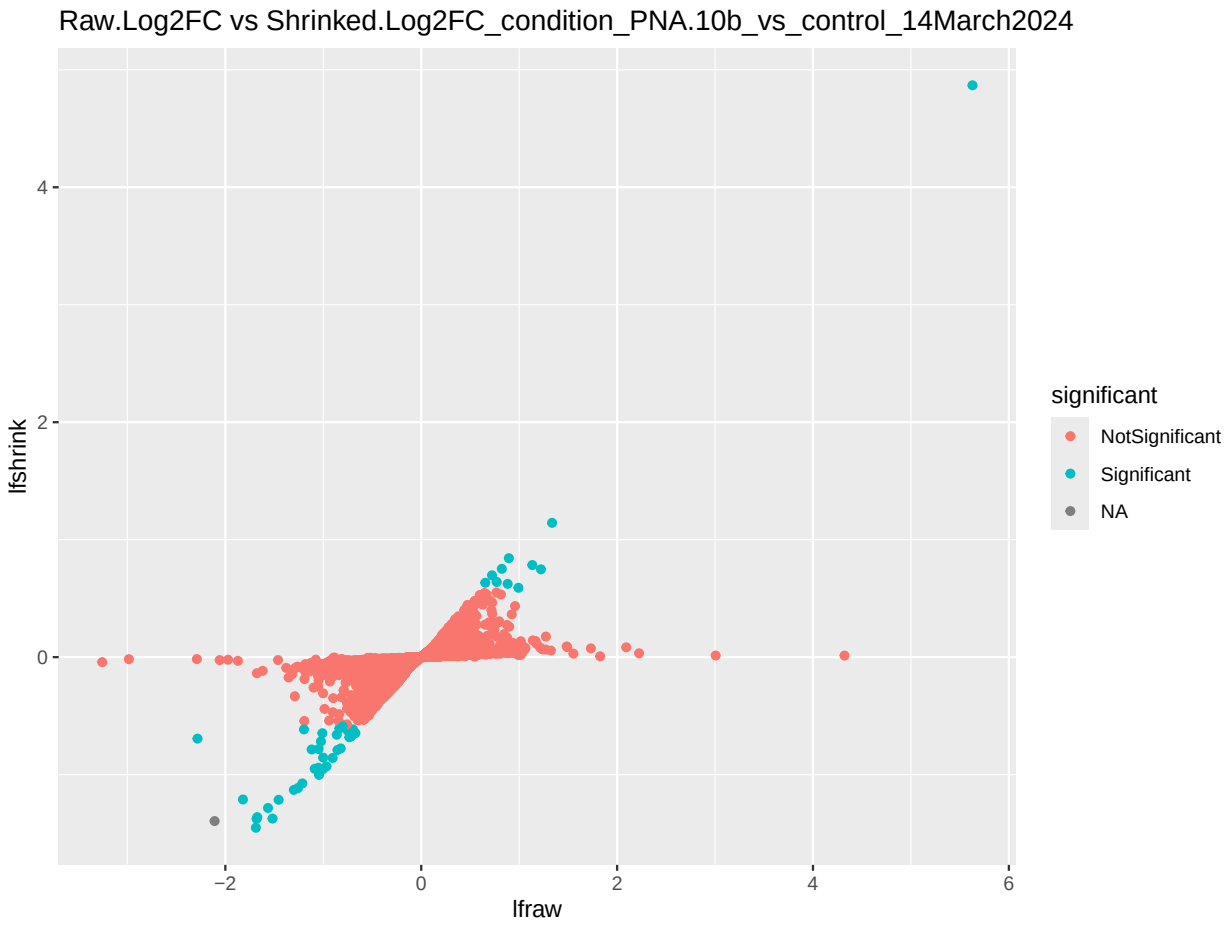


NULL

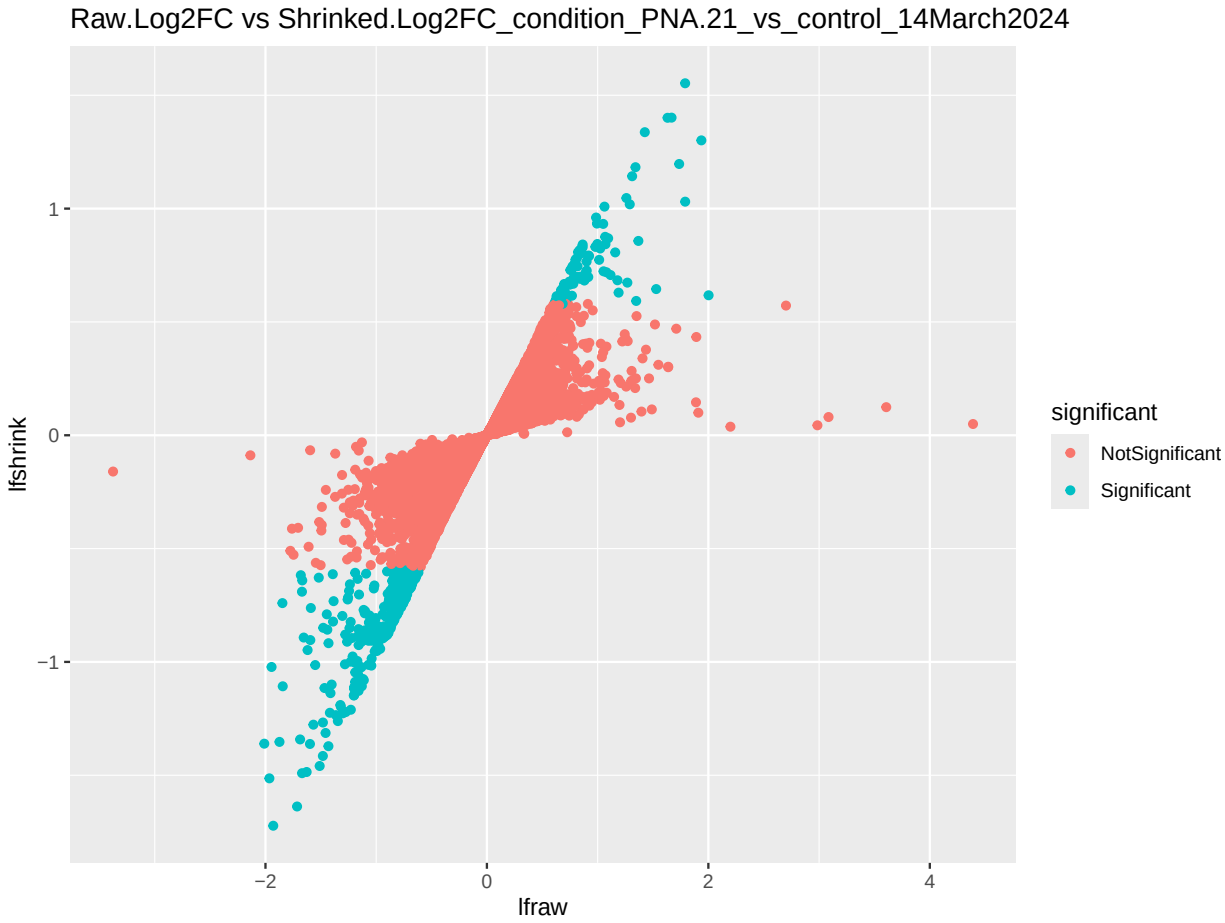
The plots below are scatter plot of Raw and shrink data to further illustrate effect of shrinkage.

Raw and Shrink log2FoldChange

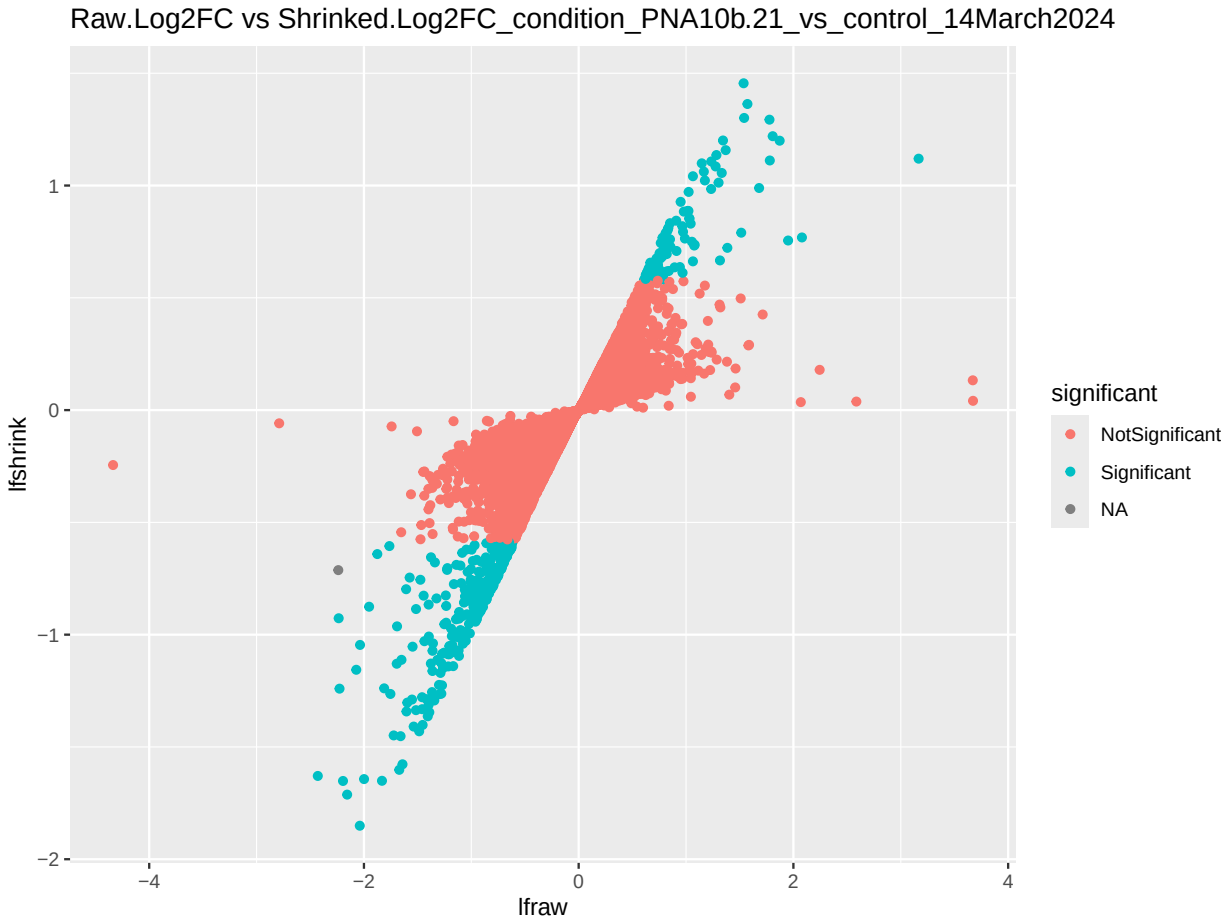
Plot : Shrink_condition_PNA.10b_vs_control_14March2024



Plot : Shrink_condition_PNA.21_vs_control_14March2024



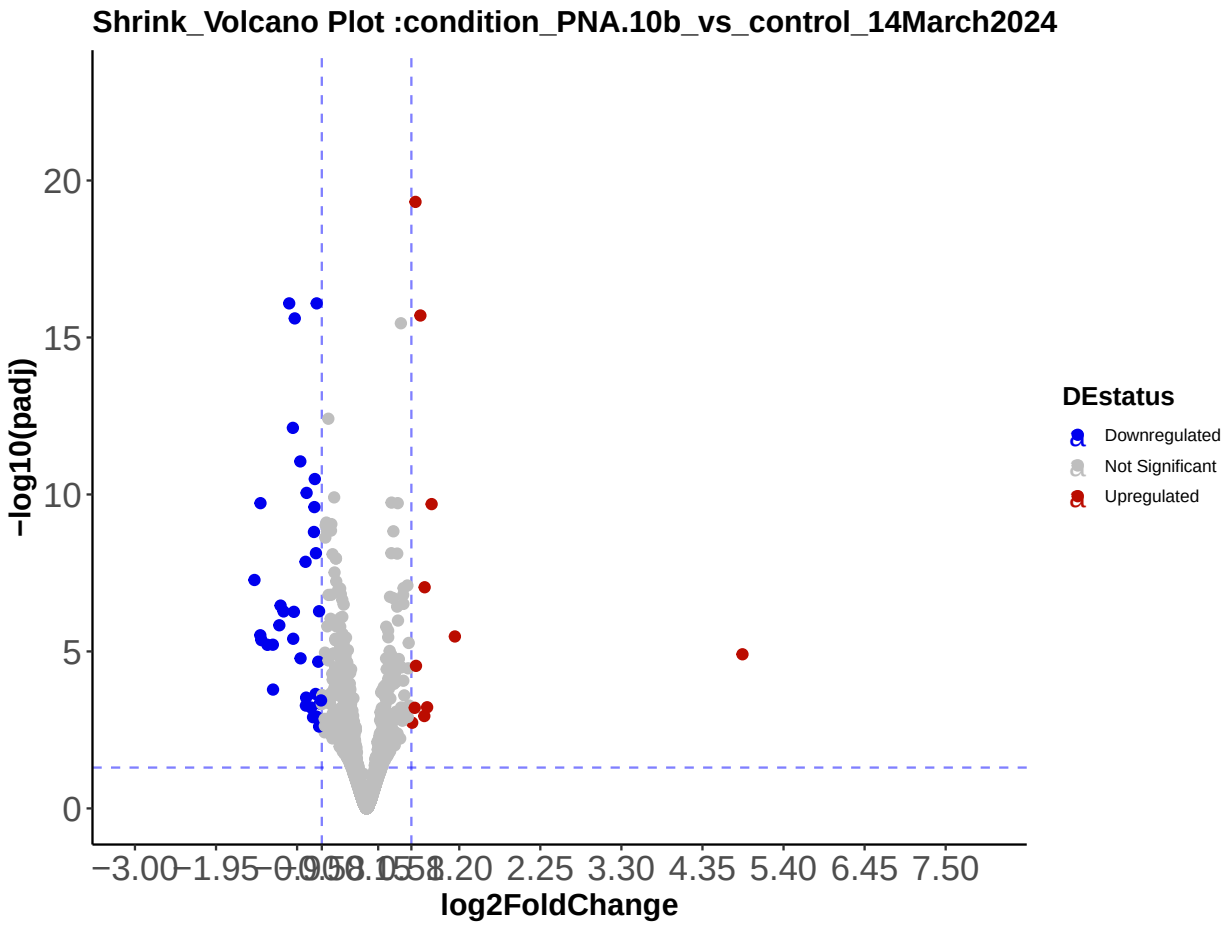
Plot : Shrink_condition_PNA10b.21_vs_control_14March2024



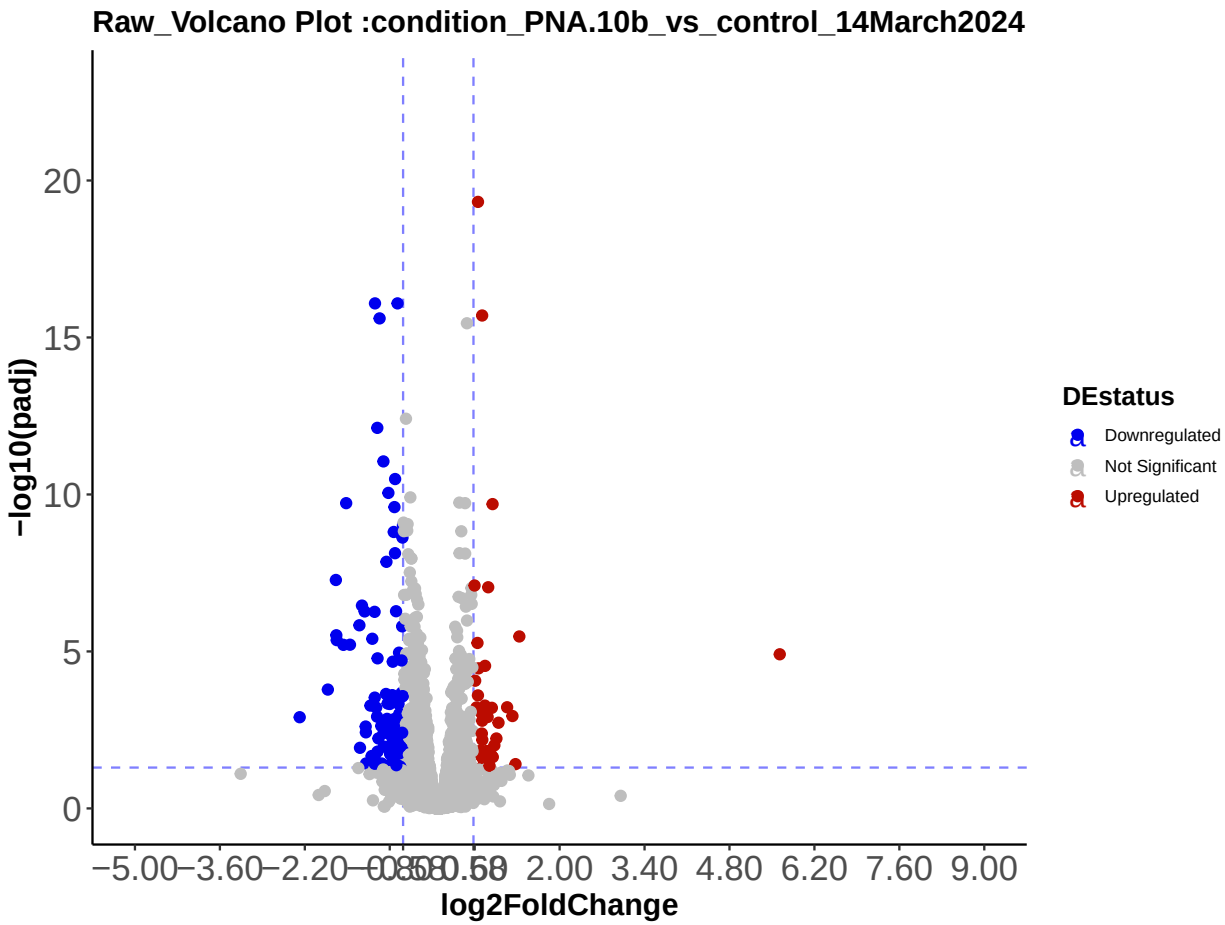
Volcano plot is a type of scatter plot that is commonly used in bioinformatics to visually display the results of differential expression analyses, among other applications. It plots significance versus fold-change on the y and x axes, respectively, making it an effective tool for quickly identifying genes that are significantly differentially expressed between two experimental conditions.

Volcano Plots

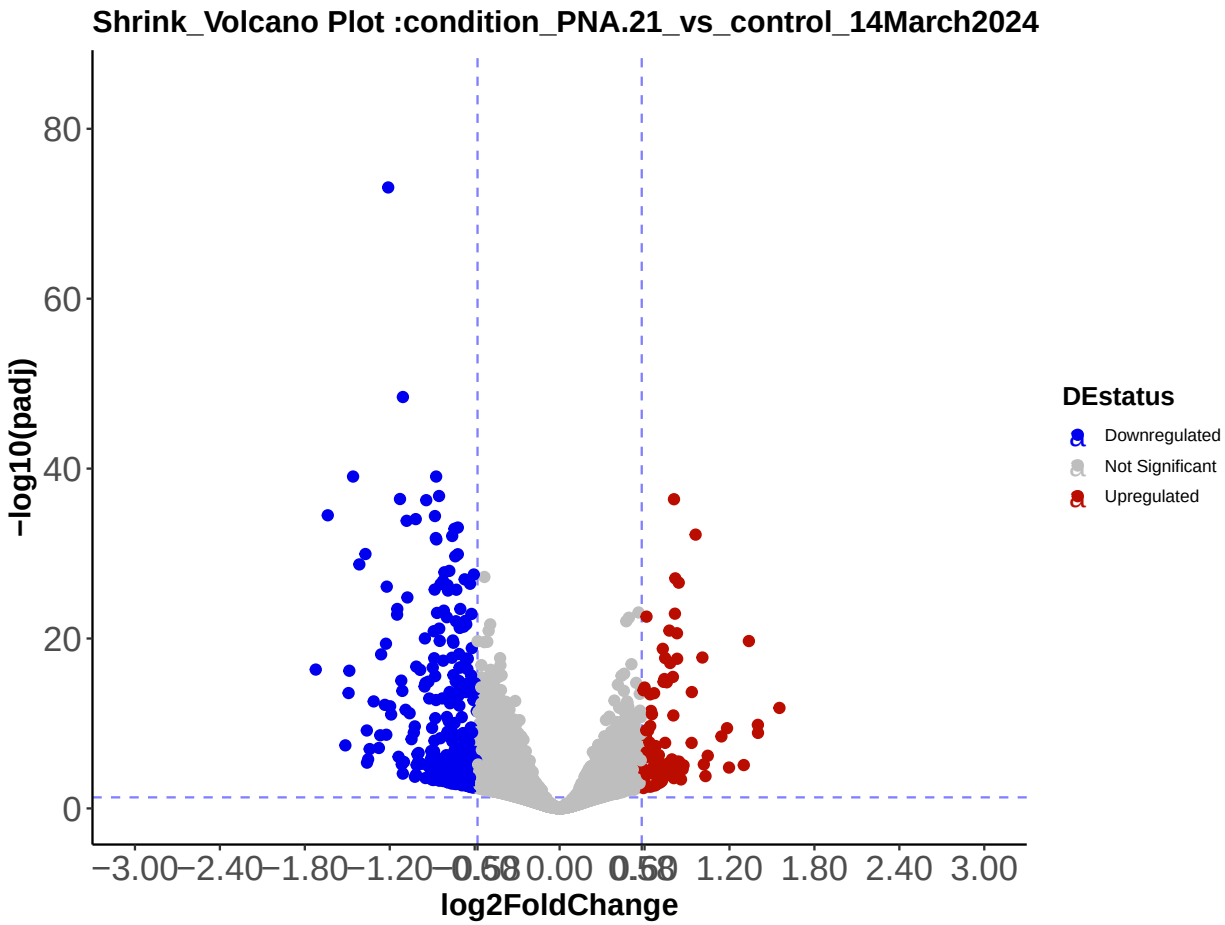
Plot : condition_PNA.10b_vs_control_14March2024



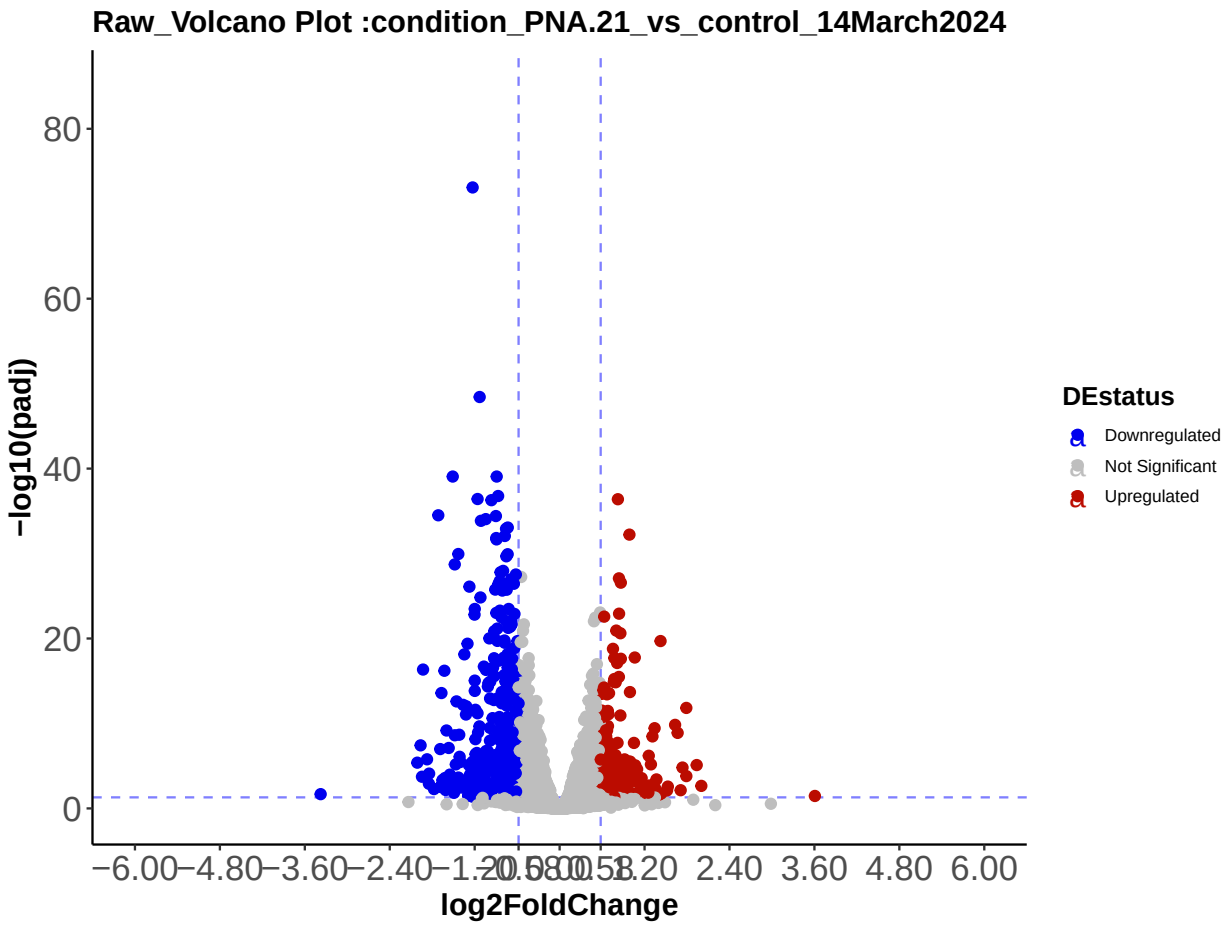
Plot : RAW_condition_PNA.10b_vs_control_14March2024



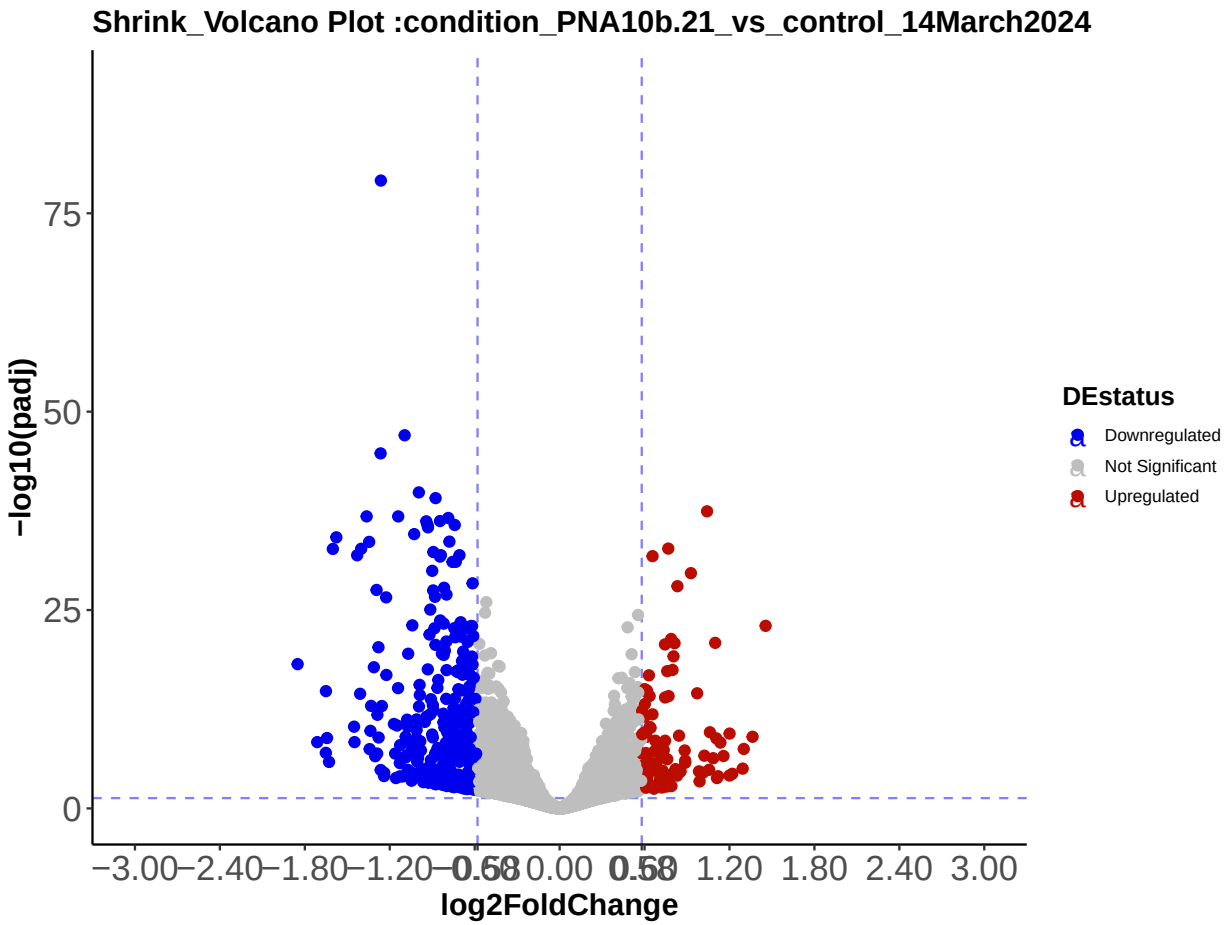
Plot : condition_PNA.21_vs_control_14March2024



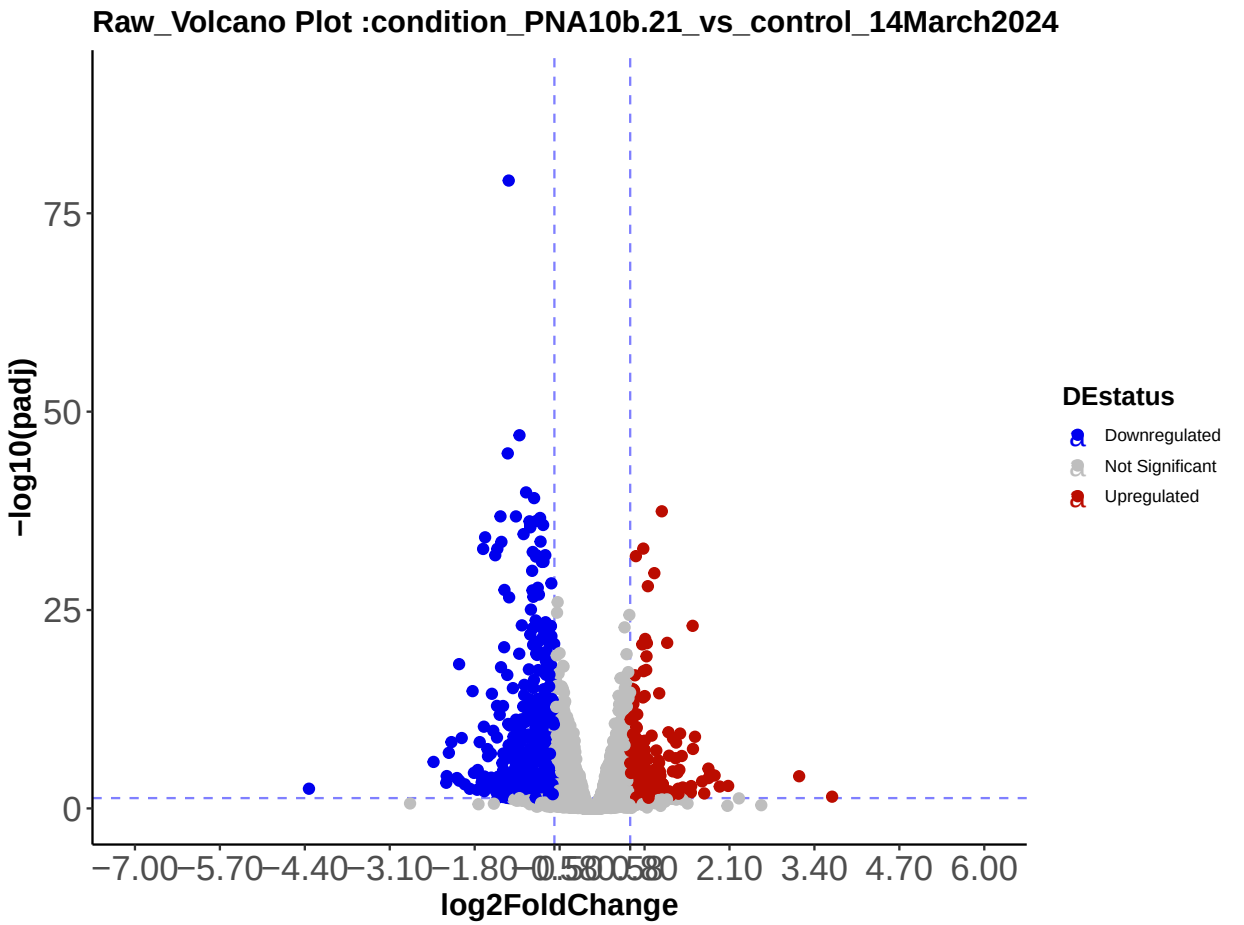
Plot : RAW_condition_PNA.21_vs_control_14March2024



Plot : condition_PNA10b.21_vs_control_14March2024



Plot : RAW_condition_PNA10b.21_vs_control_14March2024



GeneStats_VennDiagram

In this section we will explore genes that are differentially expressed between each comparison and the overlap.

```
rm(list=ls())

up_gene_list <- list()
down_gene_list <- list()
DE_gene_list <- list()
resultdf <- data.frame()

padj=0.05
log2fc=0.58
files<-list.files(".", pattern = "DE.csv$")
prefx<-c("PNA.10b_RAW","PNA.10b_shrink","PNA.21_Raw","PNA.21_shrink", "PNA10b.21_raw","PNA10b.21_shrink
for (i in 1:length(files)){
  fileName=files[i]
  pref=prefx[i]
  df<-read.csv(fileName) %>%
  column_to_rownames(., var="Row.names") %>%
  dplyr::select(.,log2FoldChange,padj,gene_name) %>%
  filter(.,padj < 0.05 & abs(log2FoldChange) > 0.58)
```

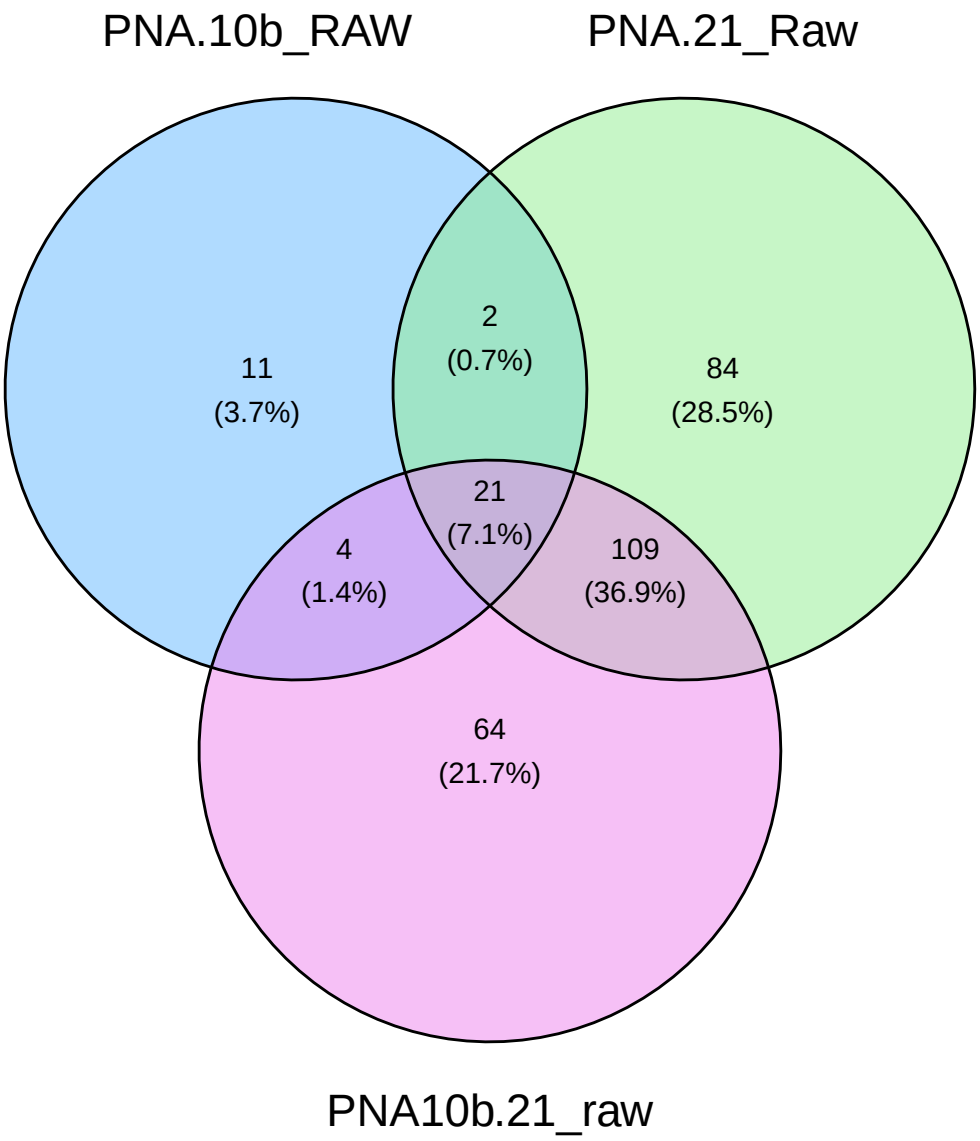
```
    up_gene_list[[pref]]<-rownames(df[df$log2FoldChange > 0.58,])
    down_gene_list[[pref]]<-rownames(df[df$log2FoldChange < (-0.58),])
    DE_gene_list[[pref]] <-rownames(df)

    resultdf<-rbind(resultdf,data.frame("Comparison"=pref,"Up"=length(up_gene_list[[pref]]), "Down"=leng
}

suppressPackageStartupMessages(library(ggvenn,quietly = TRUE))

vennplot<-list()

venngenelist<-list("up"=up_gene_list, "down"=down_gene_list, "DE"=DE_gene_list)

for(ven in names(venngenelist)){

p1<-ggvenn(
  venngenelist[[ven]][c(1,3,5)],
  fill_color = c("steelblue1","lightgreen","violet"),
  stroke_size = 0.5, set_name_size = 6
  )

p2<-ggvenn(
  venngenelist[[ven]][c(2,4,6)],
  fill_color = c("steelblue1","lightgreen","violet"),
  stroke_size = 0.5, set_name_size = 6
  )

vennplot[[paste0(ven,"_raw")]]<-p1
vennplot[[paste0(ven,"_shrink")]]<-p2
}

resultdf
```

```
##         Comparison  Up Down totalDE
## 1      PNA.10b_RAW  38   86     124
## 2   PNA.10b_shrink  11   35      46
## 3       PNA.21_Raw 216  502     718
## 4    PNA.21_shrink  90  269     359
## 5    PNA10b.21_raw 198  526     724
## 6 PNA10b.21_shrink 102  298     400
```

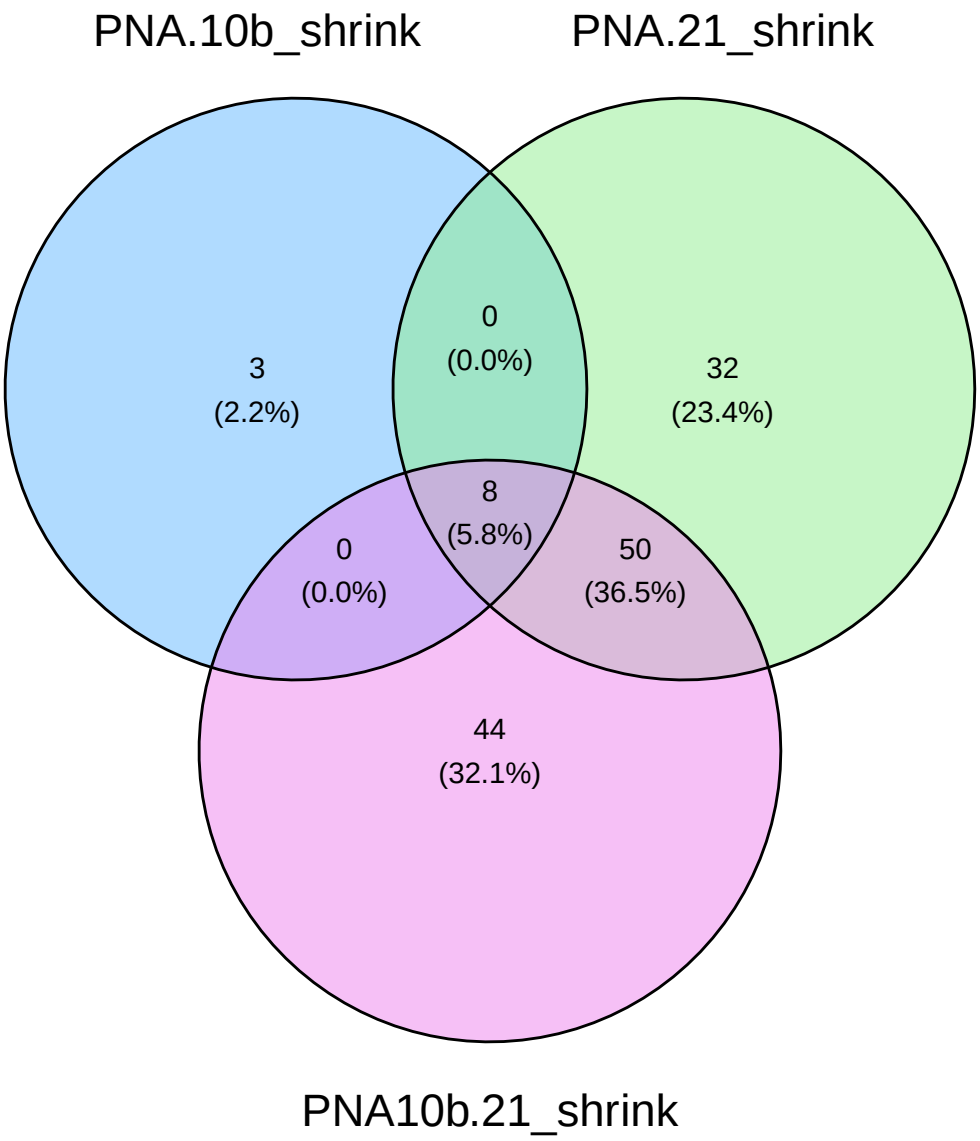
In order to get an estimate of overlap between different comparisons lets put some venn diagram together.

Venn Diagramsfor overlap of genes

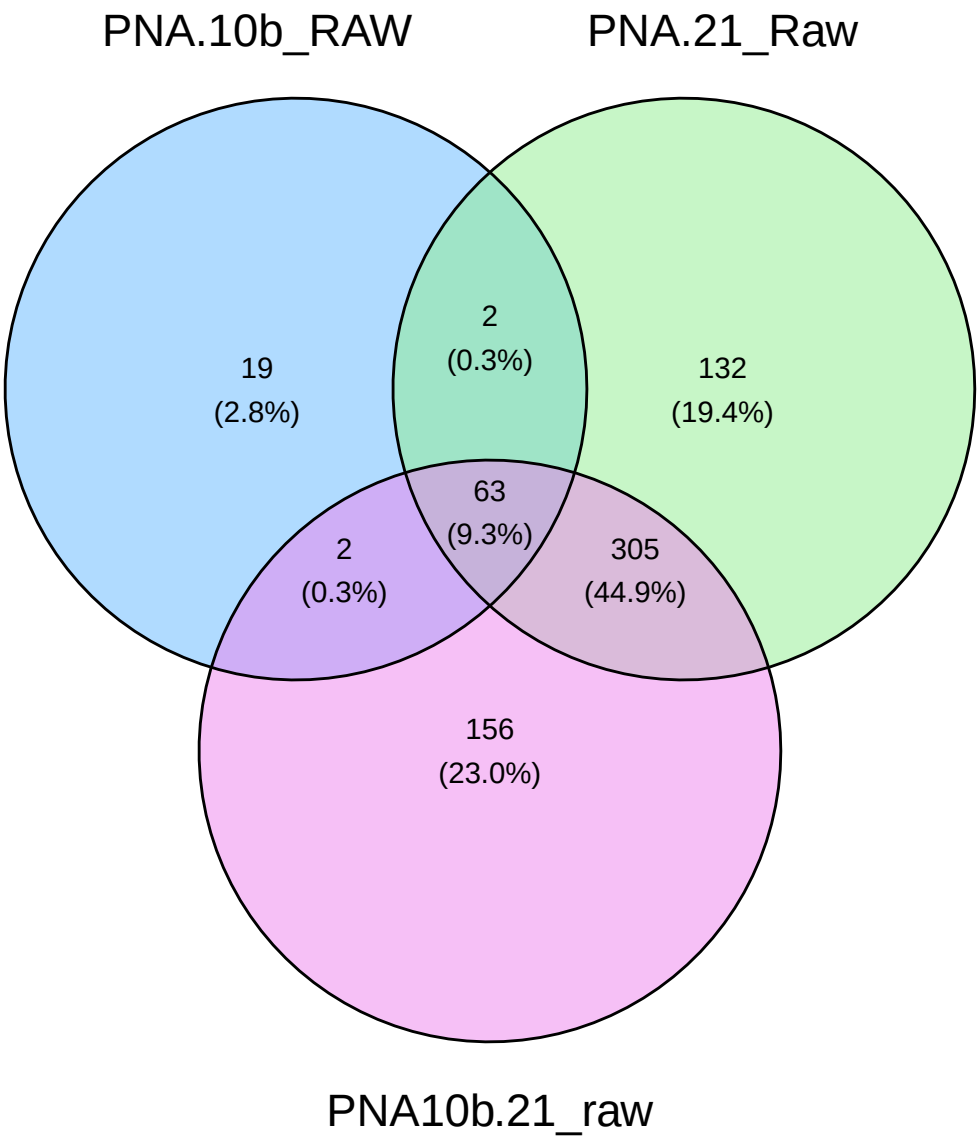
Plot : up_raw



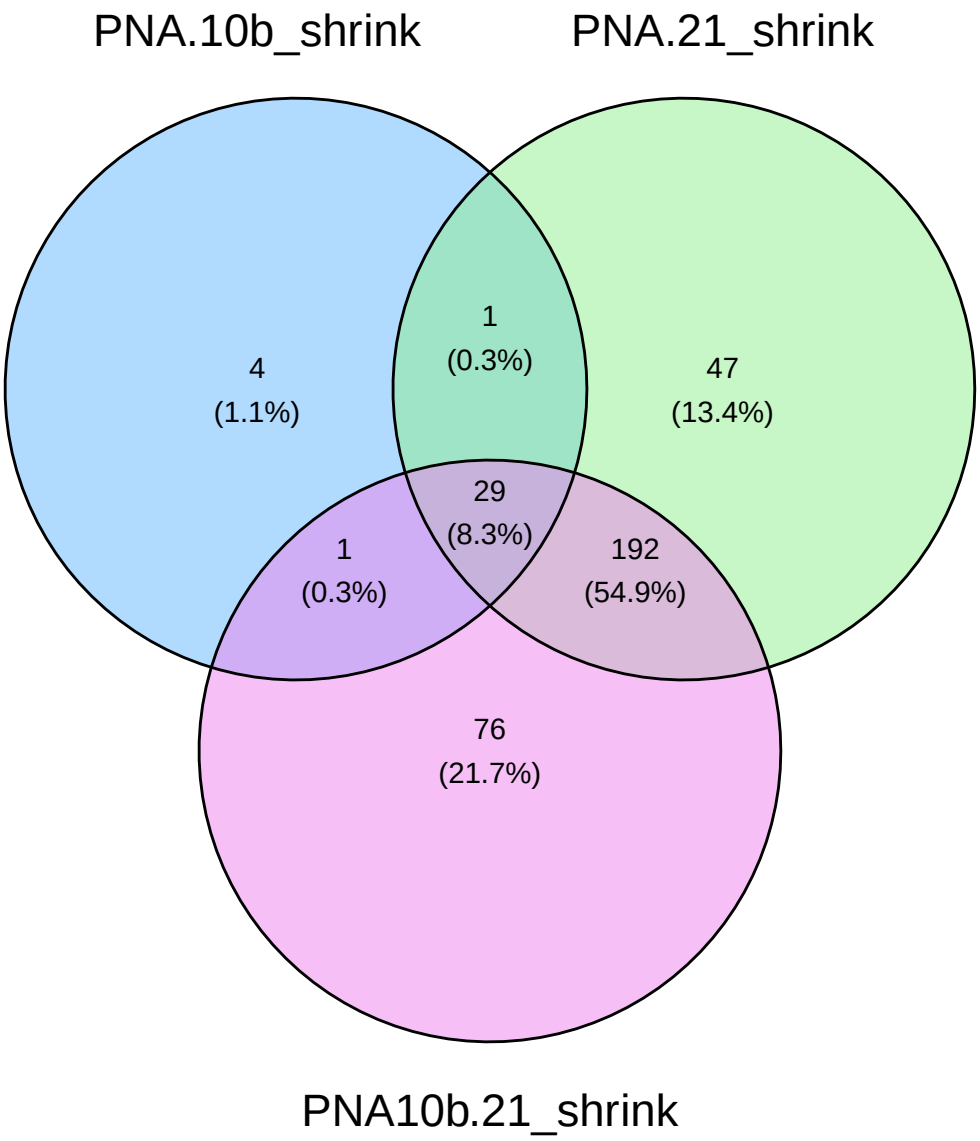
Plot : up_shrink



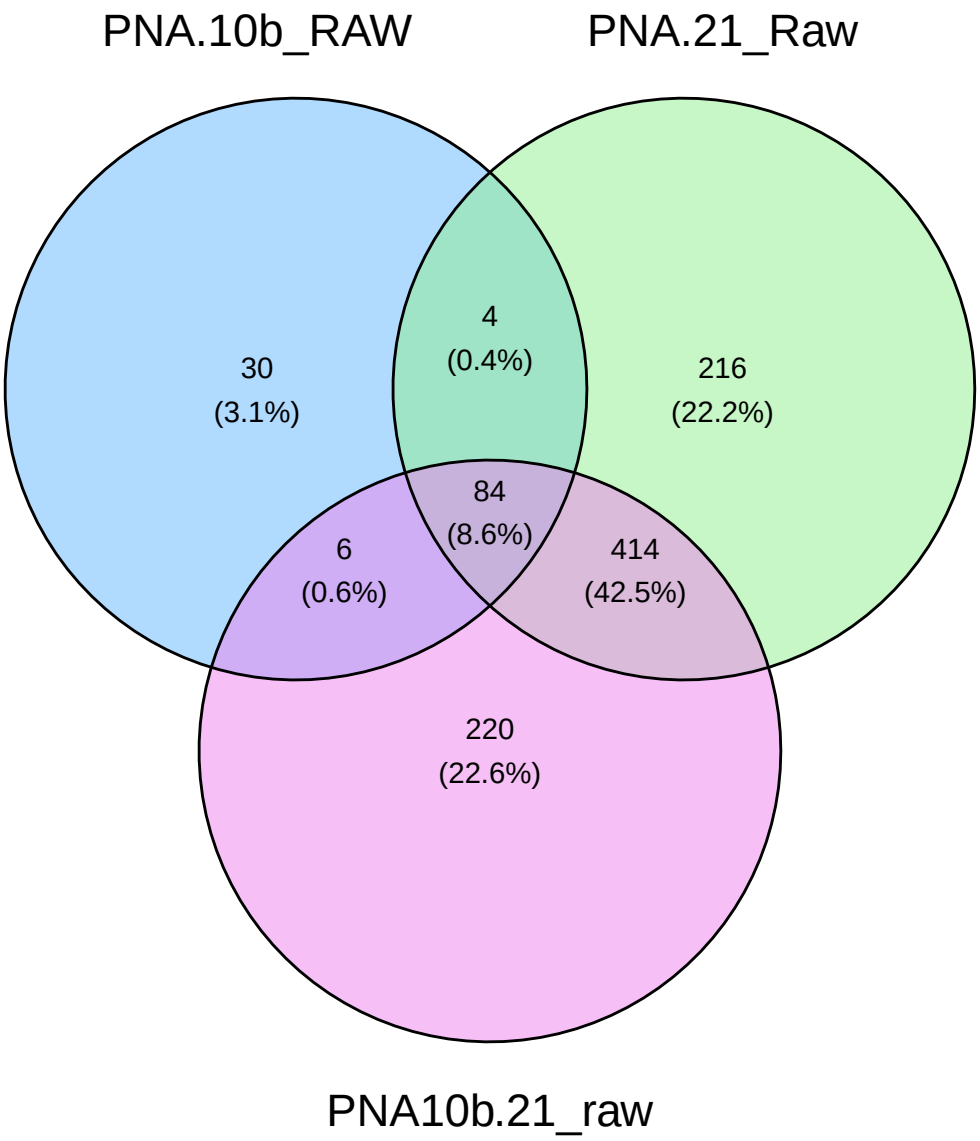
Plot : down_raw



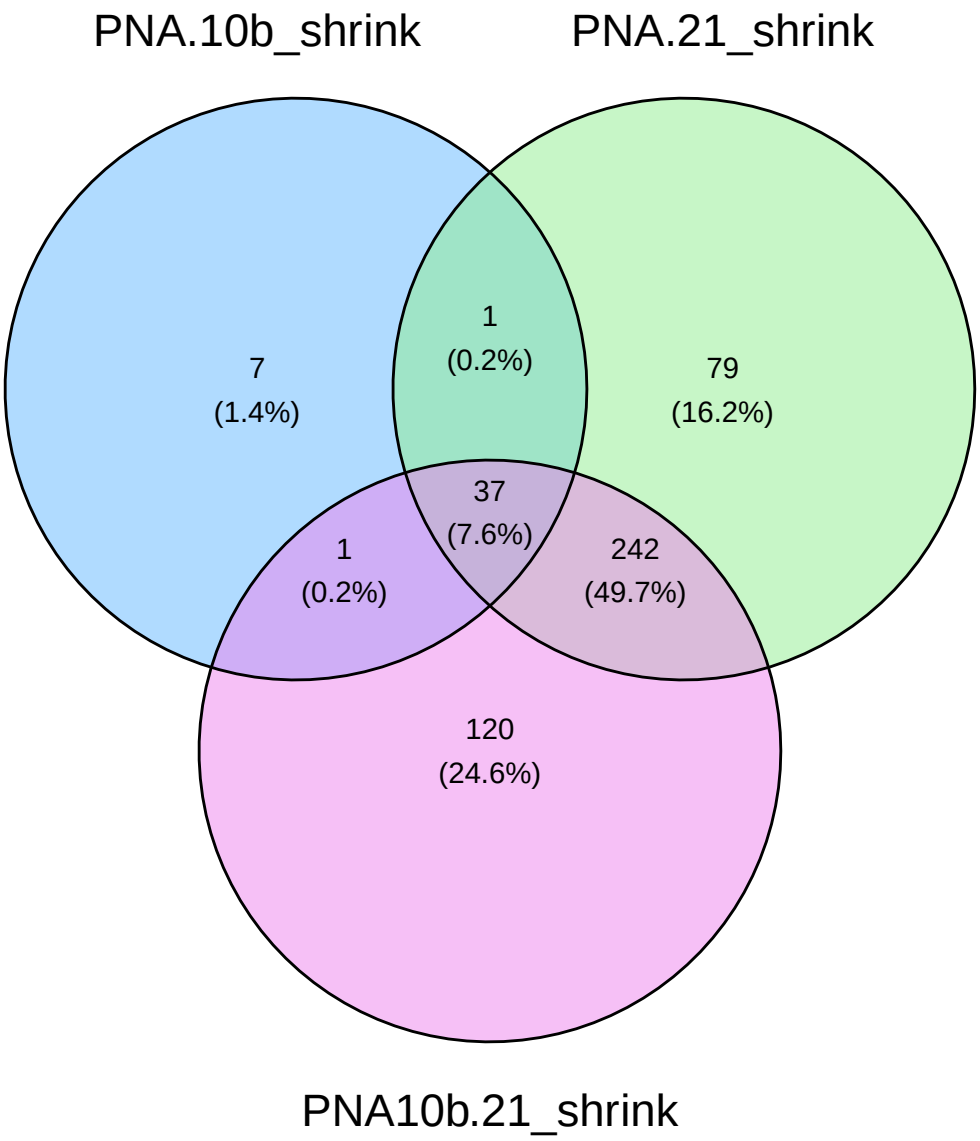
Plot : down_shrink


PNA.10b_shrink
PNA.21_shrink
4
(1.1%)
1
(0.3%)
47
(13.4%)
29
(8.3%)
1
(0.3%)
192
(54.9%)
76
(21.7%)
PNA10b.21_shrink

Plot : DE_raw



Plot : DE_shrink

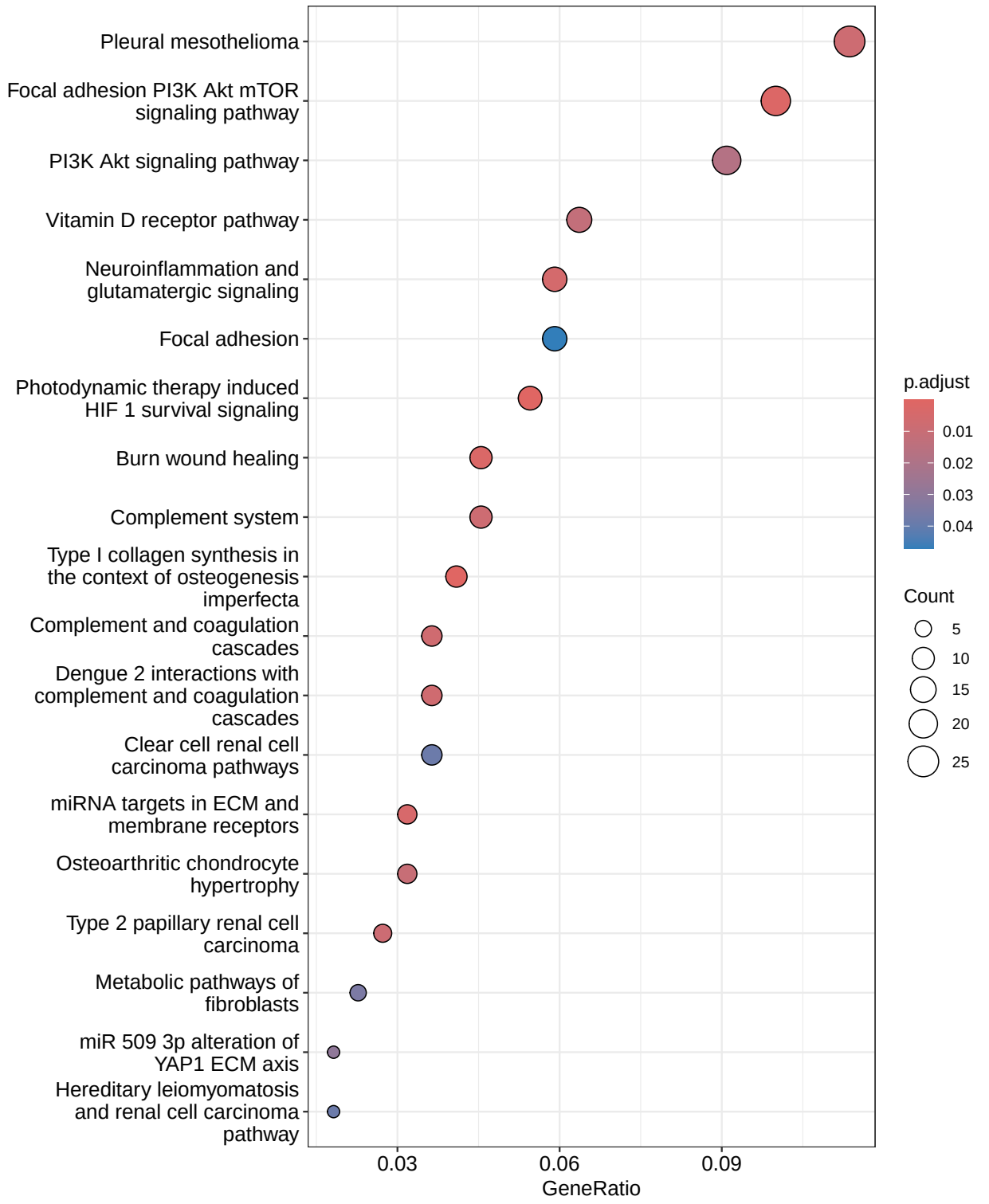


Pathway

Wiki Pathways

```
pna10b_21<-read.csv("condition_PNA10b.21_vs_control_14March2024_RawDE.csv")%>%
  column_to_rownames(., var="Row.names") %>%
  dplyr::select(., log2FoldChange,padj,gene_name,entrez)%>%
  filter(.,padj < 0.05)%>%
  arrange(desc(log2FoldChange))
colnames(pna10b_21)<-paste0("pna10b.21_",colnames(pna10b_21))
pna10b.21_rank<-pna10b_21$pna10b.21_log2FoldChange
names(pna10b.21_rank)<-as.character(pna10b_21$pna10b.21_entrez)
pna10b.21_rank_f<-pna10b.21_rank[!is.na(names(pna10b.21_rank))]
dwn_de<-names(pna10b.21_rank[pna10b.21_rank<(-0.58)])
dwn_de<-dwn_de[!is.na(dwn_de)]

wiki<-enrichWP(dwn_de, organism = "Homo sapiens")
```

```
dotplot(wiki, showCategory=30)
```




```
ggoMF <- groupGO(gene     = dwn_de,
                 OrgDb    = org.Hs.eg.db,
```

```
                ont      = "MF",
                level    = 2,
                readable = TRUE) %>%
  arrange(desc(Count))
head(ggoMF[,c(1:4)],20)
```

```
                ID                                                  Description Count
```

GO:0005488 GO:0005488 binding 354 GO:0003824 GO:0003824 catalytic activity 105 GO:0098772 GO:0098772 molecular function regulator activity 58 GO:0060089 GO:0060089 molecular transducer activity 40 GO:0005215 GO:0005215 transporter activity 38 GO:0005198 GO:0005198 structural molecule activity 32 GO:0140110 GO:0140110 transcription regulator activity 24 GO:0060090 GO:0060090 molecular adaptor activity 10 GO:0038024 GO:0038024 cargo receptor activity 6 GO:0140657 GO:0140657 ATP-dependent activity 6 GO:0045182 GO:0045182 translation regulator activity 3 GO:0003774 GO:0003774 cytoskeletal motor activity 2 GO:0016209 GO:0016209 antioxidant activity 2 GO:0030291 GO:0030291 protein serine/threonine kinase inhibitor activity 1 GO:0044183 GO:0044183 protein folding chaperone 1 GO:0140104 GO:0140104 molecular carrier activity 1 GO:0140223 GO:0140223 general transcription initiation factor activity 1 GO:0140313 GO:0140313 molecular sequestering activity 1 GO:0008603 GO:0008603 cAMP-dependent protein kinase regulator activity 0 GO:0010858 GO:0010858 calcium-dependent protein kinase regulator activity 0 GeneRatio GO:0005488 354/468 GO:0003824 105/468 GO:0098772 58/468 GO:0060089 40/468 GO:0005215 38/468 GO:0005198 32/468 GO:0140110 24/468 GO:0060090 10/468 GO:0038024 6/468 GO:0140657 6/468 GO:0045182 3/468 GO:0003774 2/468 GO:0016209 2/468 GO:0030291 1/468 GO:0044183 1/468 GO:0140104 1/468 GO:0140223 1/468 GO:0140313 1/468 GO:0008603 0/468 GO:0010858 0/468

```
ggoBP <- groupGO(gene     = dwn_de,
                OrgDb    = org.Hs.eg.db,
                ont      = "BP",
                level    = 6,
                readable = TRUE) %>%
  arrange(desc(Count))

head(ggoBP[,c(1:4)],20)
```

```
                ID
```

GO:0007166 GO:0007166 GO:0010468 GO:0010468 GO:0036211 GO:0036211 GO:0009966 GO:0009966 GO:0007399 GO:0007399 GO:0035556 GO:0035556 GO:0031326 GO:0031326 GO:0010604 GO:0010604 GO:0090304 GO:0090304 GO:0019219 GO:0019219 GO:0016070 GO:0016070 GO:0010605 GO:0010605 GO:0010556 GO:0010556 GO:0051252 GO:0051252 GO:0072359 GO:0072359 GO:0034654 GO:0034654 GO:0022008 GO:0022008 GO:0051246 GO:0051246 GO:0016310 GO:0016310 GO:0120036 GO:0120036 Description Count GO:0007166 cell surface receptor signaling pathway 102 GO:0010468 regulation of gene expression 97 GO:0036211 protein modification process 92 GO:0009966 regulation of signal transduction 89 GO:0007399 nervous system development 88 GO:0035556 intracellular signal transduction 75 GO:0031326 regulation of cellular biosynthetic process 70 GO:0010604 positive regulation of macromolecule metabolic process 68 GO:0090304 nucleic acid metabolic process 67 GO:0019219 regulation of nucleobase-containing compound metabolic process 66 GO:0016070 RNA metabolic process 65 GO:0010605 negative regulation of macromolecule metabolic process 64 GO:0010556 regulation of macromolecule biosynthetic process 63 GO:0051252 regulation of RNA metabolic process 62 GO:0072359 circulatory system development 62 GO:0034654 nucleobase-containing compound biosynthetic process 61 GO:0022008 neurogenesis 60 GO:0051246 regulation of protein metabolic process 59 GO:0016310 phosphorylation 56 GO:0120036 plasma membrane bounded cell projection organization 56 GeneRatio GO:0007166 102/468 GO:0010468 97/468 GO:0036211 92/468 GO:0009966 89/468 GO:0007399 88/468 GO:0035556 75/468 GO:0031326 70/468 GO:0010604 68/468 GO:0090304 67/468 GO:0019219 66/468 GO:0016070 65/468 GO:0010605 64/468 GO:0010556 63/468 GO:0051252 62/468 GO:0072359 62/468 GO:0034654 61/468 GO:0022008 60/468 GO:0051246 59/468 GO:0016310 56/468 GO:0120036 56/468

Survival

```
library("survival")
library("ggsurvfit")
library("gtsummary")
```

```
##
## Attaching package: 'gtsummary'
```

```
## The following object is masked from 'package:AnnotationDbi':
##
##     select
```

```
miRexp<-fread("./gdac.broadinstitute.org_GBMLGG.Merge_mirnaseq__illuminahiseq_mirnaseq__bcgsc_ca__Level

clinout<-fread("./gdac.broadinstitute.org_GBMLGG.Clinical_Pick_Tier1.Level_4.2016012800.0.0/GBMLGG.clin

head(miRexp)
```

Hybridization REF mir-10b mir-21 Status 1: TCGA-HT-8563 27498.85 404210.3 HH 2: TCGA-HT-7882 28863.98 287880.7 HH 3: TCGA-DU-7007 58068.19 272269.1 HH 4: TCGA-S9-A6WN 73185.11 203725.3 HH 5: TCGA-DU-A76L 61145.90 168169.3 HH 6: TCGA-HT-8106 41104.53 155760.2 HH

```
head(clinout)
```

Hybridization_REF years_to_birth vital_status days_to_death 1: TCGA-02-0001 44 1 358 2: TCGA-02-0003 50 1 144 3: TCGA-02-0004 59 1 345 4: TCGA-02-0006 56 1 558 5: TCGA-02-0007 40 1 705 6: TCGA-02-0009 61 1 322 days_to_last_followup tumor_tissue_site gender 1: NA brain female 2: NA brain male 3: NA brain male 4: NA brain female 5: NA brain female 6: NA brain female date_of_initial_pathologic_diagnosis days_to_last_known_alive 1: 2002 NA 2: 2003 NA 3: 2002 NA 4: 2002 NA 5: 2002 NA 6: 2003 NA radiation_therapy karnofsky_performance_score 1: yes 80 2: yes 100 3: yes 80 4: yes 80 5: yes 80 6: yes 80 histological_type race ethnicity 1: untreated primary (de novo) gbm white not hispanic or latino 2: untreated primary (de novo) gbm white not hispanic or latino 3: untreated primary (de novo) gbm white not hispanic or latino 4: untreated primary (de novo) gbm white not hispanic or latino 5: treated primary gbm white not hispanic or latino 6: untreated primary (de novo) gbm white not hispanic or latino

```
df<-merge(miRexp,clinout,by.x="Hybridization REF", by.y="Hybridization_REF",all.x=T, sort=F)

head(df)
```

Hybridization REF mir-10b mir-21 Status years_to_birth vital_status 1: TCGA-HT-8563 27498.85 404210.3 HH 30 0 2: TCGA-HT-7882 28863.98 287880.7 HH 66 1 3: TCGA-DU-7007 58068.19 272269.1 HH 33 1 4: TCGA-S9-A6WN 73185.11 203725.3 HH 38 0 5: TCGA-DU-A76L 61145.90 168169.3 HH 54 1 6: TCGA-HT-8106 41104.53 155760.2 HH 53 0 days_to_death days_to_last_followup tumor_tissue_site gender 1: NA 860 central nervous system female 2: 113 NA central nervous system male 3: 1915 NA central nervous system male 4: NA 805 central nervous system female 5: 814 NA central nervous system male 6: NA 3 central nervous system male date_of_initial_pathologic_diagnosis days_to_last_known_alive 1: 2011 NA 2: 2010 NA 3: 1997 NA 4: 2013 NA 5: 2000 NA 6: 2010 NA radiation_therapy karnofsky_performance_score histological_type race 1: yes NA astrocytoma white 2: no NA oligodendroglioma white 3: NA astrocytoma white 4: yes 90 astrocytoma white 5: yes NA oligodendroglioma white 6: no NA astrocytoma white ethnicity 1: not hispanic or latino 2: hispanic or latino 3: not hispanic or latino 4: not hispanic or latino 5: not hispanic or latino 6: not hispanic or latino

```
df$status<-0
df<-df[,c("Hybridization REF","Status","days_to_death","status")]
colnames(df)<-c("patID","exprs","time","status")
```

```
df<-df[complete.cases(df$time)]

head(df)
```

```
      patID  exprs  time status
     <char> <char> <int>  <num>
```

1: TCGA-HT-7882 HH 113 0 2: TCGA-DU-7007 HH 1915 0 3: TCGA-DU-A76L HH 814 0 4: TCGA-DU-7012 HH 199 0 5: TCGA-FG-5963 HH 775 0 6: TCGA-HT-A61C HH 537 0

```
df$expCoded<-0
df$expCoded[df$exprs=="HH"]<-1
coxph(Surv(time, status) ~ exprs, data = df)
```

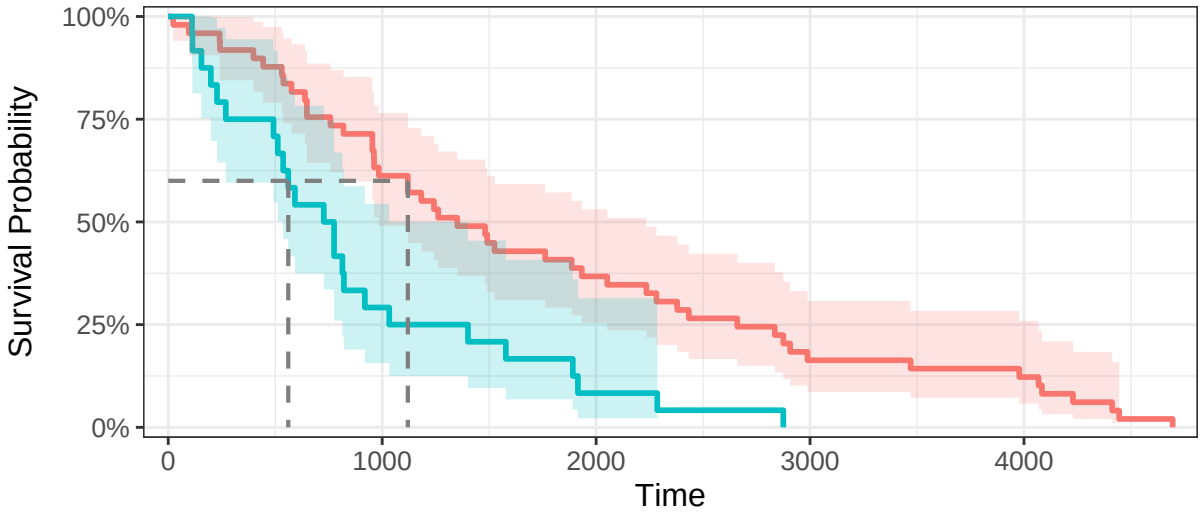
Call: coxph(formula = Surv(time, status) ~ exprs, data = df)

```
    coef exp(coef) se(coef)  z  p
```

exprsLL NA NA 0 NA NA

Likelihood ratio test=0 on 0 df, p=1 n= 73, number of events= 0

```
df<-df[complete.cases(df$patID)]

p <- survfit2(Surv(time) ~ expCoded, data = df) |>
  ggsurvfit(linewidth = 1) +
  add_confidence_interval() +
  add_risktable() +
  add_quantile(y_value = 0.6, color = "gray50", linewidth = 0.75) +
  scale_ggsurvfit()

p
```



0					
At Risk	49	30	18	8	6
Events	0	19	31	41	43
1					
At Risk	24	7	2	0	0
Events	0	17	22	24	24