

# R Notebook for linLAB R scripts

## Contents

Part I: Refined pictures used for publishing articles . . . . .	1
Part II: Other refined pictures . . . . .	61

### Part I: Refined pictures used for publishing articles

#### 1. Horizontal Box Plot

```
# Load required package
if (!require("ggplot2")) install.packages("ggplot2")

## Loading required package: ggplot2

library(ggplot2)
if (!require("pheatmap")) install.packages("pheatmap")

## Loading required package: pheatmap

library(pheatmap)
if (!require("gplots")) install.packages("gplots")

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##      lowess

library(gplots)
if (!require("vegan")) install.packages("vegan")

## Loading required package: vegan

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.6-2
```

```

library(vegan)
if (!require("permute")) install.packages("permute")
library(permute)
if (!require("lattice")) install.packages("lattice")
library(lattice)
if (!require("limma")) install.packages("limma")

## Loading required package: limma

library(limma)
if (!require("edgeR")) install.packages("edgeR")

## Loading required package: edgeR

library(edgeR)
if (!require("ggdendro")) BiocManager::install("ggdendro", version = "3.8")

## Loading required package: ggdendro

library("ggdendro")
#BiocManager::install("ggdendro", version = "3.8")
if (!require("cowplot")) install.packages("cowplot")

## Loading required package: cowplot

library("cowplot")
if (!require("biomaRt")) install.packages("biomaRt")

## Loading required package: biomaRt

library('biomaRt')
if (!require("curl")) install.packages("curl")

## Loading required package: curl

## Using libcurl 7.64.1 with Schannel

library("curl")
if (!require("tidyverse")) install.packages("tidyverse")

## Loading required package: tidyverse

## -- Attaching packages ----- tidyverse 1.3.2 --
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyverse 1.2.0    v stringr 1.4.0
## v readr    2.1.3    vforcats 0.5.2
## v purrr   0.3.4
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x readr::parse_date() masks curl::parse_date()
## x dplyr::select()    masks biomaRt::select()

```

```

library("tidyverse")
if (!require("ggpubr")) install.packages("ggpubr")

## Loading required package: ggpublisher
##
## Attaching package: 'ggpubr'
##
## The following object is masked from 'package:cowplot':
##
##     get_legend

#install.packages("ggplot2")
#install.packages("ggpubr")
library("ggpubr")
if (!require("dplyr")) install.packages("dplyr")
library(dplyr)
if (!require("forcats")) install.packages("forcats")
library(forcats)
all_matrix<-read.csv("D:\\Linlab\\_article\\figure1\\box_plot\\integrate_all.csv",row.names = 1)
head(all_matrix)

##          iL0a_readCount iL0a_MAP iL0b_readCount iL0b_MAP iL24b_readCount
## Eef1a1      973227.7   0.0402      1346952.5   0.0337      1238906.0
## Hsp90ab1    576872.9   0.0595      738180.9   0.0510      770930.8
## Hspa8       462245.7   0.1265      563471.1   0.1032      447568.8
## Pabpc1      418850.7   0.0529      455192.5   0.0419      547382.8
## Hsp90aa1    399728.1   0.0548      710423.2   0.0453      585063.2
## Actb        379285.9   0.0480      460363.3   0.0406      479406.5
##          iL24b_MAP iL48a_readCount iL48a_MAP iL48b_readCount iL48b_MAP
## Eef1a1      0.0230      1243348.8   0.0220      1068701.8   0.0135
## Hsp90ab1    0.0381      776439.2   0.0380      571242.1   0.0248
## Hspa8       0.0841      504194.7   0.0752      356730.2   0.0561
## Pabpc1      0.0247      504968.0   0.0211      378381.7   0.0149
## Hsp90aa1    0.0281      582854.4   0.0253      435721.3   0.0175
## Actb        0.0264      486223.7   0.0212      352977.3   0.0151
##          iL72a_readCount iL72a_MAP iL72b_readCount iL72b_MAP L0.MAP
## Eef1a1      1191805.5   0.0155      506483.23   0.0052 0.0122
## Hsp90ab1    783644.8   0.0299      226402.17   0.0153 0.0260
## Hspa8       516217.7   0.0502      170805.85   0.0224 0.0460
## Pabpc1      483456.8   0.0161      192789.42   0.0063 0.0253
## Hsp90aa1    629412.3   0.0170      202184.00   0.0098 0.0120
## Actb        461170.7   0.0181      99702.33    0.0097 0.0250
##          L0.Readcount L2i0a_readCount L2i0a_MAP L2i0b_readCount L2i0b_MAP
## Eef1a1      139855.08   1435716.4   0.0174      788685.9   0.0113
## Hsp90ab1    67153.33    747203.9   0.0383      422197.7   0.0305
## Hspa8       49137.08    564287.8   0.0653      335355.3   0.0404
## Pabpc1      32955.00    580035.8   0.0242      351260.0   0.0167
## Hsp90aa1    73686.67    625907.3   0.0248      365038.1   0.0169
## Actb        21681.67    426929.1   0.0245      232737.2   0.0155
##          L2i24b_readCount L2i24b_MAP L2i48a_readCount L2i48a_MAP
## Eef1a1      993356.7    0.0408      982305.3   0.0261
## Hsp90ab1    627668.2    0.0601      632177.3   0.0407

```

## Hspa8	544062.6	0.1218	582197.2	0.0857	
## Pabpc1	426941.0	0.0582	485269.5	0.0478	
## Hsp90aa1	496973.0	0.0529	507883.7	0.0408	
## Actb	370840.8	0.0696	373360.5	0.0400	
## L2i48b_readCount	L2i48b_MAP	L2i72a_readCount	L2i72a_MAP		
## Eef1a1	1127538.3	0.0287	1101597.8	0.0472	
## Hsp90ab1	718571.1	0.0465	654560.7	0.0769	
## Hspa8	646244.0	0.0886	539308.6	0.1312	
## Pabpc1	507711.5	0.0561	513249.8	0.0599	
## Hsp90aa1	588192.1	0.0435	593056.2	0.0738	
## Actb	461989.5	0.0432	425849.0	0.0540	
## L2i72b_readCount	L2i72b_MAP	X2i0_0817Readcount	NP24_0817Readcount		
## Eef1a1	1090786.2	0.0529	510276.0	384944.7	
## Hsp90ab1	679714.2	0.0813	354306.3	355823.2	
## Hspa8	570791.0	0.1412	255890.3	196796.9	
## Pabpc1	565586.6	0.0641	211858.2	210768.3	
## Hsp90aa1	626948.0	0.0782	421220.8	274980.6	
## Actb	419838.6	0.0631	249014.8	285594.8	
## NP72_0817Readcount	X2i0_0817MAP	NP24_0817MAP	NP72_0817MAP		
## Eef1a1	570233.6	0.0478	0.0557	0.0443	
## Hsp90ab1	465743.2	0.0710	0.0831	0.0710	
## Hspa8	297058.3	0.1405	0.1835	0.1461	
## Pabpc1	262656.0	0.0574	0.0601	0.0582	
## Hsp90aa1	354498.6	0.0734	0.0664	0.0631	
## Actb	381937.5	0.0474	0.0810	0.0718	
## NP0_0824Readcount	NP24_0824Readcount	NP48_0824Readcount			
## Eef1a1	620150.8	636849.3	541909.0		
## Hsp90ab1	314982.8	402573.7	416009.3		
## Hspa8	194518.9	265060.0	271726.3		
## Pabpc1	227876.2	175319.0	176660.7		
## Hsp90aa1	398497.4	323563.8	305947.2		
## Actb	195479.3	203595.8	202189.0		
## NP72_0824Readcount	NP0_0824MAP	NP24_0824MAP	NP48_0824MAP	NP72_0824MAP	
## Eef1a1	542459.0	0.0186	0.0290	0.0320	0.0361
## Hsp90ab1	397413.5	0.0249	0.0509	0.0552	0.0582
## Hspa8	256895.2	0.0672	0.1042	0.1162	0.1280
## Pabpc1	153543.7	0.0242	0.0365	0.0417	0.0510
## Hsp90aa1	258698.9	0.0280	0.0356	0.0379	0.0415
## Actb	221957.7	0.0275	0.1174	0.1927	0.2182
## NP0_1106Readcount	NP24b_1106Readcount	NP48a_1106Readcount			
## Eef1a1	380947.4	869247.7	1010127.1		
## Hsp90ab1	219993.3	552574.8	644568.8		
## Hspa8	165577.8	466033.0	455134.9		
## Pabpc1	171787.3	284794.8	305042.7		
## Hsp90aa1	202897.4	361886.1	387671.9		
## Actb	128666.3	313999.2	316902.0		
## NP6a_1106Readcount	NP72b_1106Readcount	NP0_1106MAP	NP24b_1106MAP		
## Eef1a1	656369.4	893236.7	0.0317	0.0374	
## Hsp90ab1	362028.8	614958.8	0.0454	0.0673	
## Hspa8	201809.9	438242.2	0.1007	0.1169	
## Pabpc1	276587.1	320842.0	0.0420	0.0559	
## Hsp90aa1	228948.1	332845.7	0.0452	0.0537	
## Actb	350976.7	364155.0	0.0436	0.0665	
## NP48a_1106MAP	NP6a_1106MAP	NP72b_1106MAP			

```

## Eef1a1      0.0338      0.0323      0.0512
## Hsp90ab1    0.0548      0.0551      0.0793
## Hspa8       0.1171      0.1306      0.1607
## Pabpc1      0.0491      0.0416      0.0827
## Hsp90aa1    0.0469      0.0497      0.0706
## Actb        0.0646      0.0361      0.0858

all_matrix_selected<-as.data.frame(all_matrix[,c("NPO_1106MAP","L2i0a_MAP","NP72b_1106MAP")])
#write.csv(all_matrix_selected,"matrix_selected_for_boxplot_DR.csv")
all_matrix_MAPcorr_matrix<-cor(all_matrix_selected,method = "spearman")
all_matrix_MAPcorr<-cor.test(all_matrix_selected$NPO_1106MAP,all_matrix_selected$L2i0a_MAP,method = "sp

## Warning in cor.test.default(all_matrix_selected$NPO_1106MAP,
## all_matrix_selected$L2i0a_MAP, : Cannot compute exact p-value with ties

box_final <- all_matrix_selected %>% gather(colnames(all_matrix_selected),
                                              key='class',value='MAP')
box_final2 <-as.data.frame(box_final[,c("class","MAP")])

####draw the box plot [final version]
box_final2$class = with(box_final2, reorder(class, MAP, median))
#Data$Group = factor(Data$Group, levels = c("Control", "Before thrombolytic therapy", "After thrombolytic

p1<-box_final2 %>%
  ggplot(aes(class, MAP)) +
  # geom_violin(aes(fill = class),trim = F) +
  #geom_boxplot(alpha=0.7,outlier.shape = NA,outlier.colour = NA) +
  stat_boxplot(geom = 'errorbar', width = 0.1) +
  geom_boxplot(aes(fill = class), width = 0.2, show.legend = F,outlier.shape = NA,outlier.colour = NA) +
  #geom_jitter(aes(shape = class)) +
  coord_trans(x = "identity", y = "identity", xlim = NULL, ylim = c(0,0.8))+ 
  stat_summary(fun = median, geom = 'point', color = 'dark grey', size = rel(2)) +
  #guides(shape = F, color = F) +
  labs(x = "Different cell line", y = "mRNA decay rate")+
  scale_fill_brewer(palette = 'Set2')+
  theme_bw() +
  theme( plot.title = element_text(hjust = 0.5),
         text = element_text(size=22),
         title=element_text(size = 30),
         axis.title.x = element_text(size=24),
         axis.title.y = element_text(size=24),
         axis.text.x = element_text(size=16),
         axis.text.y = element_text(size=22),
         axis.title = element_text(face="bold"))

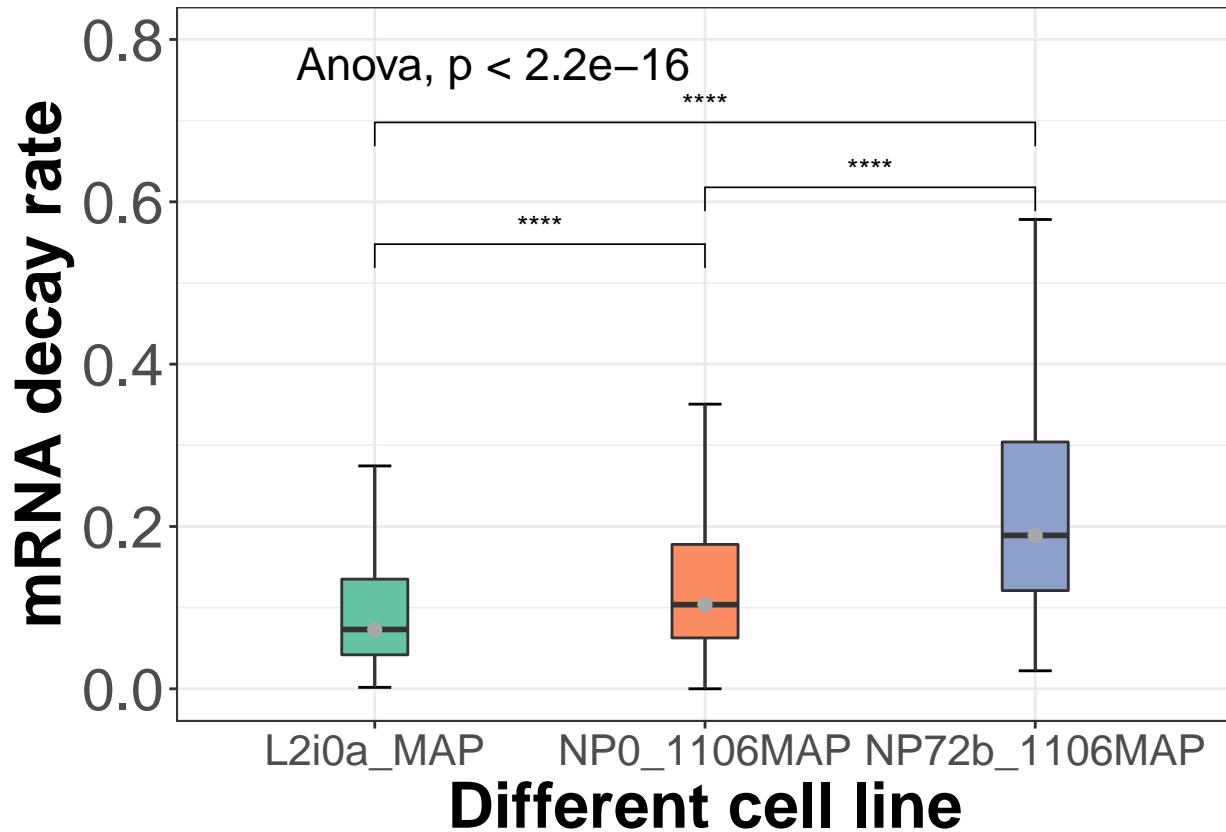
# theme_bw() +
# scale_fill_brewer(palette = 'Set2')
mycomparision <- list(c("L2i0a_MAP", "NPO_1106MAP"), c("NPO_1106MAP", "NP72b_1106MAP"), c("L2i0a_MAP",
#data: Sepal.Length by Species
#Bartlett's K-squared = 16.006, df = 2, p-value = 0.0003345, For heteroscedasticity, use non-parametric
#p1 <- p + stat_compare_means(method = 'wilcox.test', comparisons = mycomparision, label = 'p.signif',l
#p1
##without annova value:
p1_1 <- p1 + stat_compare_means(method = 't.test', comparisons = mycomparision, label = 'p.signif',labe

```

```
#p1_1
```

```
##with annova value:
```

```
p1_2 <- p1 + stat_compare_means(method = 't.test', comparisons = mycomparision, label = 'p.signif',label.y = 0.8)
```

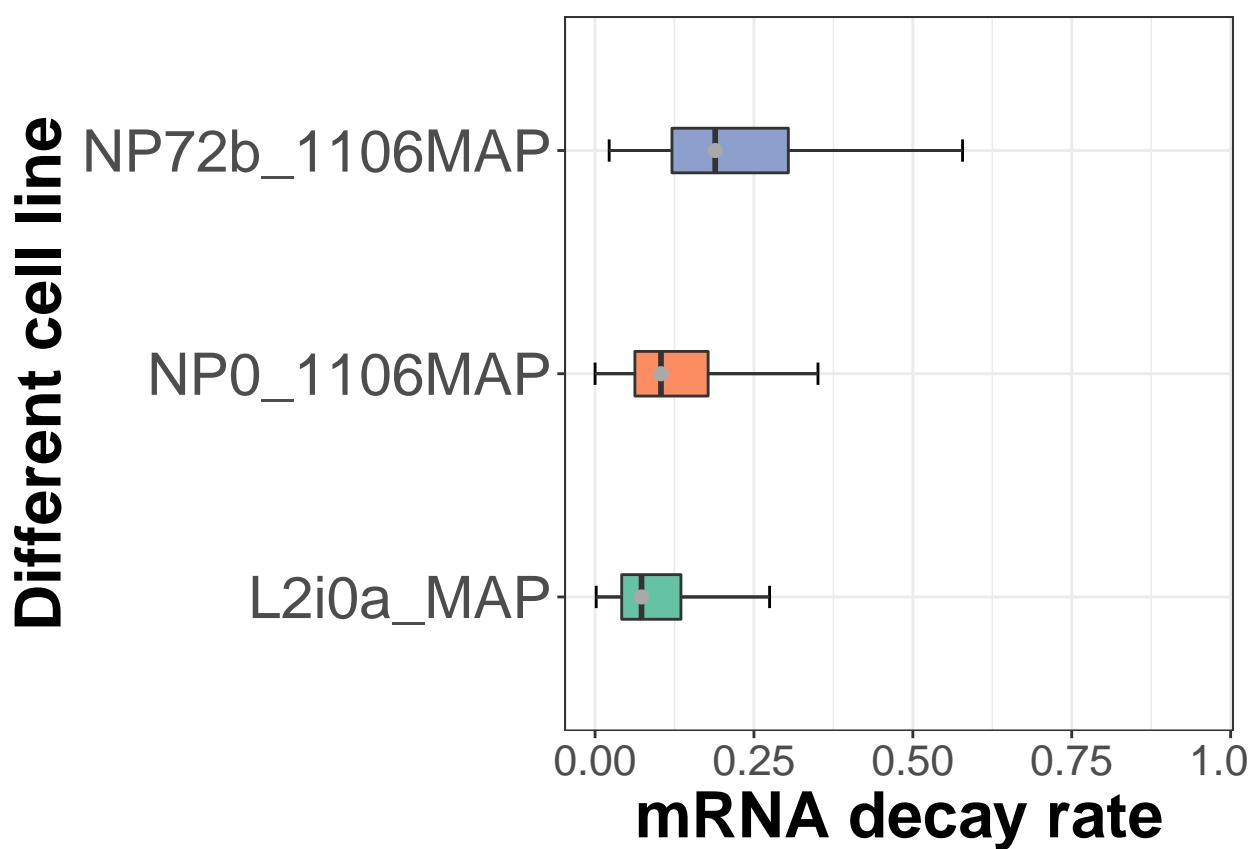


```
##with horizontal box plot:
```

```
horizontal<-p1+coord_flip()
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```

```
plot(horizontal, choix = "ind")
```



## 2. Trinity Plot

```
# Load required package
if (!require("ggplot2")) install.packages("ggplot2")
library(ggplot2)
if (!require("pheatmap")) install.packages("pheatmap")
library(pheatmap)
if (!require("gplots")) install.packages("gplots")
library(gplots)
if (!require("vegan")) install.packages("vegan")
library(vegan)
if (!require("permute")) install.packages("permute")
library(permute)
if (!require("lattice")) install.packages("lattice")
library(lattice)
if (!require("limma")) install.packages("limma")
library(limma)
if (!require("edgeR")) install.packages("edgeR")
library(edgeR)
if (!require("ggdendro")) BiocManager::install("ggdendro", version = "3.8")
library("ggdendro")
if (!require("cowplot")) install.packages("cowplot")
library("cowplot")
if (!require("biomaRt")) install.packages("biomaRt")
```

```

library('biomaRt')
if (!require("curl")) install.packages("curl")
library("curl")
if (!require("tidyverse")) install.packages("tidyverse")
library("tidyverse")
if (!require("ggpubr")) install.packages("ggpubr")
library("ggpubr")
if (!require("dplyr")) install.packages("dplyr")
library(dplyr)
if (!require("forcats")) install.packages("forcats")
library(forcats)
if (!require("ggtern")) install.packages("ggtern")

## Loading required package: ggtern

## Registered S3 methods overwritten by 'ggtern':
##   method           from
##   grid.draw.ggplot ggplot2
##   plot.ggplot     ggplot2
##   print.ggplot    ggplot2

## --
## Remember to cite, run citation(package = 'ggtern') for further info.
## --

##
## Attaching package: 'ggtern'

## The following objects are masked from 'package:ggplot2':
## 
##   aes, annotate, ggplot, ggplot_build, ggplot_gtable, ggplotGrob,
##   ggsave, layer_data, theme_bw, theme_classic, theme_dark,
##   theme_gray, theme_light, theme_linedraw, theme_minimal, theme_void

library(ggtern)
if (!require("viridis")) install.packages("viridis")

## Loading required package: viridis

## Loading required package: viridisLite

library(viridis)
if (!require("RColorBrewer")) install.packages("RColorBrewer")

## Loading required package: RColorBrewer

library(RColorBrewer)
all_matrix<-read.csv("D:\\Linlab\\_article\\figure1\\box_plot\\integrate_all.csv",row.names = 1)
head(all_matrix)

```

```

##          iL0a_readCount iL0a_MAP iL0b_readCount iL0b_MAP iL24b_readCount
## Eef1a1      973227.7   0.0402    1346952.5   0.0337    1238906.0
## Hsp90ab1     576872.9   0.0595    738180.9   0.0510    770930.8
## Hspa8        462245.7   0.1265    563471.1   0.1032    447568.8
## Pabpc1       418850.7   0.0529    455192.5   0.0419    547382.8
## Hsp90aa1     399728.1   0.0548    710423.2   0.0453    585063.2
## Actb         379285.9   0.0480    460363.3   0.0406    479406.5
##          iL24b_MAP iL48a_readCount iL48a_MAP iL48b_readCount iL48b_MAP
## Eef1a1       0.0230    1243348.8   0.0220    1068701.8   0.0135
## Hsp90ab1     0.0381    776439.2    0.0380    571242.1    0.0248
## Hspa8         0.0841    504194.7    0.0752    356730.2    0.0561
## Pabpc1       0.0247    504968.0    0.0211    378381.7    0.0149
## Hsp90aa1     0.0281    582854.4    0.0253    435721.3    0.0175
## Actb         0.0264    486223.7    0.0212    352977.3    0.0151
##          iL72a_readCount iL72a_MAP iL72b_readCount iL72b_MAP L0.MAP
## Eef1a1       1191805.5   0.0155    506483.23   0.0052    0.0122
## Hsp90ab1     783644.8    0.0299    226402.17   0.0153    0.0260
## Hspa8         516217.7   0.0502    170805.85   0.0224    0.0460
## Pabpc1       483456.8    0.0161    192789.42   0.0063    0.0253
## Hsp90aa1     629412.3    0.0170    202184.00   0.0098    0.0120
## Actb         461170.7    0.0181    99702.33    0.0097    0.0250
##          L0.Readcount L2i0a_readCount L2i0a_MAP L2i0b_readCount L2i0b_MAP
## Eef1a1       139855.08   1435716.4   0.0174    788685.9    0.0113
## Hsp90ab1     67153.33    747203.9   0.0383    422197.7    0.0305
## Hspa8         49137.08   564287.8   0.0653    335355.3    0.0404
## Pabpc1       32955.00    580035.8   0.0242    351260.0    0.0167
## Hsp90aa1     73686.67    625907.3   0.0248    365038.1    0.0169
## Actb         21681.67    426929.1   0.0245    232737.2    0.0155
##          L2i24b_readCount L2i24b_MAP L2i48a_readCount L2i48a_MAP
## Eef1a1       993356.7    0.0408    982305.3    0.0261
## Hsp90ab1     627668.2    0.0601    632177.3    0.0407
## Hspa8         544062.6   0.1218    582197.2    0.0857
## Pabpc1       426941.0    0.0582    485269.5    0.0478
## Hsp90aa1     496973.0    0.0529    507883.7    0.0408
## Actb         370840.8    0.0696    373360.5    0.0400
##          L2i48b_readCount L2i48b_MAP L2i72a_readCount L2i72a_MAP
## Eef1a1       1127538.3   0.0287    1101597.8   0.0472
## Hsp90ab1     718571.1    0.0465    654560.7    0.0769
## Hspa8         646244.0   0.0886    539308.6    0.1312
## Pabpc1       507711.5    0.0561    513249.8    0.0599
## Hsp90aa1     588192.1    0.0435    593056.2    0.0738
## Actb         461989.5    0.0432    425849.0    0.0540
##          L2i72b_readCount L2i72b_MAP X2i0_0817Readcount NP24_0817Readcount
## Eef1a1       1090786.2   0.0529    510276.0    384944.7
## Hsp90ab1     679714.2    0.0813    354306.3    355823.2
## Hspa8         570791.0   0.1412    255890.3    196796.9
## Pabpc1       565586.6    0.0641    211858.2    210768.3
## Hsp90aa1     626948.0    0.0782    421220.8    274980.6
## Actb         419838.6    0.0631    249014.8    285594.8
##          NP72_0817Readcount X2i0_0817MAP NP24_0817MAP NP72_0817MAP
## Eef1a1       570233.6    0.0478    0.0557    0.0443
## Hsp90ab1     465743.2    0.0710    0.0831    0.0710
## Hspa8         297058.3   0.1405    0.1835    0.1461
## Pabpc1       262656.0    0.0574    0.0601    0.0582

```

```

## Hsp90aa1      354498.6      0.0734      0.0664      0.0631
## Actb         381937.5      0.0474      0.0810      0.0718
##          NP0_0824Readcount NP24_0824Readcount NP48_0824Readcount
## Eef1a1       620150.8      636849.3      541909.0
## Hsp90ab1     314982.8      402573.7      416009.3
## Hspa8        194518.9      265060.0      271726.3
## Pabpc1       227876.2      175319.0      176660.7
## Hsp90aa1     398497.4      323563.8      305947.2
## Actb         195479.3      203595.8      202189.0
##          NP72_0824Readcount NP0_0824MAP NP24_0824MAP NP48_0824MAP NP72_0824MAP
## Eef1a1       542459.0      0.0186      0.0290      0.0320      0.0361
## Hsp90ab1     397413.5      0.0249      0.0509      0.0552      0.0582
## Hspa8        256895.2      0.0672      0.1042      0.1162      0.1280
## Pabpc1       153543.7      0.0242      0.0365      0.0417      0.0510
## Hsp90aa1     258698.9      0.0280      0.0356      0.0379      0.0415
## Actb         221957.7      0.0275      0.1174      0.1927      0.2182
##          NP0_1106Readcount NP24b_1106Readcount NP48a_1106Readcount
## Eef1a1       380947.4      869247.7      1010127.1
## Hsp90ab1     219993.3      552574.8      644568.8
## Hspa8        165577.8      466033.0      455134.9
## Pabpc1       171787.3      284794.8      305042.7
## Hsp90aa1     202897.4      361886.1      387671.9
## Actb         128666.3      313999.2      316902.0
##          NP6a_1106Readcount NP72b_1106Readcount NP0_1106MAP NP24b_1106MAP
## Eef1a1       656369.4      893236.7      0.0317      0.0374
## Hsp90ab1     362028.8      614958.8      0.0454      0.0673
## Hspa8        201809.9      438242.2      0.1007      0.1169
## Pabpc1       276587.1      320842.0      0.0420      0.0559
## Hsp90aa1     228948.1      332845.7      0.0452      0.0537
## Actb         350976.7      364155.0      0.0436      0.0665
##          NP48a_1106MAP NP6a_1106MAP NP72b_1106MAP
## Eef1a1       0.0338      0.0323      0.0512
## Hsp90ab1     0.0548      0.0551      0.0793
## Hspa8        0.1171      0.1306      0.1607
## Pabpc1       0.0491      0.0416      0.0827
## Hsp90aa1     0.0469      0.0497      0.0706
## Actb         0.0646      0.0361      0.0858

####select three group for final bar_plot
all_matrix_selected<-as.data.frame(all_matrix[,c("NP0_1106MAP","L2i0a_MAP","NP72b_1106MAP")])
#write.csv(all_matrix_selected,"matrix_selected_for_boxplot_DR.csv")
colnames(all_matrix_selected)<-c("ground","naive","formative")
colormap<-rev(brewer.pal(20,'Spectral'))
```

## Warning in brewer.pal(20, "Spectral"): n too large, allowed maximum for palette Spectral is 11  
## Returning the palette you asked for with that many colors

```

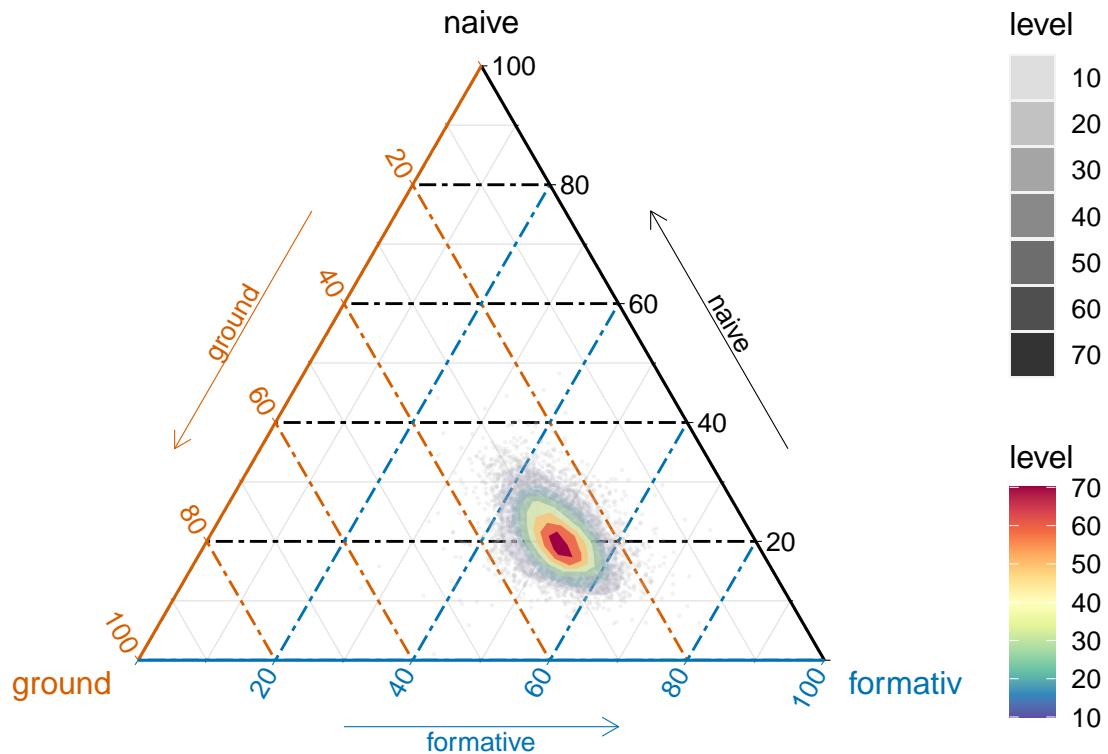
p2<-ggtern(data=all_matrix_selected, aes(x=ground, y=naive, z=formative)) +
  geom_point(size=0.09,alpha=0.2,colour="grey")+
  stat_density_tern(
    geom='polygon',
    bdl.val = NA,
    aes(fill=..level..,
```

```

    alpha=..level..))++
scale_fill_gradientn(colours=colormap)+#
#scale_color_viridis()
#scale_fill_gradient(low = "blue",
#                     high = "red")+
#scale_fill_gradientn(colours=colormap)
theme_bbw()
plot(p2, choix = "ind")

```

## Warning: Removed 1 rows containing non-finite values (StatDensityTern).



### 3. SOM clustered heatmap and line plot for each cluster

```

# Load required package
if (!require("ggplot2")) install.packages("ggplot2")
library(ggplot2)
if (!require("pheatmap")) install.packages("pheatmap")
library(pheatmap)
if (!require("gplots")) install.packages("gplots")
library(gplots)
if (!require("vegan")) install.packages("vegan")
library(vegan)
if (!require("permute")) install.packages("permute")

```

```

library(permute)
if (!require("lattice")) install.packages("lattice")
library(lattice)
if (!require("limma")) install.packages("limma")
library(limma)
if (!require("edgeR")) install.packages("edgeR")
library(edgeR)
if (!require("ggdendro")) BiocManager::install("ggdendro", version = "3.8")
library("ggdendro")
if (!require("cowplot")) install.packages("cowplot")
library("cowplot")
if (!require("biomaRt")) install.packages("biomaRt")
library('biomaRt')
if (!require("curl")) install.packages("curl")
library("curl")
if (!require("tidyverse")) install.packages("tidyverse")
library("tidyverse")
if (!require("ggpubr")) install.packages("ggpubr")
library("ggpubr")
if (!require("dplyr")) install.packages("dplyr")
library(dplyr)
if (!require("forcats")) install.packages("forcats")
library(forcats)
if (!require("ggtern")) install.packages("ggtern")
library(ggtern)
if (!require("viridis")) install.packages("viridis")
library(viridis)
if (!require("RColorBrewer")) install.packages("RColorBrewer")
library(RColorBrewer)
if (!require("som")) install.packages("som")

```

## Loading required package: som

```

library(som)
if (!require("reshape")) install.packages("reshape")

```

## Loading required package: reshape

```

##
## Attaching package: 'reshape'

```

```

## The following object is masked from 'package:dplyr':
##
```

```

##      rename

```

```

## The following objects are masked from 'package:tidyr':
##
```

```

##      expand, smiths

```

```

## The following object is masked from 'package:cowplot':
##
```

```

##      stamp

```

```

library(reshape)
if (!require("kohonen")) BiocManager::install("kohonen")

## Loading required package: kohonen

##
## Attaching package: 'kohonen'

## The following object is masked from 'package:som':
##      som

## The following object is masked from 'package:purrr':
##      map

library(kohonen)
if (!require("pheatmap")) install.packages("pheatmap")
library(pheatmap)
if (!require("magrittr")) install.packages("magrittr")

## Loading required package: magrittr

##
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':
##      set_names

## The following object is masked from 'package:tidyverse':
##      extract

library(magrittr)

summarySE_median <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                               conf.interval=.95, .drop=TRUE) {
  library(plyr)
  # New version of length which can handle NA's: if na.rm==T, don't count them
  length2 <- function (x, na.rm=FALSE) {
    if (na.rm) sum(!is.na(x))
    else      length(x)
  }
  # This does the summary. For each group's data frame, return a vector with
  dataac <- ddply(data, groupvars, .drop=.drop, .fun = function(xx, col) {
    c(sum    = sum(as.numeric(as.character(xx[[col]]))),
      N     = length2(xx[[col]]),
      median = median(as.numeric(as.character(xx[[col]]))),    #replace the mean by median
      sd    = sd(as.numeric(as.character(xx[[col]]))))
  }
}

```

```

        )
},
measurevar
)
# Rename the "mean" column
# dataac <- rename(dataac, c("mean" = measurevar))
dataac$se <- dataac$sd / sqrt(dataac$N) # Calculate standard error of the mean
ciMult <- qt(conf.interval/2 + .5, dataac$N-1)
dataac$ci <- dataac$se * ciMult
return(dataac)
}

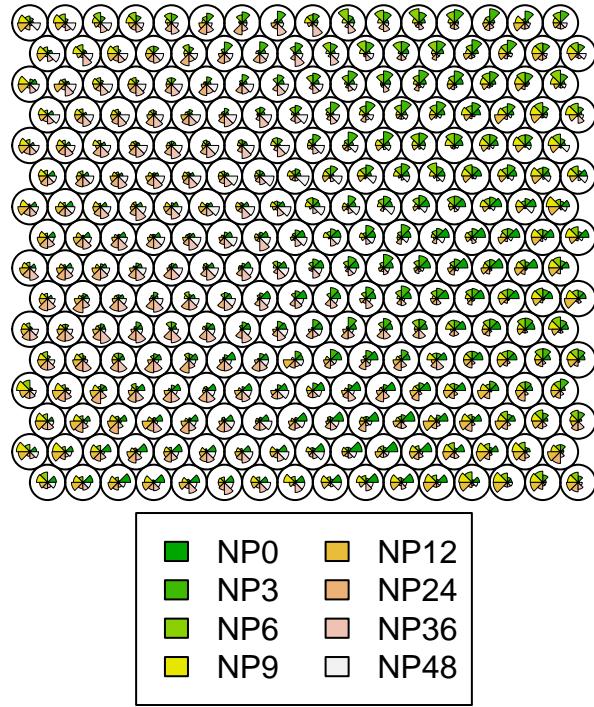
np_verified_cpm<- as.data.frame(na.omit(read.csv("D:\\Linlab\\_article\\01current_draft\\figure_integr"))
np_verified_map<- as.data.frame(na.omit(read.csv("D:\\Linlab\\_article\\01current_draft\\figure_integr

#filter the genes by cpm >10 (less than 1)
keep.exprs3 <- rowSums(np_verified_cpm>10)>=1
np_verified_cpm_filtered <- np_verified_cpm[keep.exprs3,]
# Set seed, just make sure you keep the same.
# Has to do with the randomization process.
np_verified_cpm_filtered<-as.data.frame(np_verified_cpm_filtered)
logFC_np<-cbind(log2(np_verified_cpm_filtered$NP0/apply(np_verified_cpm_filtered,1,min)),
                  log2(np_verified_cpm_filtered$NP3/apply(np_verified_cpm_filtered,1,min)),
                  log2(np_verified_cpm_filtered$NP6/apply(np_verified_cpm_filtered,1,min)),
                  log2(np_verified_cpm_filtered$NP9/apply(np_verified_cpm_filtered,1,min)),
                  log2(np_verified_cpm_filtered$NP12/apply(np_verified_cpm_filtered,1,min)),
                  log2(np_verified_cpm_filtered$NP24/apply(np_verified_cpm_filtered,1,min)),
                  log2(np_verified_cpm_filtered$NP36/apply(np_verified_cpm_filtered,1,min)),
                  log2(np_verified_cpm_filtered$NP48/apply(np_verified_cpm_filtered,1,min)))
rownames(logFC_np)<-rownames(np_verified_cpm_filtered)
colnames(logFC_np)<-colnames(np_verified_cpm_filtered)

#higher than 0.5:
scale_data <- as.matrix(t(scale(t(logFC_np))))
y1.som <- som(scale_data, somgrid(16, 16, "hexagonal"))
plot(y1.som, type = "codes") # Plots SOM clustering result.

```

## Codes plot

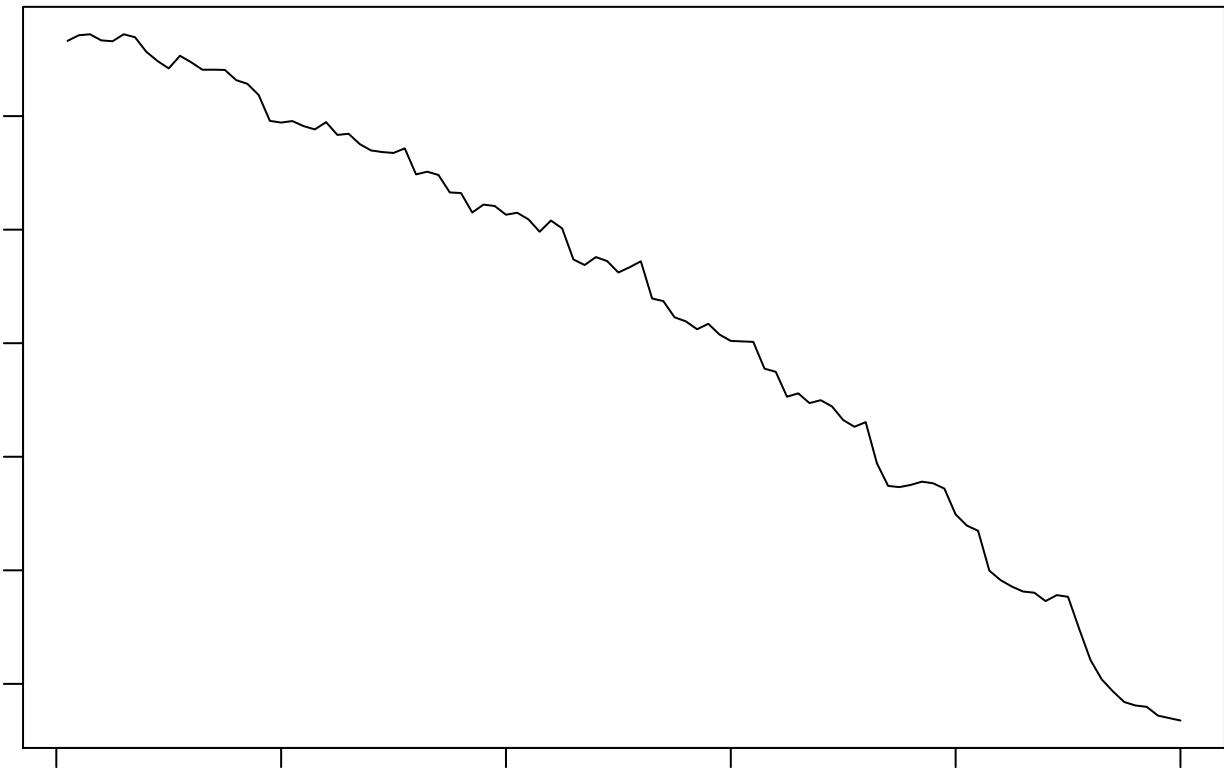


```
summary(y1.som)
```

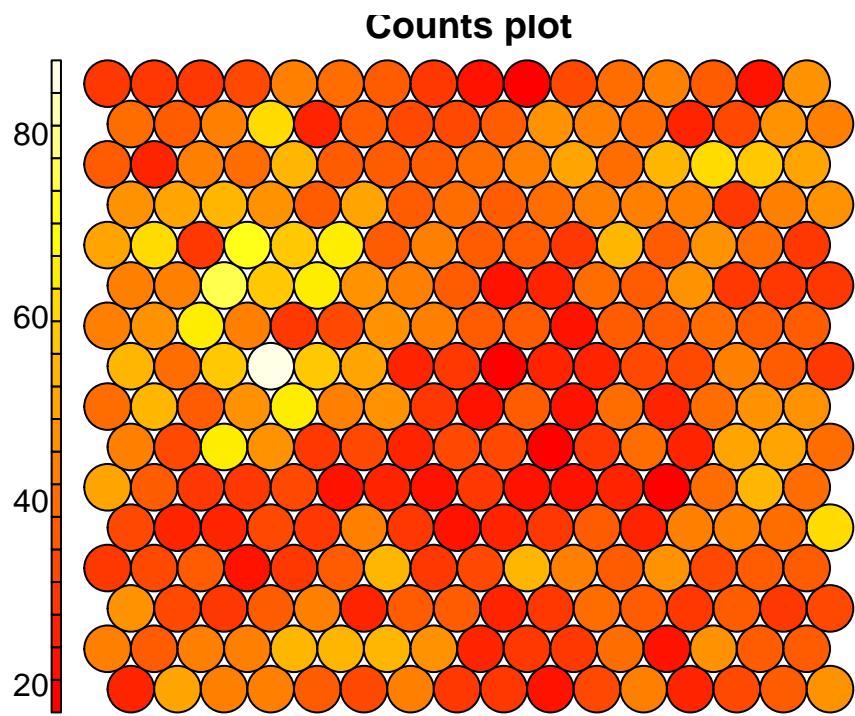
```
## SOM of size 16x16 with a hexagonal topology and a bubble neighbourhood function.  
## The number of data layers is 1.  
## Distance measure(s) used: sumofsquares.  
## Training data included: 9918 objects.  
## Mean distance to the closest unit in the map: 0.892.
```

```
plot(y1.som, type = "changes")
```

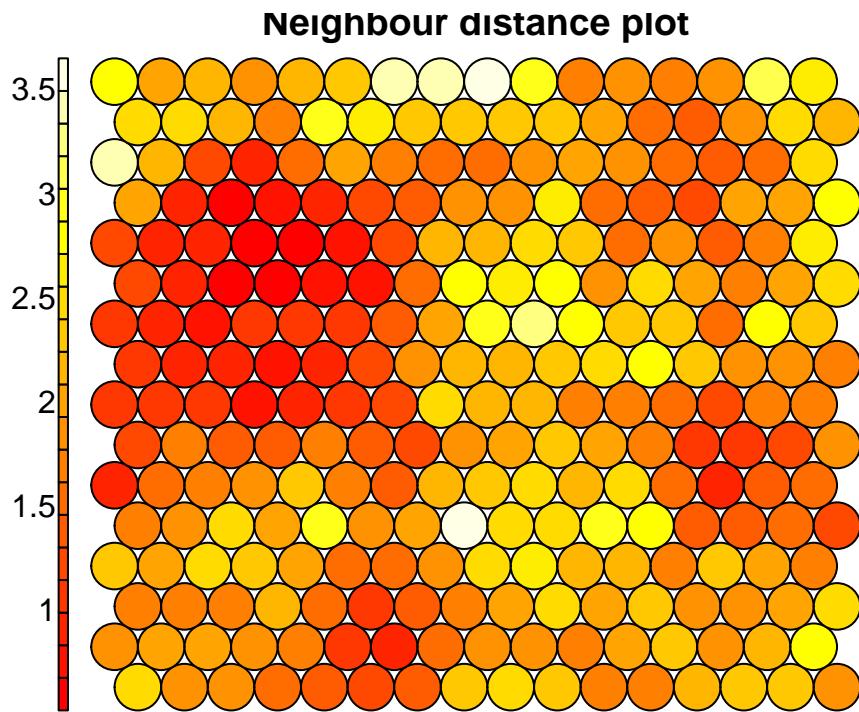
## Training progress



```
#plot(y1.som, type = "codes")
plot(y1.som, type = "counts")
```



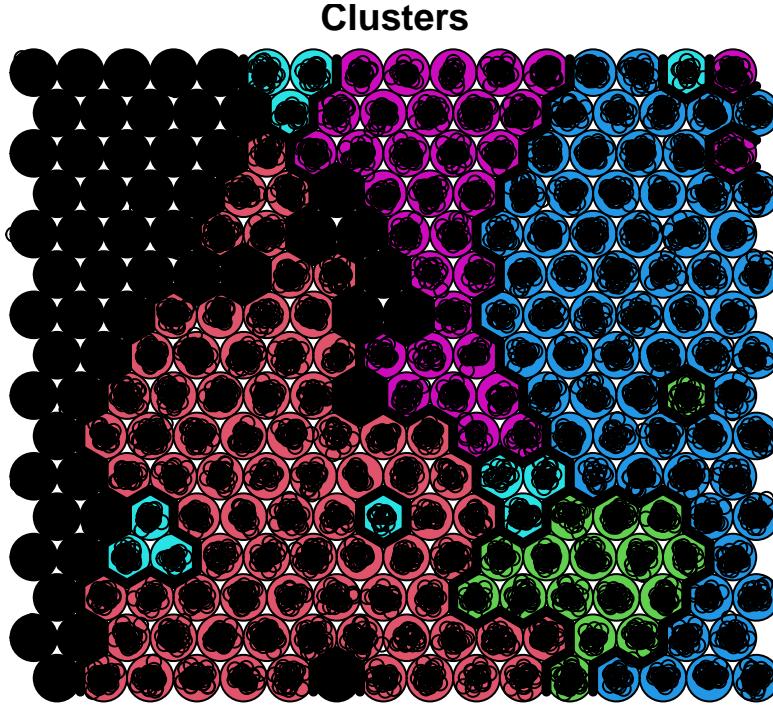
```
plot(y1.som, type="dist.neighbours")
```



```
#changed to dataframe to extract column names easier.
y1.som$data <- data.frame(y1.som$data)

## use hierarchical clustering to cluster the codebook vectors
som_cluster <- cutree(hclust(dist(y1.som$codes[[1]])), 6)

# plot these results:
plot(y1.som, type="mapping", bgcol = som_cluster, main = "Clusters")
add.cluster.boundaries(y1.som, som_cluster)
```



```
# Attach the hierachal cluster to the larger dataset data.val.
som_clusterKey <- data.frame(som_cluster)
som_clusterKey$unit.classif <- c(1:256)

data.val <- cbind(rownames(scale_data),logFC_np,y1.som$unit.classif,y1.som$distances)
head(data.val)

##          NPO          NP3
## mt-Co1 "mt-Co1"  "0.432088992257136" "0.214690241517864"
## Eef1a1 "Eef1a1"  "0"                  "0.156270912578575"
## mt-Nd1 "mt-Nd1"  "0.34228395223851"  "0.744794033998615"
## Hsp90ab1 "Hsp90ab1" "0.0399763900141415" "0.0782776731863463"
## mt-Cytb "mt-Cytb" "0.278109182854636"  "0.226537629352194"
## Gapdh  "Gapdh"   "0.217630088013921"  "0.130031086937028"
##          NP6          NP9          NP12
## mt-Co1 "0.524004175388898" "0.490142638392925" "0.142847634837492"
## Eef1a1 "0.122115260512587"  "0.0951302922652706" "0.145085124759032"
## mt-Nd1 "1.17750198435713"  "1.02027418621903"  "0.37503677983374"
## Hsp90ab1 "0.0467174203002468" "0"                  "0.193034316223319"
## mt-Cytb "0.512013600407915"  "0.538539655227453" "0.121559960908149"
## Gapdh  "0.0898547390247229"  "0"                  "0.218163274709048"
##          NP24         NP36         NP48
## mt-Co1 "0.2573644494170773" "0"          "0.231015389176267"  "144"
## Eef1a1 "0.198009794026292"  "0.258860808531396" "0.0496070299295384" "246"
## mt-Nd1 "0.290597302563465"  "0"          "0.318841388670731"  "224"
## Hsp90ab1 "0.11904819877968"  "0.338008162150881" "0.0122738951936183" "83"
```

```

## mt-Cytb "0.137425531834211" "0" "0.410763831303992" "192"
## Gapdh "0.491749933858927" "0.648614854221864" "0.301953632142631" "84"
##
## mt-Co1 "2.31169207172724"
## Eef1a1 "1.68173022934407"
## mt-Nd1 "1.27237542338171"
## Hsp90ab1 "2.1914344926092"
## mt-Cytb "2.95667115106877"
## Gapdh "0.821992709079061"

#Merge data.val with som_clusterKey
##change data.val to match som_cluster key
colnames(data.val)<-c("genes",colnames(logFC_np),"unit.classif","distance")
data.val <- merge(data.val, som_clusterKey, by.x = "unit.classif" ) #ignore warning, this is what you w
head(data.val)

##   unit.classif   genes      NPO      NP3      NP6      NP9
## 1          1 Sall4 0.956295719047547 0 0.838850713180436 0.923641878594033
## 2          1 Arid2 0.514842881610637 0 0.614294120746557 0.798495734480484
## 3          1 Gatad1 0.162820408362136 0 0.360861647163073 0.170373171046971
## 4          1 Tex15 0.586287022689415 0 0.808054705891715 0.614741492243626
## 5          1 Pms1 0.901595473827754 0 0.87828749631193 1.00901189372587
## 6          1 Smim19 0.818271745774312 0 0.738105571782232 0.77527789059655
##          NP12      NP24      NP36      NP48
## 1 0.953867031644463 0.681969511480935 0.628807234594752 1.31797217982866
## 2 0.480002619482986 0.484728223210413 0.540899464807753 0.656462234554201
## 3 0.283242492576157 0.091629610882687 0.144314382746968 0.254899333053696
## 4 0.440554496944988 0.715577605489961 0.0951454608506728 0.397081028601948
## 5 0.769516711937792 0.640257514670676 0.644976961447721 0.536389367161299
## 6 0.426786342052295 0.680367727993281 0.337163871167425 1.11329479798146
##       distance som_cluster
## 1 1.60369258750051 1
## 2 1.08752174240547 1
## 3 3.13017081706047 1
## 4 2.58438909207986 1
## 5 1.04048348464668 1
## 6 2.02014464636767 1

#write.csv(data.val,"som_cluster_220612_fc_zscore_6_final.csv")

data.val<-as.data.frame(data.val)
cluster_set_finalorder_c12<-data.val[order(data.val$som_cluster,decreasing=F),]

bk = unique(c(seq(0,1, length=100)))
cluster_set_finalorder_c12<-as.data.frame(cluster_set_finalorder_c12[,-2])
cluster_set_finalorder_c12=apply(cluster_set_finalorder_c12,2,as.numeric)
cluster_set_finalorder_c13<-data.val[order(data.val$som_cluster,decreasing=F),]
rownames(cluster_set_finalorder_c12)<-cluster_set_finalorder_c13[,2]

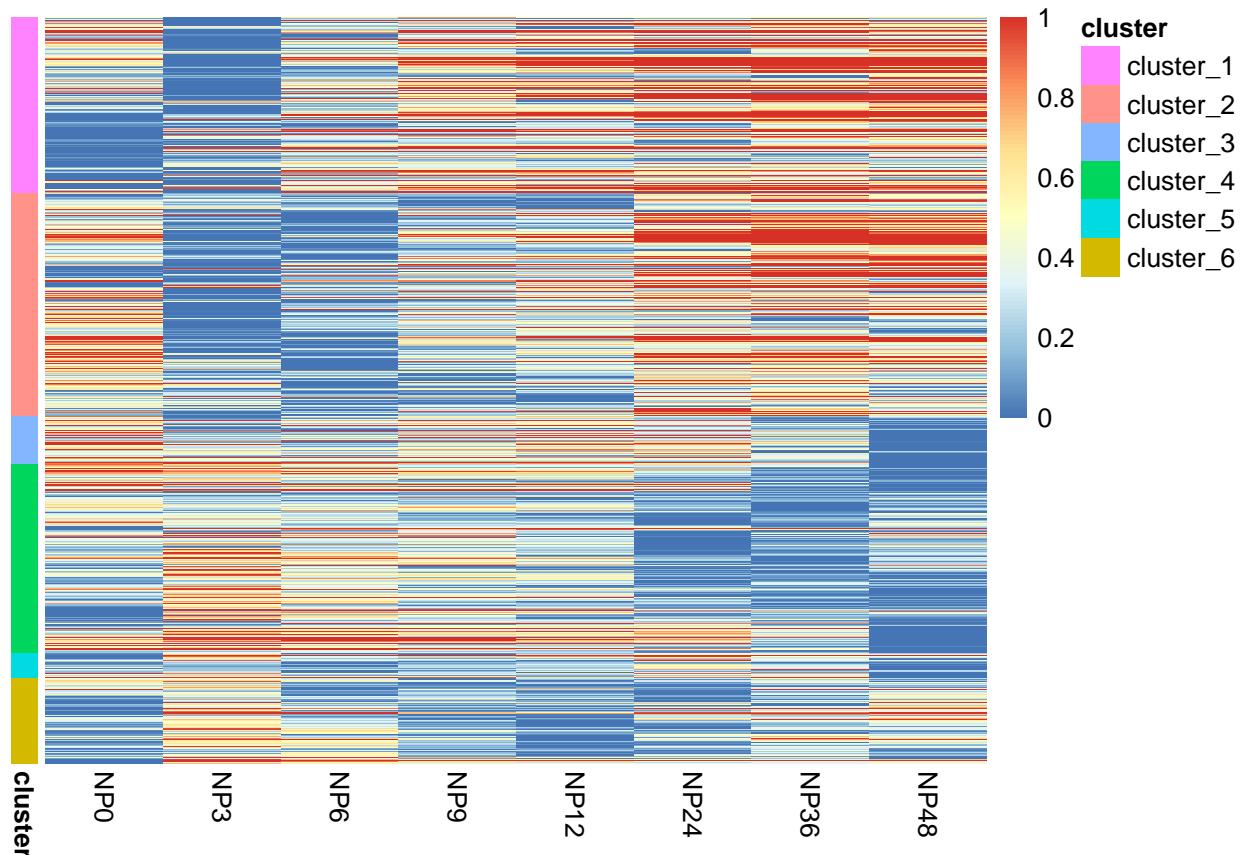
my_gene_col<-as.data.frame(gsub("^", "cluster_", cluster_set_finalorder_c13$som_cluster))
rownames(my_gene_col)<-rownames(cluster_set_finalorder_c12)
colnames(my_gene_col)<-c("cluster")
pheatmap( cluster_set_finalorder_c12[,c(2:9)])

```

```

        , annotation_row = my_gene_col
        ,breaks = bk
        ,scale ="none"
<#,annotation_row = pos_df
<#,clustering_method = "ward.D"
<#,clustering_distance_rows = "correlation"
        ,cluster_cols = FALSE
        ,cluster_rows = FALSE
        ,show_rownames= F
        ,show_colnames= T
        ,border_color = "NA"
)

```



```

##line_plot
cluster_set_finalorder_c122<-cluster_set_finalorder_c12[,c(2:9)]
cluster_set_finalorder_c122<-cbind(rownames(cluster_set_finalorder_c122),cluster_set_finalorder_c122)
#colnames(cluster_set_finalorder_c122)[9]<-"NP0"
cluster_set_finalorder_c122<-as.data.frame(cluster_set_finalorder_c122)
box_final <- cluster_set_finalorder_c122 %>% gather(colnames(cluster_set_finalorder_c122)[-1],
key='condition',value='abTR')
cluster_set<-cbind(cluster_set_finalorder_c12[,c("unit.classif","som_cluster")],
rownames(cluster_set_finalorder_c12))
colnames(cluster_set)<-c("unit.classif","som_cluster","V1")
box_final2 <- merge(cluster_set, box_final, by.x = "V1" )

```

```

box_final2_summary<-summarySE_median(box_final2,measurevar = "abTR",groupvars = c("som_cluster","condit

## -----
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## -----
## Attaching package: 'plyr'

## The following objects are masked from 'package:reshape':
##   rename, round_any

## The following object is masked from 'package:ggpubr':
##   mutate

## The following objects are masked from 'package:dplyr':
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarise

## The following object is masked from 'package:purrr':
##   compact

#box_final2_summary[is.na(box_final2_summary)] <- "NPO"
##line plot:
box_final2_summary$condition <- factor(box_final2_summary$condition,levels=c("NPO","NP3","NP6","NP9","N
p3<-ggplot(box_final2_summary,
            aes(x = condition, y = as.numeric(as.character(median)), group = som_cluster,color=som_cluste
            #geom_line(size = 2) +
            geom_smooth(se = FALSE, span = 0.95,size = 2)+
            labs(x = "Time after transition", y = "mRNA FC(log2)")+
            ylim(0,1)+
            scale_color_brewer(palette = 'Set3')+
            theme_bw() +
            theme( plot.title = element_text(hjust = 0.5),
                  text = element_text(size=22),
                  title=element_text(size = 30),
                  axis.title.x = element_text(size=24),
                  axis.title.y = element_text(size=24),
                  axis.text.x = element_text(size=16),
                  axis.text.y = element_text(size=22),
                  axis.title = element_text(face="bold")))
plot(p3)

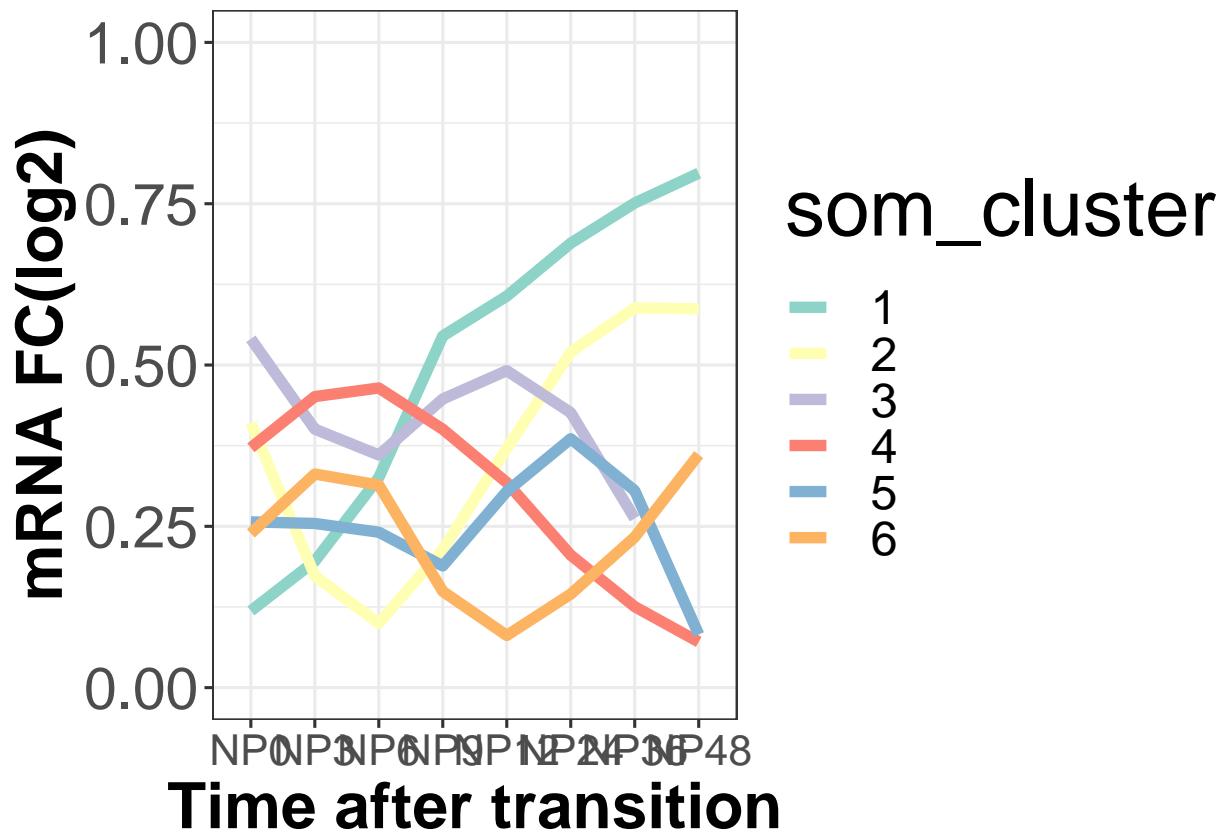
```

```

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 1 rows containing missing values (geom_smooth).

```



#### 4. Seurat Dot Plot for scRNA-seq and Colored D-score

```

# Load required package
if (!require("org.Mm.eg.db")) install.packages("org.Mm.eg.db")

## Loading required package: org.Mm.eg.db

## Loading required package: AnnotationDbi

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'


```

```

## The following objects are masked from 'package:parallel':
##
##   clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##   clusterExport, clusterMap, parApply, parCapply, parLapply,
##   parLapplyLB, parRapply, parSapply, parSapplyLB

## The following object is masked from 'package:som':
##
##   normalize

## The following objects are masked from 'package:dplyr':
##
##   combine, intersect, setdiff, union

## The following object is masked from 'package:limma':
##
##   plotMA

## The following objects are masked from 'package:stats':
##
##   IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##   anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##   dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##   grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##   order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##   rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##   union, unique, unsplit, which.max, which.min

## Loading required package: Biobase

## Welcome to Bioconductor
##
##   Vignettes contain introductory material; view with
##   'browseVignettes()'. To cite Bioconductor, see
##   'citation("Biobase")', and for packages 'citation("pkgname")'.

## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:plyr':
##
##   rename

```

```

## The following objects are masked from 'package:reshape':
##
##      expand, rename

## The following objects are masked from 'package:dplyr':
##
##      first, rename

## The following object is masked from 'package:tidyR':
##
##      expand

## The following object is masked from 'package:gplots':
##
##      space

## The following object is masked from 'package:base':
##
##      expand.grid

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:plyr':
##
##      desc

## The following objects are masked from 'package:dplyr':
##
##      collapse, desc, slice

## The following object is masked from 'package:purrr':
##
##      reduce

## The following object is masked from 'package:grDevices':
##
##      windows

##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:dplyr':
##
##      select

##

```

```

library(org.Mm.eg.db)
if (!require("ggplot2")) install.packages("ggplot2")
library(ggplot2)
if (!require("pheatmap")) install.packages("pheatmap")
library(pheatmap)
if (!require("gplots")) install.packages("gplots")
library(gplots)
if (!require("vegan")) install.packages("vegan")
library(vegan)
if (!require("permute")) install.packages("permute")
library(permute)
if (!require("lattice")) install.packages("lattice")
library(lattice)
if (!require("limma")) install.packages("limma")
library(limma)
if (!require("edgeR")) install.packages("edgeR")
library(edgeR)
if (!require("ggdendro")) BiocManager::install("ggdendro", version = "3.8")
library("ggdendro")
if (!require("cowplot")) install.packages("cowplot")
library("cowplot")
if (!require("biomaRt")) install.packages("biomaRt")
library('biomaRt')
if (!require("curl")) install.packages("curl")
library("curl")
if (!require("Seurat")) install.packages("Seurat")

```

```
## Loading required package: Seurat
```

```
## Attaching SeuratObject
```

```
## Attaching sp
```

```

library(Seurat)
if (!require("dplyr")) install.packages("dplyr")
library(dplyr)
if (!require("patchwork")) install.packages("patchwork")

```

```
## Loading required package: patchwork
```

```
##
## Attaching package: 'patchwork'
```

```

## The following object is masked from 'package:cowplot':
##
##     align_plots

```

```

library(patchwork)

summarySE_median <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                           conf.interval=.95, .drop=TRUE) {

```

```

library(plyr)
# New version of length which can handle NA's: if na.rm==T, don't count them
length2 <- function (x, na.rm=FALSE) {
  if (na.rm) sum(!is.na(x))
  else      length(x)
}

# This does the summary. For each group's data frame, return a vector with
datac <- ddply(data, groupvars, .drop=.drop,.fun = function(xx, col) {
  c(sum      = sum(as.numeric(as.character(xx[[col]]))),
    N       = length2(xx[[col]]),
    median = median(as.numeric(as.character(xx[[col]]))),   #replace the mean by median
    sd     = sd(as.numeric(as.character(xx[[col]])))
  )
},
measurevar
)
# Rename the "mean" column
# datac <- rename(datac, c("mean" = measurevar))
datac$se <- datac$sd / sqrt(datac$N)  # Calculate standard error of the mean
ciMult <- qt(conf.interval/2 + .5, datac$N-1)
datac$ci <- datac$se * ciMult
return(datac)
}

#do the pca part for 146w clustering:
matrix_anno <-as.data.frame(read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\classifier_rescSEQ_220823\\"))
tsne_clustering <-as.data.frame(read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\classifier_rescSEQ_220823\\"))
x2i_DR<-as.data.frame(read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\classifier_rescSEQ_220823\\lasso\\"))
rownames(x2i_DR)<-gsub("^", "X2i", rownames(x2i_DR))
pri_DR<-as.data.frame(read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\classifier_rescSEQ_220823\\lasso\\"))
rownames(pri_DR)<-gsub("^", "F48", rownames(pri_DR))
deg_26<-rbind(x2i_DR,pri_DR)
cell_cycle<-read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\classifier_rescSEQ_220823\\lasso_1113union.csv")

all_matrix<-na.omit(cbind(rownames(deg_26),cell_cycle[rownames(deg_26),],matrix_anno[rownames(deg_26),2:10]))
#install.packages("tidyverse")
colnames(all_matrix)<-c("cell","cell_cycle","state","D_score",colnames(tsne_clustering))
library(tidyverse)
tag <- paste(all_matrix[, "cell_cycle"], all_matrix[, "state"], sep="_", collapse = NULL)
all_matrix<-cbind(all_matrix,tag)
colnames(all_matrix)<-c("cell","cell_cycle","state","D_score",colnames(tsne_clustering),"tag")
all_matrix$D_score<-scale(all_matrix$D_score)
all_matrix$D_score[which(all_matrix$D_score>1)]<-1
all_matrix$D_score[which(abs(all_matrix$D_score)>1)]<-(-1)

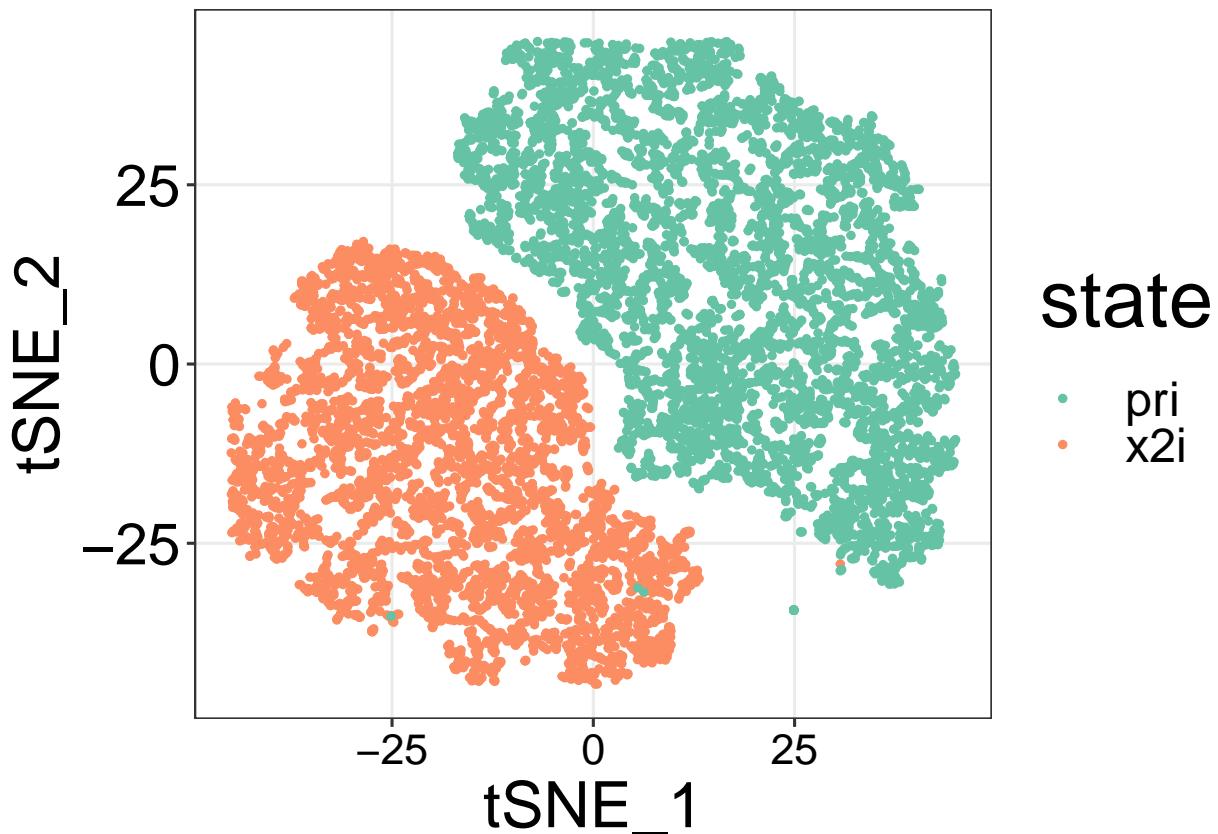
p4<-ggplot(all_matrix, aes(x=as.numeric(as.character(tSNE_1)), y=as.numeric(as.character(tSNE_2)), colour=tag))
  geom_point(size = 1) +
  #colour="lightcoral"
  labs(x = "tSNE_1", y = "tSNE_2")+
  #scale_colour_manual(values=cc)+
  scale_color_brewer(palette = 'Set2')+
  theme_bw()+
  ylim(-45,45)+
  xlim(-45,45)+
```

```

theme(
  text = element_text(size=22),
  panel.background = element_rect(fill = "transparent", colour = NA),
  panel.grid.minor = element_blank(),
  axis.text=element_text(color='black'),
  plot.title = element_text(hjust = 0.5),
  title=element_text(size = 30),
  axis.title.x = element_text(size=24),
  axis.title.y = element_text(size=24),
  axis.text.x = element_text(size=16),
  axis.text.y = element_text(size=22))
plot(p4)

```

## Warning: Removed 680 rows containing missing values (geom\_point).



```

#colored DR for the dot plot
####remember to replace >1 and <-1(use abs)
p4_2<-ggplot(all_matrix, aes(x=as.numeric(as.character(tSNE_1)), y=as.numeric(as.character(tSNE_2)),shape=as.character(as.factor(dr)),color=as.factor(as.character(dr)),size=2))
  geom_point(size = 2) +
  #colour="lightcoral"
  labs(x = "tSNE_1", y = "tSNE_2")+
  #scale_fill_continuous(low='#fcbb1',high='#67000d')+
  scale_colour_gradient2(
    low = "#1a9850",
    high = "#67000d")

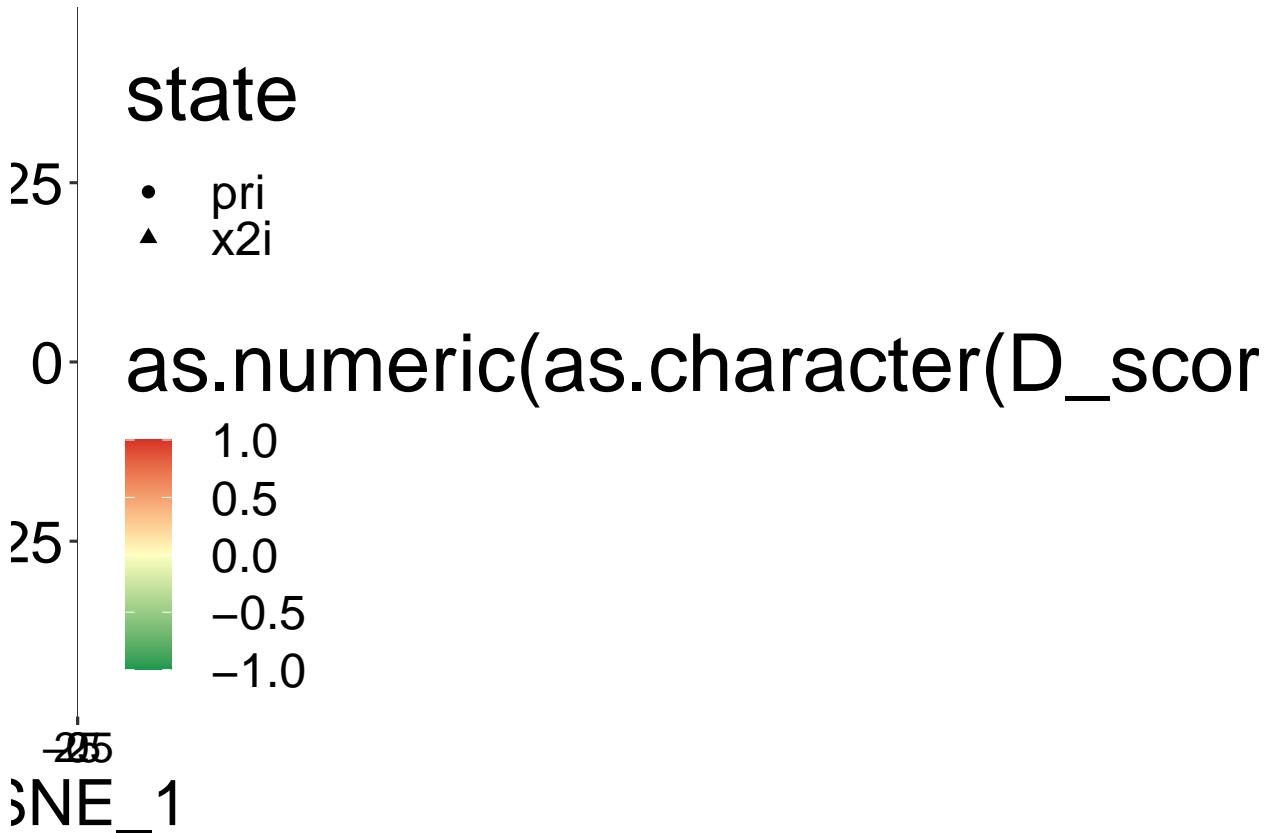
```

```

limits=c(-1, 1),
mid = "#ffffbf",
midpoint = 0,
#space = "Lab",
high = "#d73027")+
#scale_colour_manual(values=cc)+
theme_bw()+
ylim(-45,45)+
xlim(-45,45) +
theme(
  text = element_text(size=22),
  panel.background = element_rect(fill = "transparent", colour = NA),
  panel.grid.minor = element_blank(),
  axis.text=element_text(color='black'),
  plot.title = element_text(hjust = 0.5),
  title=element_text(size = 30),
  axis.title.x = element_text(size=24),
  axis.title.y = element_text(size=24),
  axis.text.x = element_text(size=16),
  axis.text.y = element_text(size=22))
plot(p4_2)

```

## Warning: Removed 680 rows containing missing values (geom\_point).

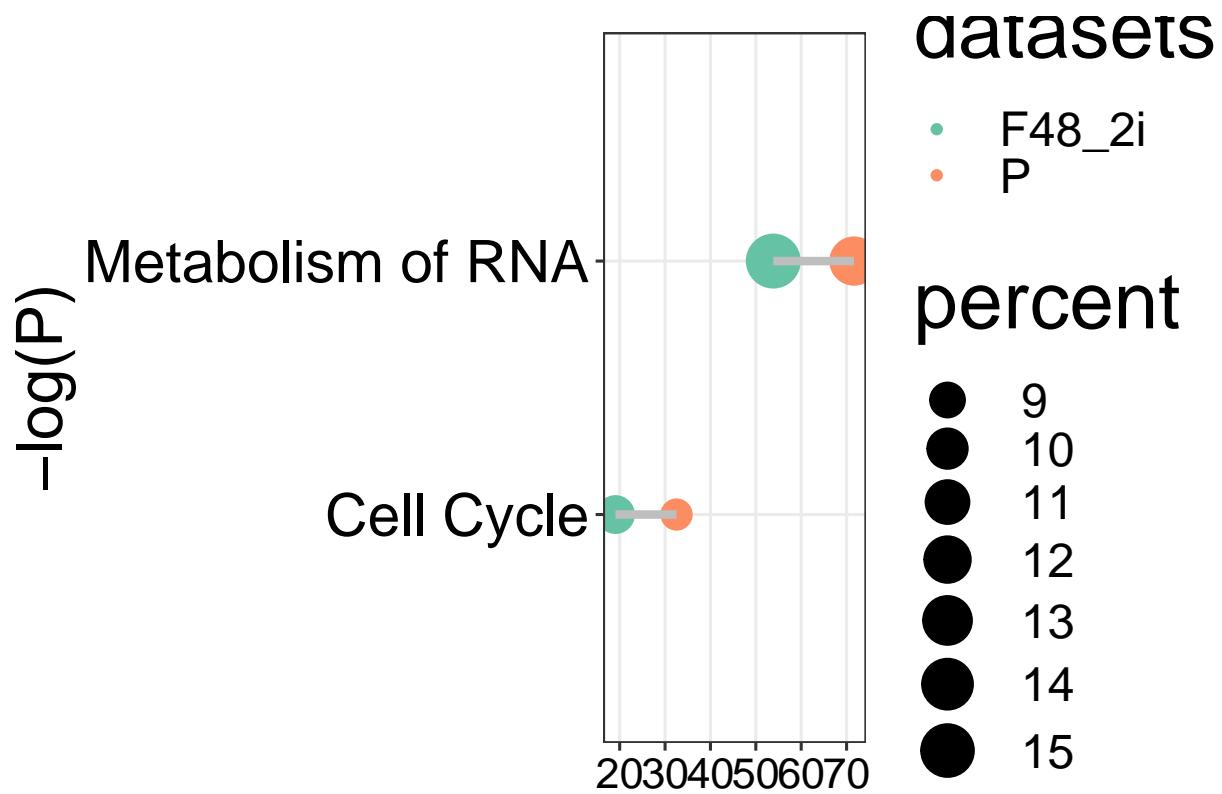


## 5. Dot-Line Plot for GO Clusters

```
if (!require("ggplot2")) install.packages("ggplot2")
library(ggplot2)
if (!require("gplots")) install.packages("gplots")
library(gplots)
#do the pca part for 146w clustering:
matrix_anno <-as.data.frame(read.csv("D:\\Linlab\\_article\\02_version1\\figure2\\04_go_plot\\go_two_s
##12-4 for drawing
p5<-ggplot(matrix_anno, aes(x=Description, y=Log10.P., colour=datasets, size=percent, group=Description)) +
  geom_point() +
  #colour="lightcoral"
  labs(x = "-log(P)", y = "")+
  #scale_colour_manual(values=cc)+
  scale_color_brewer(palette = 'Set2')+
  theme_bw()+
  #ylim(-45,45)+
  #xlim(-45,45)+
  scale_x_discrete(expand = c(0.45,0.45))+ 
  scale_size_continuous(range = c(5,9))+ 
  coord_flip()+
  theme(
    text = element_text(size=22),
    panel.background = element_rect(fill = "transparent", colour = NA),
    panel.grid.minor = element_blank(),
    axis.text=element_text(color='black'),
    plot.title = element_text(hjust = 0.5),
    title=element_text(size = 30),
    axis.title.x = element_text(size=24),
    axis.title.y = element_text(size=24),
    axis.text.x = element_text(size=16),
    axis.text.y = element_text(size=22))

## Coordinate system already present. Adding new coordinate system, which will replace the existing one

#matrix_anno$Description<-factor(matrix_anno$Description)
p5+geom_line(size = 1.5, color = "grey")
```



## 6. Venn Plot

```

if (!require("VennDiagram")) install.packages("VennDiagram")

## Loading required package: VennDiagram

## Loading required package: grid

## Loading required package: futile.logger

##
## Attaching package: 'VennDiagram'

## The following object is masked from 'package:ggbpubr':
## 
##     rotate

#install.packages("VennDiagram")
library(VennDiagram)
x2iF48<-read.csv("D:\\Linlab\\\\OCT4 induction\\\\mechanism\\\\classifier_rescSEQ_220823\\\\anova_1115_YMS\\\\2iF48")
P<-read.csv("D:\\Linlab\\\\OCT4 induction\\\\mechanism\\\\classifier_rescSEQ_220823\\\\anova_1115_YMS\\\\P_qvalue")
gene_final<-read.csv("D:\\Linlab\\\\OCT4 induction\\\\mechanism\\\\classifier_rescSEQ_220823\\\\anova_1115_YMS\\\\nrow(na.omit(x2iF48[gene_final[, \"X\"],]))")

```

```

## [1] 271

nrow(na.omit(P[gene_final[, "X"], ]))

## [1] 182

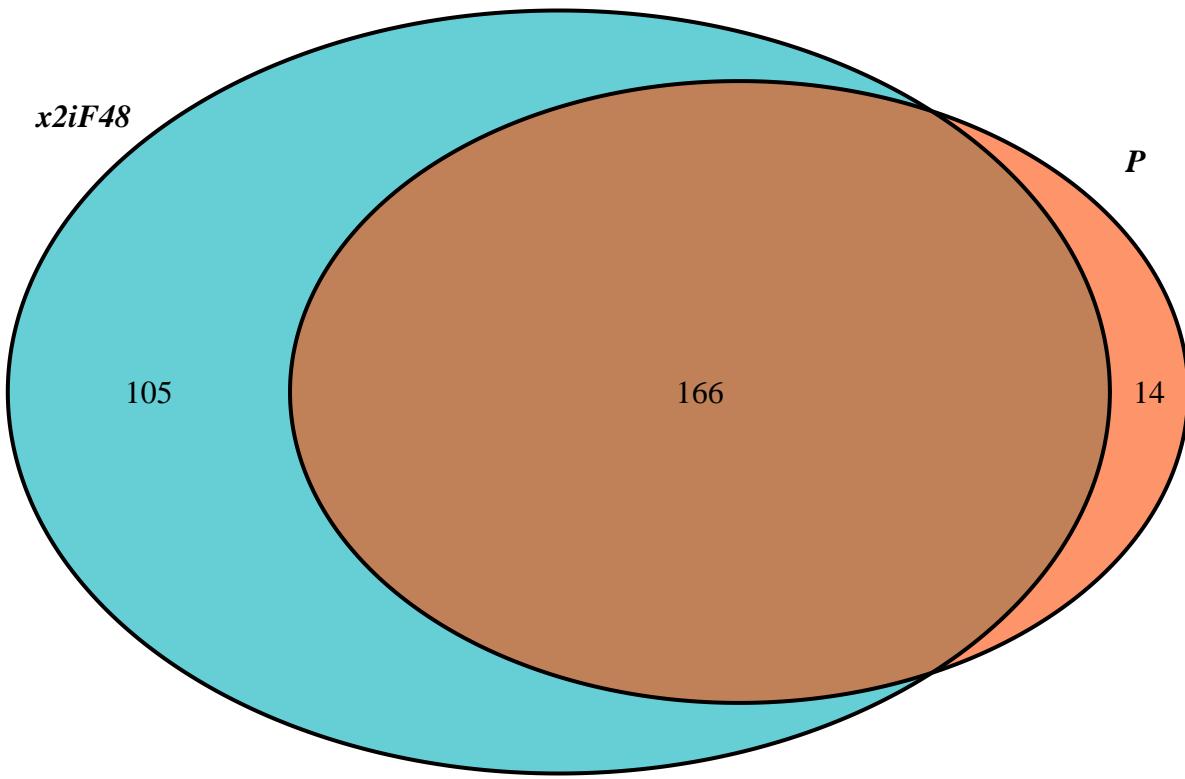
x2iF48_final<-na.omit(na.omit(x2iF48[gene_final[, "X"], ]))
P_final<-na.omit(na.omit(P[gene_final[, "X"], ]))
length(intersect(rownames(x2iF48_final), rownames(P_final)))

## [1] 166

all_final<-cbind(x2iF48_final[intersect(rownames(x2iF48_final), rownames(P_final)), ],
                  P_final[intersect(rownames(x2iF48_final), rownames(P_final)), ])
colnames(all_final)<-c("x2iF48_X", "x2iF48_score", "x2iF48_p_values", "x2iF48_q_values", "P_X", "P_score", "P_q")

#mydata<-list(Lif2i=1:271, NP=105:287)
#ggVennDiagram(mydata)
#+
#  scale_fill_brewer(palette="Set2")
temp<-venn.diagram(list(x2iF48=1:271, P=106:285),
                    alpha=c(0.6,0.6),
                    fill=c("#00AFBB", "#FC4E07"), cat.fontface=4,
                    main.cex= 2, main.fontface= 2,
                    filename= NULL, scaled=T)
grid.draw(temp)

```



7. Line Plot for Specific Genes in Different Condition (seperate patches)

```

if (!require("dplyr")) install.packages("dplyr")
library(dplyr)
if (!require("pheatmap")) install.packages("pheatmap")
library(pheatmap)
if (!require("gplots")) install.packages("gplots")
library(gplots)
if (!require("vegan")) install.packages("vegan")
library(vegan)
if (!require("permute")) install.packages("permute")
library(permute)
if (!require("lattice")) install.packages("lattice")
library(lattice)
if (!require("magrittr")) install.packages("magrittr")
library(magrittr)
if (!require("tidyverse")) install.packages("tidyverse")
library(tidyverse)
if (!require("patchwork")) install.packages("patchwork")
library(patchwork)
##Deadenylation-dependent mRNA decay
deade_genes2<-read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\classifier_rescSEQ_220823\\anova_1115_YM"
deade_genes2<-cbind(rownames(deade_genes2),deade_genes2)

```

```

library(patchwork)
deade_genes_Pabpc1<-deade_genes2[["Pabpc1",]
line_Pabpc1 <- deade_genes_Pabpc1 %>% gather(colnames(deade_genes_Pabpc1)[-1],
                                               key='condition',value='cpm')
colnames(line_Pabpc1)<-c("genes","condition","cpm")
line_Pabpc1_1<-line_Pabpc1[c(1:4),]
line_Pabpc1_1[, "cpm"]<-scale(line_Pabpc1_1[, "cpm"])
#pab<-line_Pabpc1[line_Pabpc1[, 1]=="Pabpc1",]
pdf(file = "line_Pabpc1.pdf",width = 9.5,height = 3)
p1<-ggplot(line_Pabpc1_1,
            aes(x = condition, y = as.numeric(as.character(cpm)),group = 1))+ 
  geom_line(colour = "gray",size = 1,alpha=2) +
  scale_x_discrete(expand = c(0.02, 0.02))+
  labs(x = "", y = "z-score")+
  ylim(-1.5,1.5)+
  theme_bw() +
  theme( plot.title = element_text(hjust = 0.5),
         text = element_text(size=22),
         title=element_text(size = 30),
         panel.grid.major = element_line(colour="white"),
         panel.grid.minor = element_line(colour="white"),
         axis.title.x = element_text(size=24),
         axis.title.y = element_text(size=8),
         axis.text.x = element_text(size=8),
         axis.text.y = element_text(size=8),
         axis.title = element_text(face="bold"),
      )
)

line_Pabpc1_2<-line_Pabpc1[c(5:8),]
line_Pabpc1_2[, "cpm"]<-scale(line_Pabpc1_2[, "cpm"])
p2<-ggplot(line_Pabpc1_2,
            aes(x = condition, y = as.numeric(as.character(cpm)),group = 1))+ 
  geom_line(colour = "gray",size = 1,alpha=2) +
  scale_x_discrete(expand = c(0.02, 0.02))+
  labs(x = "", y = "z-score")+
  ylim(-1.5,1.5)+
  theme_bw() +
  theme( plot.title = element_text(hjust = 0.5),
         text = element_text(size=22),
         title=element_text(size = 30),
         panel.grid.major = element_line(colour="white"),
         panel.grid.minor = element_line(colour="white"),
         axis.title.x = element_text(size=24),
         axis.title.y = element_text(size=8),
         axis.text.x = element_text(size=8),
         axis.text.y = element_text(size=8),
         axis.title = element_text(face="bold"))

library(ggplot2)
line_Pabpc1_3<-line_Pabpc1[c(9:12),]
line_Pabpc1_3[, "cpm"]<-scale(line_Pabpc1_3[, "cpm"])
p3<-ggplot(line_Pabpc1_3,
            aes(x = condition, y = as.numeric(as.character(cpm)),group = 1))+
```

```

geom_line(colour = "gray",size = 1,alpha=2) +
scale_x_discrete(expand = c(0.02, 0.02))+ 
labs(x = "", y = "z-score")+
ylim(-1.5,1.5)+ 
theme_bw() +
theme( plot.title = element_text(hjust = 0.5),
text = element_text(size=22),
title=element_text(size = 30),
panel.grid.major = element_line(colour="white"),
panel.grid.minor = element_line(colour="white"),
axis.title.x = element_text(size=24),
axis.title.y = element_text(size=8),
axis.text.x = element_text(size=8),
axis.text.y = element_text(size=8),
axis.title = element_text(face="bold"))

p1|p2|p3
dev.off()

## pdf
## 2

library(patchwork)
deade_genes_Eif4a2<-deade_genes2[["Eif4a2",]
line_Eif4a2 <- deade_genes_Eif4a2 %>% gather(colnames(deade_genes_Eif4a2)[-1],
key='condition',value='cpm')
colnames(line_Eif4a2)<-c("genes","condition","cpm")
line_Eif4a2_1<-line_Eif4a2[c(1:4),]
line_Eif4a2_1[, "cpm"]<-scale(line_Eif4a2_1[, "cpm"])
#pab<-line_Eif4a2[line_Eif4a2[, 1]=="Eif4a2",]
pdf(file = "line_Eif4a2.pdf",width = 9.5,height = 3)
p1_Eif4a2<-ggplot(line_Eif4a2_1,
aes(x = condition, y = as.numeric(as.character(cpm)),group = 1))+
geom_line(colour = "gray",size = 1,alpha=2) +
scale_x_discrete(expand = c(0.02, 0.02))+ 
labs(x = "", y = "z-score")+
ylim(-1.5,1.5)+ 
theme_bw() +
theme( plot.title = element_text(hjust = 0.5),
text = element_text(size=22),
title=element_text(size = 30),
panel.grid.major = element_line(colour="white"),
panel.grid.minor = element_line(colour="white"),
axis.title.x = element_text(size=24),
axis.title.y = element_text(size=8),
axis.text.x = element_text(size=8),
axis.text.y = element_text(size=8),
axis.title = element_text(face="bold"),
)

line_Eif4a2_2<-line_Eif4a2[c(5:8),]
line_Eif4a2_2[, "cpm"]<-scale(line_Eif4a2_2[, "cpm"])
p2_Eif4a2<-ggplot(line_Eif4a2_2,

```

```

aes(x = condition, y = as.numeric(as.character(cpm)),group = 1))+  

geom_line(colour = "gray",size = 1,alpha=2) +  

scale_x_discrete(expand = c(0.02, 0.02))+  

labs(x = "", y = "z-score") +  

ylim(-1.5,1.5)+  

theme_bw() +  

theme( plot.title = element_text(hjust = 0.5),  

text = element_text(size=22),  

title=element_text(size = 30),  

panel.grid.major = element_line(colour="white"),  

panel.grid.minor = element_line(colour="white"),  

axis.title.x = element_text(size=24),  

axis.title.y = element_text(size=8),  

axis.text.x = element_text(size=8),  

axis.text.y = element_text(size=8),  

axis.title = element_text(face="bold"))

library(ggplot2)
line_Eif4a2_3<-line_Eif4a2[c(9:12),]
line_Eif4a2_3[, "cpm"]<-scale(line_Eif4a2_3[, "cpm"])
p3_Eif4a2<-ggplot(line_Eif4a2_3,
                    aes(x = condition, y = as.numeric(as.character(cpm)),group = 1))+  

geom_line(colour = "gray",size = 1,alpha=2) +  

scale_x_discrete(expand = c(0.02, 0.02))+  

labs(x = "", y = "z-score") +  

ylim(-1.5,1.5)+  

theme_bw() +  

theme( plot.title = element_text(hjust = 0.5),  

text = element_text(size=22),  

title=element_text(size = 30),  

panel.grid.major = element_line(colour="white"),  

panel.grid.minor = element_line(colour="white"),  

axis.title.x = element_text(size=24),  

axis.title.y = element_text(size=8),  

axis.text.x = element_text(size=8),  

axis.text.y = element_text(size=8),  

axis.title = element_text(face="bold"))

p1_Eif4a2|p2_Eif4a2|p3_Eif4a2
dev.off()

```

```

## pdf
##    2

#then use AI to modify the picture

```

## 8. Distribution/Density Plot

```

if (!require("ggplot2")) install.packages("ggplot2")
library("ggplot2")
if (!require("dplyr")) install.packages("dplyr")

```

```

library("dplyr")
x2iP2A<-read.table("D:\\Linlab\\OCT4 induction\\mechanism\\PIPA_seq\\PIPA-220214\\results\\sample_2iP2A")
x2iP2A<-cbind(x2iP2A,"2iP2A")
colnames(x2iP2A)<-c("length","group")
nrow(x2iP2A)

## [1] 181185

x2iP2AD<-read.table("D:\\Linlab\\OCT4 induction\\mechanism\\PIPA_seq\\PIPA-220214\\results\\sample_2iP2AD")
x2iP2AD<-cbind(x2iP2AD,"2iP2AD")
colnames(x2iP2AD)<-c("length","group")
nrow(x2iP2AD)

## [1] 62469

x2i<-read.table("D:\\Linlab\\OCT4 induction\\mechanism\\PIPA_seq\\PIPA-220214\\results\\sample_2i_trimm")
x2i<-cbind(x2i,"2i")
colnames(x2i)<-c("length","group")
lif<-read.table("D:\\Linlab\\OCT4 induction\\mechanism\\PIPA_seq\\PIPA-220214\\results\\sample_lif_trimm")
lif<-cbind(lif,"lif")
nrow(lif)

## [1] 92687

colnames(lif)<-c("length","group")
Pri<-read.table("D:\\Linlab\\OCT4 induction\\mechanism\\PIPA_seq\\PIPA-220214\\results\\sample_Pri_trimm")
Pri<-cbind(Pri,"Pri")
colnames(Pri)<-c("length","group")
nrow(Pri)

## [1] 1073653

U_total<-as.data.frame(rbind(x2iP2A,x2iP2AD,x2i,lif,Pri))
#U_total<-as.data.frame(rbind(x2i,Pri))
U_total %>%
  group_by(group) %>%
  summarise(avg = mean(length))

##      avg
## 1 67.9566

#change the Adjacency matrix to density matrix
#####draw the density plot
#pdf(file = "poly(A)_length_distribution.pdf",width=20,height=16)
p8<-ggplot(U_total,aes(x=as.numeric(as.character(length)),colour=group))+
  # ylim(0,0.4)+
  # xlim(2,20)+
  geom_density(size = 2)+
  ggtitle("Poly(A) length distribution") +
  labs(y = "Density", x = "Poly(A) length")+

```

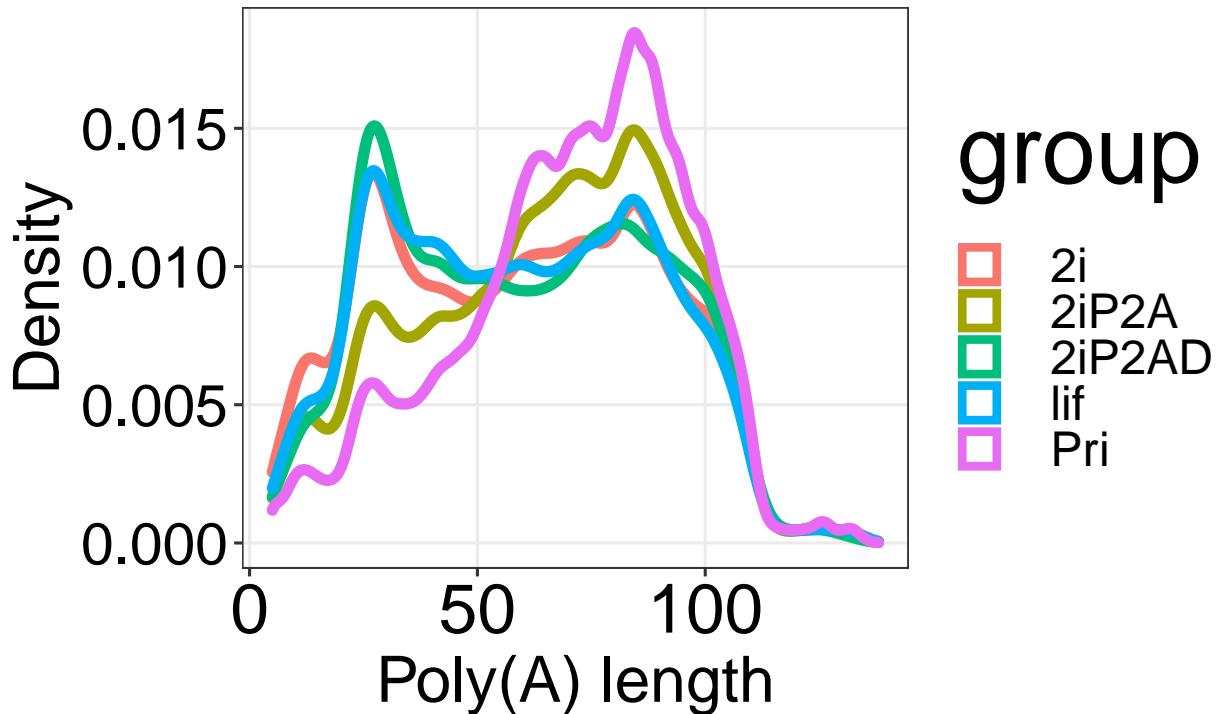
```

theme_bw()+
theme(
  text = element_text(size=22),
  panel.background = element_rect(fill = "transparent", colour = NA),
  panel.grid.minor = element_blank(),
  axis.text=element_text(color='black'),
  plot.title = element_text(hjust = 0.5),
  title=element_text(size = 36),
  axis.title.x = element_text(size=24),
  axis.title.y = element_text(size=24),
  axis.text.x = element_text(size=26),
  axis.text.y = element_text(size=22))

plot(p8)

```

# Poly(A) length distribution



```
#dev.off()
```

## 9. Dot Plot With Density Info

```

if (!require("ggplot2")) install.packages("ggplot2")
library(ggplot2)
if (!require("gplots")) install.packages("gplots")
library(gplots)

```

```

if (!require("vegan")) install.packages("vegan")
library(vegan)
if (!require("permute")) install.packages("permute")
library(permute)
if (!require("lattice")) install.packages("lattice")
library(lattice)
if (!require("limma")) install.packages("limma")
library(limma)
if (!require("edgeR")) install.packages("edgeR")
library(edgeR)
if (!require("ggdendro")) BiocManager::install("ggdendro", version = "3.8")
library("ggdendro")
if (!require("cowplot")) install.packages("cowplot")
library("cowplot")
if (!require("biomaRt")) install.packages("biomaRt")
library('biomaRt')
if (!require("curl")) install.packages("curl")
library("curl")
if (!require("tidyverse")) install.packages("tidyverse")
library("tidyverse")
if (!require("plyr")) install.packages("plyr")
library(plyr)
if (!require("dplyr")) install.packages("dplyr")
library(dplyr)
if (!require("viridis")) install.packages("viridis")
library(viridis)
total_MAP_211119 <- as.data.frame(read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\perturbation_exp_091"))
total_MAP_211205 <- as.data.frame(read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\perturbation_exp_091"))
TNPSiNP_MAP_rep1 <- as.data.frame(read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\perturbation_exp_091"))
#delete genes which deg == 0
keep.exprs1 <- rowSums(total_MAP_211119>0)>=ncol(total_MAP_211119)
total_MAP_211119 <- total_MAP_211119[keep.exprs1,]

keep.exprs2 <- rowSums(total_MAP_211205>0)>=ncol(total_MAP_211205)
total_MAP_211205 <- total_MAP_211205[keep.exprs2,]

keep.exprs3 <- rowSums(TNPSiNP_MAP_rep1>0)>=ncol(TNPSiNP_MAP_rep1)
TNPSiNP_MAP_rep1 <- TNPSiNP_MAP_rep1[keep.exprs3,]
total_TNPSiNP<-read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\perturbation_exp_0910\\SEQ220505_TNPSiNP")
#filter by Rcounts>50 >ncol/2 cpm(4.58)
total_TNPSiNP<-total_TNPSiNP[,c(1,2,3,7,8,9)]
keep.exprs_total <- rowSums(total_TNPSiNP>4.58)>=ncol(total_TNPSiNP)/2
total_TNPSiNP <- total_TNPSiNP[keep.exprs_total,]
TNPSiNP_MAP_rep1<-na.omit(TNPSiNP_MAP_rep1[rownames(total_TNPSiNP),])

#the pdf format is 8-6 rep1
get_density <- function(x,y,...){
  dens <- MASS::kde2d(x, y, ...)
  ix <- findInterval(x, dens$x)
  iy <- findInterval(y, dens$y)
  ii <- cbind(ix, iy)
  return(dens$z[ii])
}

```

```

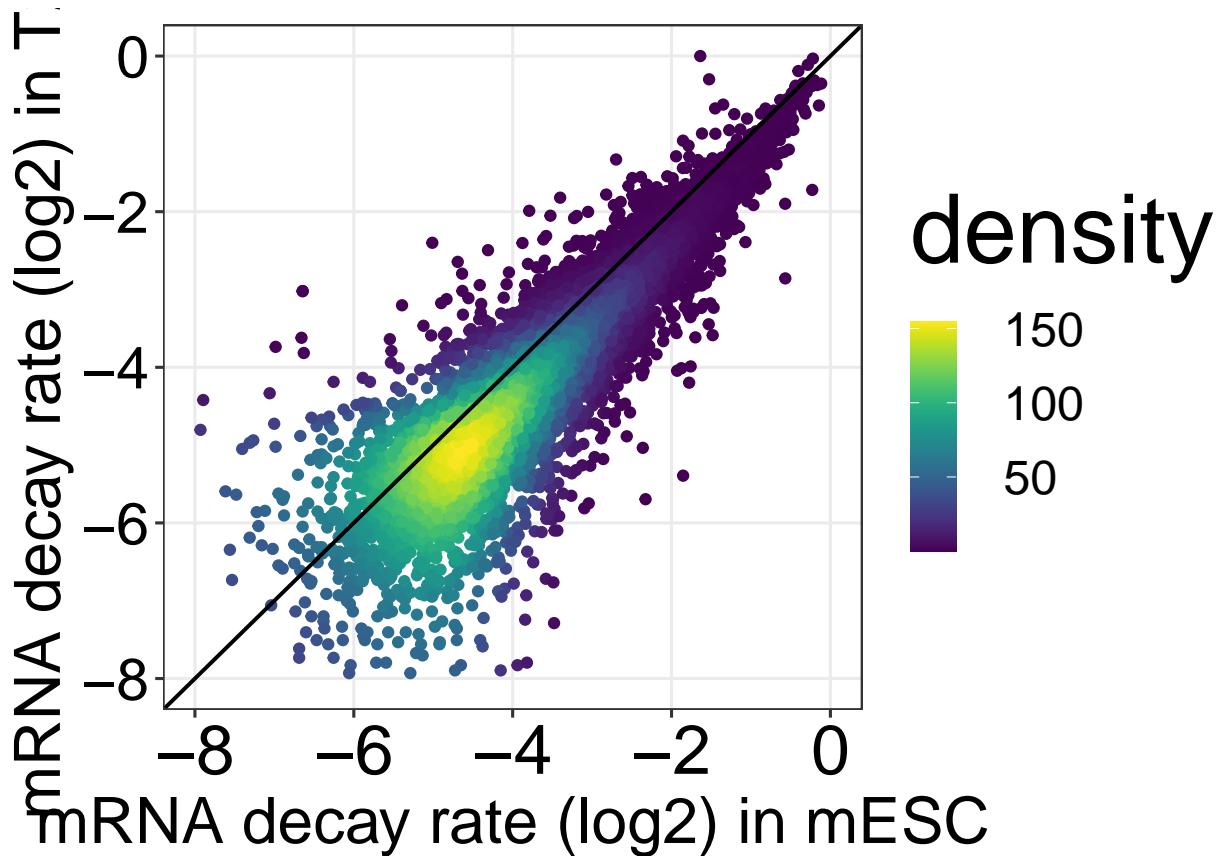
TNPSiNP_MAP_rep1$density <- get_density(TNPSiNP_MAP_rep1$X2iP2A.MAP, TNPSiNP_MAP_rep1$X2iP2AD.MAP,n = 100)
#dot2<-ggplot(TNPSiNP_MAP_rep1, aes(y=log2(as.numeric(as.character(X2iP2AD.MAP))), x=log2(as.numeric(as.character(X2iP2A.MAP)))) , color=TNPSiNP_MAP_rep1$X2iP2AD.MAP)

p9<-ggplot(TNPSiNP_MAP_rep1) +
  geom_point(aes(x=log2(as.numeric(as.character(X2iP2A.MAP))), y=log2(as.numeric(as.character(X2iP2AD.MAP)))), color=TNPSiNP_MAP_rep1$X2iP2AD.MAP)
  scale_color_viridis()+
  guides(alpha="none") +
  ylim(-8,0)+ 
  xlim(-8,0)+ 
  labs(x = "mRNA decay rate (log2) in mESC", y = "mRNA decay rate (log2) in T2iD")+
  theme_bw()+
  theme(
    text = element_text(size=22),
    panel.background = element_rect(fill = "transparent",colour = NA),
    panel.grid.minor = element_blank(),
    axis.text=element_text(color='black'),
    plot.title = element_text(hjust = 0.5),
    title=element_text(size = 36),
    axis.title.x = element_text(size=24),
    axis.title.y = element_text(size=24),
    axis.text.x = element_text(size=26),
    axis.text.y = element_text(size=22))

p9<-p9+
  geom_abline(slope=1,lwd=0.7,intercept = 0)
plot(p9)

```

## Warning: Removed 44 rows containing missing values (geom\_point).



```
t.test(TNPsNP_MAP_rep1$X2iP2A.MAP, TNPsNP_MAP_rep1$X2iP2AD.MAP, paired = TRUE)
```

```
##
##  Paired t-test
##
## data: TNPsNP_MAP_rep1$X2iP2A.MAP and TNPsNP_MAP_rep1$X2iP2AD.MAP
## t = 63.433, df = 8921, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.02734052 0.02908416
## sample estimates:
## mean of the differences
##                      0.02821234
```

#### 10. Line Plot for FoldChange of Specific Genes Between Condition

```
if (!require("ggplot2")) install.packages("ggplot2")
library(ggplot2)
if (!require("gplots")) install.packages("gplots")
library(gplots)
if (!require("curl")) install.packages("curl")
library("curl")
if (!require("tidyverse")) install.packages("tidyverse")
```

```

library("tidyverse")
if (!require("plyr")) install.packages("plyr")
library(plyr)
if (!require("dplyr")) install.packages("dplyr")
library(dplyr)
if (!require("viridis")) install.packages("viridis")
library(viridis)
if (!require("magrittr")) install.packages("magrittr")
library(magrittr)
if (!require("ggpubr")) install.packages("ggpubr")
library(ggpubr)
summarySE_median <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                               conf.interval=.95, .drop=TRUE) {
  library(plyr)
  # New version of length which can handle NA's: if na.rm==T, don't count them
  length2 <- function (x, na.rm=FALSE) {
    if (na.rm) sum(!is.na(x))
    else      length(x)
  }
  # This does the summary. For each group's data frame, return a vector with
  dataac <- ddply(data, groupvars, .drop=.drop, .fun = function(xx, col) {
    c(sum      = sum(as.numeric(as.character(xx[[col]]))),
      N       = length2(xx[[col]]),
      median = median(as.numeric(as.character(xx[[col]]))),    #replace the mean by median
      sd     = sd(as.numeric(as.character(xx[[col]]))))
  })
},
measurevar
)
# Rename the "mean" column
# dataac <- rename(dataac, c("mean" = measurevar))
dataac$se <- dataac$sd / sqrt(dataac$N)  # Calculate standard error of the mean
ciMult <- qt(conf.interval/2 + .5, dataac$N-1)
dataac$ci <- dataac$se * ciMult
return(dataac)
}
#data_msn2msn_forbar_CFP <- summarySE_median(data_CFP_L2, measurevar="CFP", groupvars=c("frame"))

summarySE_mean <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                           conf.interval=.95, .drop=TRUE) {
  library(plyr)
  # New version of length which can handle NA's: if na.rm==T, don't count them
  length2 <- function (x, na.rm=FALSE) {
    if (na.rm) sum(!is.na(x))
    else      length(x)
  }
  # This does the summary. For each group's data frame, return a vector with
  dataac <- ddply(data, groupvars, .drop=.drop,
                  .fun = function(xx, col) {
                    c(sum      = sum(as.numeric(as.character(xx[[col]]))),
                      mean    = mean(as.numeric(as.character(xx[[col]]))),    #replace the mean by median
                      sd     = sd(as.numeric(as.character(xx[[col]]))),
                      N      = length2(xx[[col]]))

```

```

        )
    },
    measurevar
)
# Rename the "mean" column
# dataac <- rename(dataac, c("mean" = measurevar))
dataac$se <- dataac$sd / sqrt(dataac$N) # Calculate standard error of the mean
ciMult <- qt(conf.interval/2 + .5, dataac$N-1)
dataac$ci <- dataac$se * ciMult
return(dataac)
}

total_MAP_211119 <- as.data.frame(read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\perturbation_exp_091"))
total_MAP_211205 <- as.data.frame(read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\perturbation_exp_091"))
TNPSiNP_MAP_rep1 <- as.data.frame(read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\perturbation_exp_091"))
#delete genes which deg == 0
keep.exprs1 <- rowSums(total_MAP_211119>0)>=ncol(total_MAP_211119)
total_MAP_211119 <- total_MAP_211119[keep.exprs1,]

keep.exprs2 <- rowSums(total_MAP_211205>0)>=ncol(total_MAP_211205)
total_MAP_211205 <- total_MAP_211205[keep.exprs2,]

keep.exprs3 <- rowSums(TNPSiNP_MAP_rep1>0)>=ncol(TNPSiNP_MAP_rep1)
TNPSiNP_MAP_rep1 <- TNPSiNP_MAP_rep1[keep.exprs3,]
total_TNPSiNP<-read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\perturbation_exp_0910\\SEQ220505_TNPSiNP")
#filter by Rcounts>50 >ncol/2 cpm(4.58)
total_TNPSiNP<-total_TNPSiNP[,c(1,2,3,7,8,9)]
keep.exprs_total <- rowSums(total_TNPSiNP>4.58)>=ncol(total_TNPSiNP)/2
total_TNPSiNP <- total_TNPSiNP[keep.exprs_total,]
TNPSiNP_MAP_rep1<-na.omit(TNPSiNP_MAP_rep1[rownames(total_TNPSiNP),])

fc_TNPSiNP_MAP_rep1<-as.data.frame(cbind(log2(TNPSiNP_MAP_rep1$X2iP2Asi.MAP/TNPSiNP_MAP_rep1$X2iP2A.MAP)
                                             log2(TNPSiNP_MAP_rep1$X2iP2AD.MAP/TNPSiNP_MAP_rep1$X2iP2A.MAP),
                                             log2(TNPSiNP_MAP_rep1$priP2A.MAP/TNPSiNP_MAP_rep1$X2iP2A.MAP),
                                             log2(TNPSiNP_MAP_rep1$priP2AD.MAP/TNPSiNP_MAP_rep1$priP2A.MAP),
                                             log2(TNPSiNP_MAP_rep1$priPDAsi.MAP/TNPSiNP_MAP_rep1$priP2A.MAP),
                                             log2(TNPSiNP_MAP_rep1$priPDAsi.MAP/TNPSiNP_MAP_rep1$X2iP2A.MAP),
                                             log2(TNPSiNP_MAP_rep1$priP2AD.MAP/TNPSiNP_MAP_rep1$X2iP2A.MAP),
                                             log2(TNPSiNP_MAP_rep1$X2iP2A.MAP/TNPSiNP_MAP_rep1$LP2A.MAP),
                                             log2(TNPSiNP_MAP_rep1$LP2AD.MAP/TNPSiNP_MAP_rep1$LP2A.MAP),
                                             log2(TNPSiNP_MAP_rep1$LP2SSi.MAP/TNPSiNP_MAP_rep1$LP2A.MAP)))
rownames(fc_TNPSiNP_MAP_rep1)<-rownames(TNPSiNP_MAP_rep1)
colnames(fc_TNPSiNP_MAP_rep1)<-c("fc2isi_2i","fc2iD_2i","fcpri_2i","fcpriD_pri",
                                    "fcprisi_pri","fcprisi_2i","fcpriD_2i","fc2i_lif","fc2i_lif","fc2i_lif",
#colnames(fc_TNPSiNP_MAP_rep1)<-c("a","b","c","d")

test1<-fc_TNPSiNP_MAP_rep1[fc_TNPSiNP_MAP_rep1$fcpri_2i>1,]
#colnames(fc_TNPSiNP_MAP_rep1)<-c("fc2isi_2i","fc2iD_2i","fcpri_2i","fcpriD_pri")
test2<-cbind(test1$fcpri_2i,test1$fcpriD_pri,test1$fcprisi_2i)
colnames(test2)<-c("a","b","c")
rownames(test2)<-rownames(test1)

```

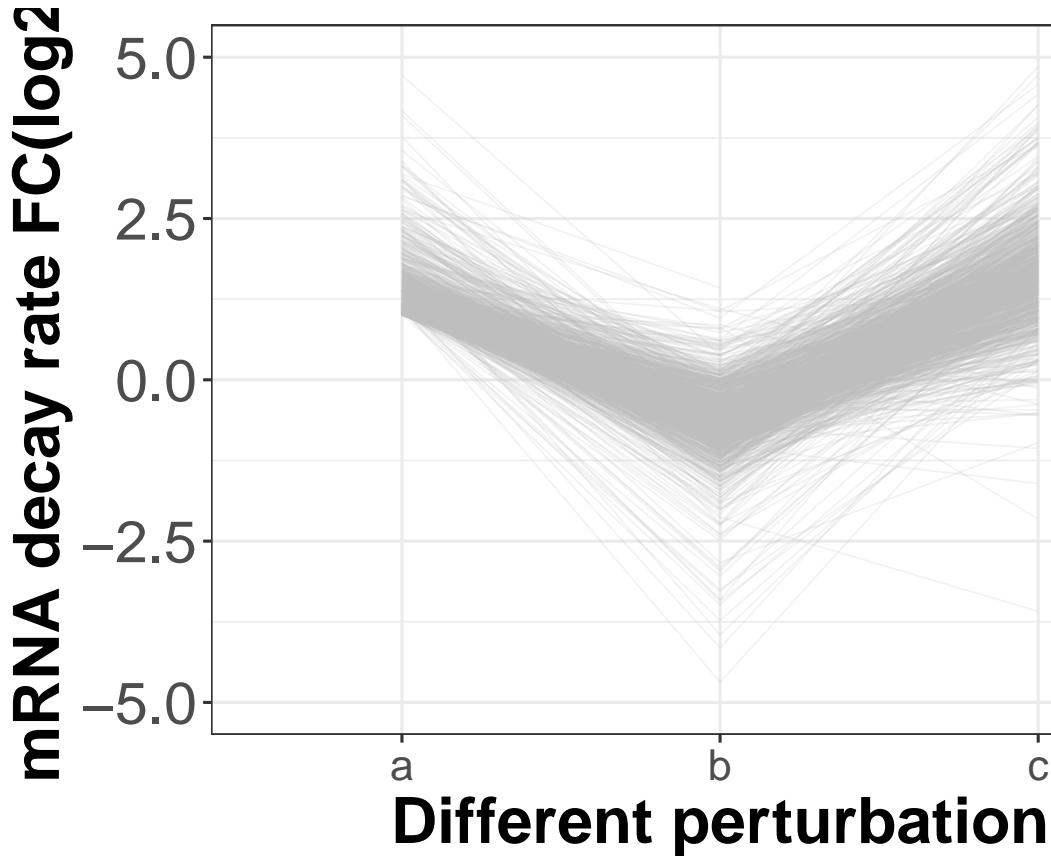
```

data_CFP_L = matrix(nrow=1,ncol=3)
for (n in 1:ncol(test2)) {
  data_CFP_N = as.matrix(test2[,n])
  data_CFP_NN = cbind(data_CFP_N,colnames(test2)[n],rownames(test2))
  data_CFP_L = rbind(data_CFP_L, data_CFP_NN)
}
data_CFP_L<-data_CFP_L[-1,]
colnames(data_CFP_L)<-c("fc","class","genes")
fc_TNPsiNP_MAP_draw <- as.data.frame(data_CFP_L)

p10_1<-ggplot(fc_TNPsiNP_MAP_draw,
  aes(x = class, y = as.numeric(as.character(fc)), group = genes))+ 
  geom_line(colour = "gray",size = 0.3,alpha=0.2) +
  labs(x = "Different perturbation", y = "mRNA decay rate FC(log2)")+
  ylim(-5,5) +
  theme_bw() +
  theme( plot.title = element_text(hjust = 0.5),
    text = element_text(size=22),
    title=element_text(size = 30),
    axis.title.x = element_text(size=24),
    axis.title.y = element_text(size=24),
    axis.text.x = element_text(size=16),
    axis.text.y = element_text(size=22),
    axis.title = element_text(face="bold"))
plot(p10_1)

```

## Warning: Removed 7 row(s) containing missing values (geom\_path).



```

##draw it as bar plot
pri_fc<-cbind(fc_TNPsINP_MAP_rep1$fcpri_2i,
                 fc_TNPsINP_MAP_rep1$fcpriD_2i,
                 fc_TNPsINP_MAP_rep1$fcprisi_2i)
rownames(pri_fc)<-rownames(TNPsINP_MAP_rep1)
colnames(pri_fc)<-c("fcpri_2i","fcpriD_2i","fcprisi_2i")
data_CFP_L = matrix(nrow=1,ncol=3)
for (n in 1:ncol(pri_fc)) {
  data_CFP_N = as.matrix(pri_fc[,n])
  data_CFP_NN = cbind(data_CFP_N,colnames(pri_fc)[n],rownames(pri_fc))
  data_CFP_L = rbind(data_CFP_L, data_CFP_NN)
}
data_CFP_L<-data_CFP_L[-1,]
colnames(data_CFP_L)<-c("fc","class","genes")
fc_pri_draw <- as.data.frame(data_CFP_L)

library(magrittr)
library(ggpubr)
#pri_fc
#8*8
p10_2<-ggplot(fc_pri_draw,aes(x = class, y = as.numeric(as.character(fc)), fill = class)) +
  # geom_violin(aes(fill = class),trim = F) +
  #geom_boxplot(alpha=0.7,outlier.shape = NA,outlier.colour = NA) +
  stat_boxplot(geom = 'errorbar', width = 0.1) +
  geom_boxplot(aes(fill = class), width = 0.2, show.legend = F,outlier.shape = NA,outlier.colour = NA) +
  #geom_jitter(aes(shape = class)) +

```

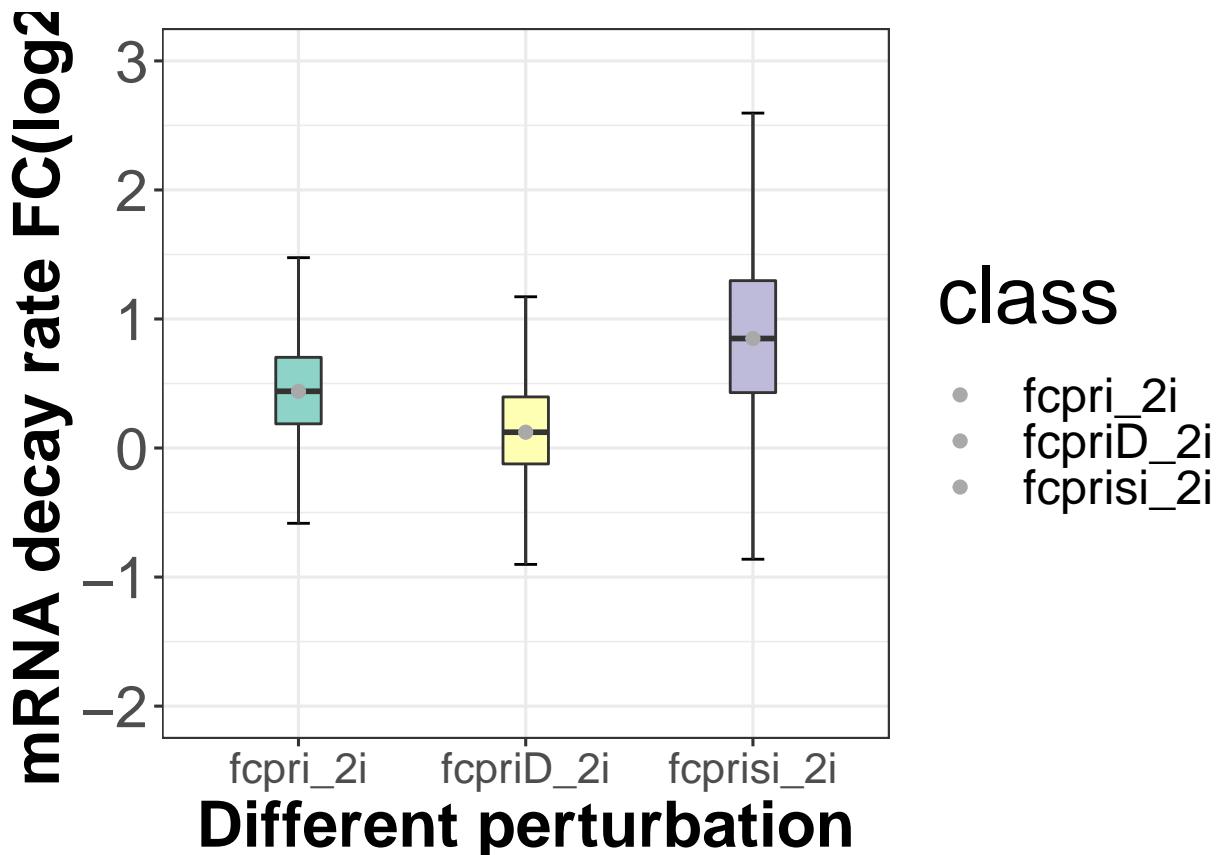
```

coord_trans(x = "identity", y = "identity", xlim = NULL, ylim = c(-2,3))+
stat_summary(fun = median, geom = 'point', color = 'dark grey', size = rel(2)) +
#guides(shape = F, color = F) +
labs(x = "Different perturbation", y = "mRNA decay rate FC(log2)")+
scale_fill_brewer(palette = 'Set3')+
theme_bw() +
theme( plot.title = element_text(hjust = 0.5),
      text = element_text(size=22),
      title=element_text(size = 30),
      axis.title.x = element_text(size=24),
      axis.title.y = element_text(size=24),
      axis.text.x = element_text(size=16),
      axis.text.y = element_text(size=22),
      axis.title = element_text(face="bold"))

```

## Coordinate system already present. Adding new coordinate system, which will replace the existing one

```
plot(p10_2)
```



11. Bin Dot Plot

```

if (!require("ggplot2")) install.packages("ggplot2")
library(ggplot2)
if (!require("gplots")) install.packages("gplots")
library(gplots)
if (!require("vegan")) install.packages("vegan")
library(vegan)
if (!require("permute")) install.packages("permute")
library(permute)
if (!require("ggdendro")) BiocManager::install("ggdendro", version = "3.8")
library("ggdendro")
if (!require("cowplot")) install.packages("cowplot")
library("cowplot")
if (!require("curl")) install.packages("curl")
library("curl")
if (!require("tidyverse")) install.packages("tidyverse")
library("tidyverse")
if (!require("ggpubr")) install.packages("ggpubr")
library("ggpubr")
if (!require("dplyr")) install.packages("dplyr")
library(dplyr)
if (!require("forcats")) install.packages("forcats")
library(forcats)
if (!require("reshape")) install.packages("reshape")
library(reshape)
if (!require("kohonen")) install.packages("kohonen")
library(kohonen)
if (!require("RColorBrewer")) install.packages("RColorBrewer")
library(RColorBrewer)
summarySE_median <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                               conf.interval=.95, .drop=TRUE) {
  library(plyr)
  # New version of length which can handle NA's: if na.rm==T, don't count them
  length2 <- function (x, na.rm=FALSE) {
    if (na.rm) sum(!is.na(x))
    else      length(x)
  }
  # This does the summary. For each group's data frame, return a vector with
  dataac <- ddply(data, groupvars, .drop=.drop,
                  .fun = function(xx, col) {
                    c(N      = length2(xx[[col]]),
                      mean = mean(as.numeric(as.character(xx[[col]]))),   #replace the mean by median
                      sd    = sd(as.numeric(as.character(xx[[col]]))))
                  })
  },
  measurevar
)
# Rename the "mean" column
dataac <- rename(dataac, c("mean" = measurevar))
dataac$se <- dataac$sd / sqrt(dataac$N)  # Calculate standard error of the mean
ciMult <- qt(conf.interval/2 + .5, dataac$N-1)
dataac$ci <- dataac$se * ciMult
return(dataac)
}

```

```

summarySE_mean <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                           conf.interval=.95, .drop=TRUE) {
  library(plyr)
  # New version of length which can handle NA's: if na.rm==T, don't count them
  length2 <- function (x, na.rm=FALSE) {
    if (na.rm) sum(!is.na(x))
    else      length(x)
  }
  # This does the summary. For each group's data frame, return a vector with
  dataac <- ddply(data, groupvars, .drop=.drop,
                  .fun = function(xx, col) {
                    c(N      = length2(xx[[col]]),
                      mean = mean(as.numeric(as.character(xx[[col]]))),    #replace the mean by median
                      sd   = sd(as.numeric(as.character(xx[[col]]))))
                  })
  },
  measurevar
)
# Rename the "mean" column
dataac <- rename(dataac, c("mean" = measurevar))
dataac$se <- dataac$sd / sqrt(dataac$N)  # Calculate standard error of the mean
ciMult <- qt(conf.interval/2 + .5, dataac$N-1)
dataac$ci <- dataac$se * ciMult
return(dataac)
}      #This is the basic function you need for summarizing the data, especially for error-bar plotting

total_cpm<-as.data.frame(read.csv("D:\\Linlab\\_article\\figure2\\ERT2_OCT4\\EO_191118\\trimmed_1227\\EO_191118<-as.data.frame(read.csv("D:\\Linlab\\_article\\figure2\\ERT2_OCT4\\EO_191118\\trimmed_1227\\EO_191118_MAP<-EO_191118[,c(grep("MAP",colnames(EO_191118)))]EO_191118_MAP<-cbind(EO_191118[,2],EO_191118_MAP)EO_191118_MAP<-na.omit(EO_191118_MAP)EO_191118_MAP<-EO_191118_MAP[!duplicated(EO_191118_MAP[,1]),]rownames(EO_191118_MAP)<-EO_191118_MAP[,1]
#EO_191118_RC<-EO_191118[,c(grep("Readcount",colnames(EO_191118)))]corr_EO<-read.csv("D:\\Linlab\\_article\\figure2\\ERT2_OCT4\\EO_191118\\trimmed_1227\\EO_lif_2i_corr.c
EO_191118_MAP<-na.omit(EO_191118_MAP[rownames(corr_EO),])

corr_DR_2i<-cbind(corr_EO[, "corr_2i"], EO_191118_MAP[, "X2i0.MAP"])
colnames(corr_DR_2i)<-c("corr_2i", "dr_2i")
corr_DR_lif<-cbind(corr_EO[, "corr_lif"], EO_191118_MAP[, "L0.MAP"])
colnames(corr_DR_lif)<-c("corr_lif", "dr_lif")
corr_DR_2i<-as.data.frame(corr_DR_2i)
corr_DR_lif<-as.data.frame(corr_DR_lif)

bin_2i<-corr_DR_2i %>% mutate(new_bin = cut(corr_2i, breaks=c(-1,-0.7,-0.4,0,0.4,0.7,1)))
bin_2i_sum<-summary_matrix1 <- as.data.frame(summarySE_mean(bin_2i, measurevar="corr_2i", groupvars=c("bin_2i"))
bin_2i_sum
```

	new_bin	N	corr_2i	sd	se	ci
## 1	(-1,-0.7]	54	-0.8018667	0.06830716	0.009295427	0.018644265
## 2	(-0.7,-0.4]	205	-0.5185489	0.08443650	0.005897301	0.011627477
## 3	(-0.4,0]	549	-0.1728316	0.11716359	0.005000420	0.009822337
## 4	(0,0.4]	1136	0.2237447	0.11426376	0.003390153	0.006651672

```

## 5  (0.4,0.7] 1767  0.5678771 0.08523259 0.002027624 0.003976796
## 6  (0.7,1] 3844  0.8685610 0.08383391 0.001352160 0.002651020

bin_2i_sum2<-summary_matrix1 <- as.data.frame(summarySE_mean(bin_2i, measurevar="dr_2i", groupvars=c("new_bin"))
bin_2i_sum2

##      new_bin     N    dr_2i        sd        se        ci
## 1  (-1,-0.7]   54  0.1524352 0.09736602 0.013249837 0.026575807
## 2 (-0.7,-0.4]  205  0.1861356 0.11546153 0.008064183 0.015899835
## 3  (-0.4,0]   549  0.1917128 0.11676731 0.004983507 0.009789115
## 4  (0,0.4]   1136  0.2135662 0.13052506 0.003872619 0.007598296
## 5  (0.4,0.7]  1767  0.2329694 0.14665611 0.003488847 0.006842704
## 6  (0.7,1]   3844  0.3176046 0.22194960 0.003579832 0.007018553

bin_lif<-corr_DR_lif %>% mutate(new_bin = cut(corr_lif, breaks=c(-1,-0.7,-0.4,0,0.4,0.7,1)))
bin_lif_sum<-summary_matrix1 <- as.data.frame(summarySE_mean(bin_lif, measurevar="corr_lif", groupvars=c("new_bin"))
bin_lif_sum

##      new_bin     N   corr_lif        sd        se        ci
## 1  (-1,-0.7]  196 -0.7939961 0.06423384 0.004588132 0.009048732
## 2 (-0.7,-0.4]  577 -0.5311128 0.08221017 0.003422454 0.006722012
## 3  (-0.4,0]   1294 -0.1856109 0.11443134 0.003181104 0.006240691
## 4  (0,0.4]   1987  0.2088388 0.11540070 0.002588866 0.005077178
## 5  (0.4,0.7]  1810  0.5549314 0.08586341 0.002018222 0.003958290
## 6  (0.7,1]   1691  0.8314459 0.07973032 0.001938883 0.003802865

bin_lif_sum2<-summary_matrix1 <- as.data.frame(summarySE_mean(bin_lif, measurevar="dr_lif", groupvars=c("new_bin"))
bin_lif_sum2

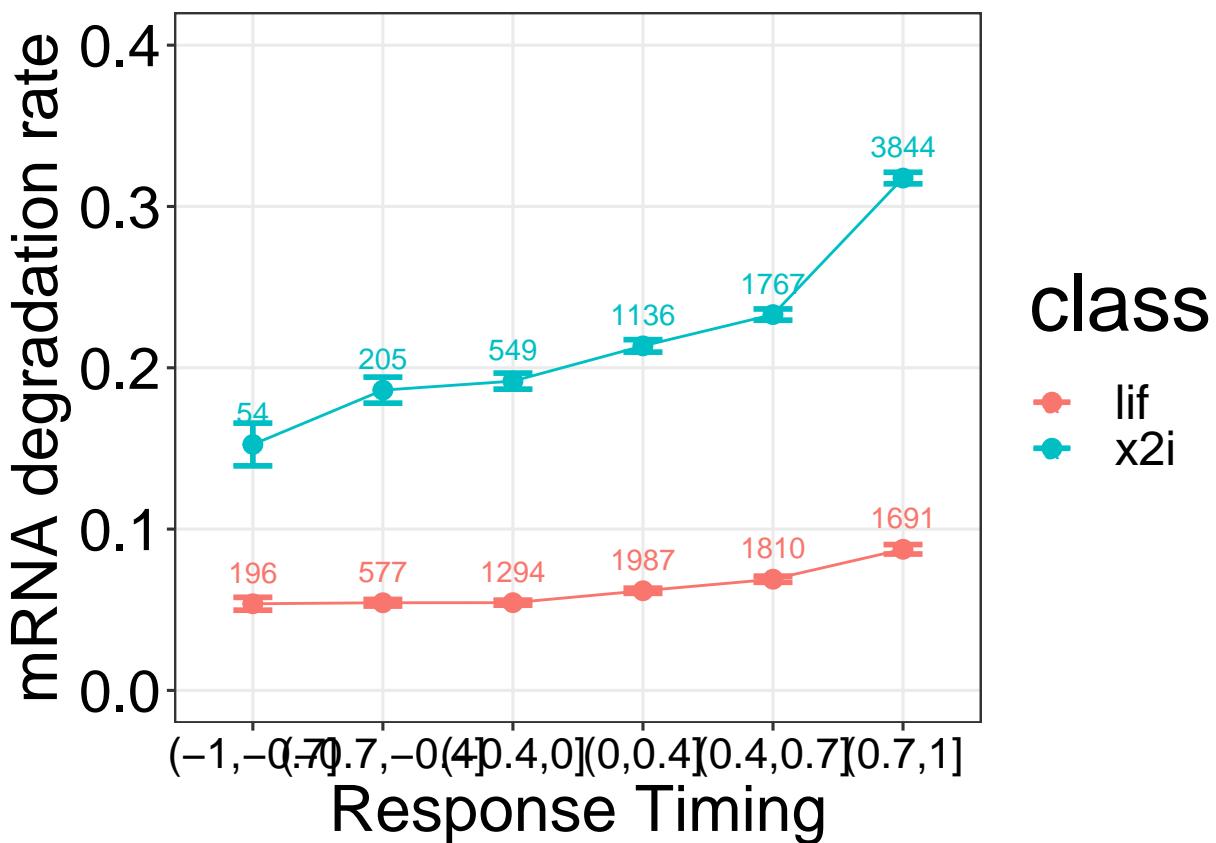
##      new_bin     N    dr_lif        sd        se        ci
## 1  (-1,-0.7]  196  0.05370306 0.05577943 0.003984245 0.007857745
## 2 (-0.7,-0.4]  577  0.05433068 0.04927022 0.002051146 0.004028638
## 3  (-0.4,0]   1294  0.05437295 0.05605784 0.001558365 0.003057201
## 4  (0,0.4]   1987  0.06176467 0.07127816 0.001599033 0.003135959
## 5  (0.4,0.7]  1810  0.06891337 0.08073286 0.001897628 0.003721773
## 6  (0.7,1]   1691  0.08746688 0.12084875 0.002938802 0.005764074

#p1<-ggplot(bin_2i_sum2, aes(x=new_bin, y=dr_lif, color= TF)) +
bin_lif_sum2<-cbind(bin_lif_sum2,"lif")
bin_2i_sum2<-cbind(bin_2i_sum2,"x2i")
colnames(bin_lif_sum2)<-colnames(bin_2i_sum2)
bin_final<-rbind(bin_2i_sum2,bin_lif_sum2)
colnames(bin_final)<-c("new_bin","N","dr","sd","se","ci","class")
#draw the dot plot
####size:8-6
p11<-ggplot(bin_final, aes(x=new_bin, y=dr, color= class,group= class,label = N)) +
  geom_point(size = 3) +
  geom_line(aes(color=class))+
  geom_errorbar(aes(ymin=dr-se, ymax=dr+se),
                width=.3,size=1)+
  geom_text(vjust = -1)+
```

```

scale_fill_brewer(palette = "Set3")+
labs(x = "Response Timing", y = "mRNA degradation rate")+
ylim(0,0.4)+
theme_bw()+
theme(
  text = element_text(size=22),
  panel.background = element_rect(fill = "transparent", colour = NA),
  panel.grid.minor = element_blank(),
  axis.text=element_text(color='black'),
  plot.title = element_text(hjust = 0.5),
  title=element_text(size = 30),
  axis.title.x = element_text(size=24),
  axis.title.y = element_text(size=24),
  axis.text.x = element_text(size=16),
  axis.text.y = element_text(size=22))
plot(p11)

```



12. PCA Dot Plot

```

if (!require("DESeq2")) BiocManager::install("DESeq2")
## Loading required package: DESeq2

```

```

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

## The following objects are masked from 'package:Biobase':
##      anyMissing, rowMedians

## The following object is masked from 'package:plyr':
##      count

## The following object is masked from 'package:dplyr':
##      count

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##      colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##      colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##      colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##      colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##      colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##      colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##      colWeightedMeans, colWeightedMedians, colWeightedSds,
##      colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##      rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##      rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##      rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##      rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##      rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##      rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##      rowWeightedSds, rowWeightedVars

## The following object is masked from 'package:Biobase':
##      rowMedians

##
## Attaching package: 'SummarizedExperiment'

```

```

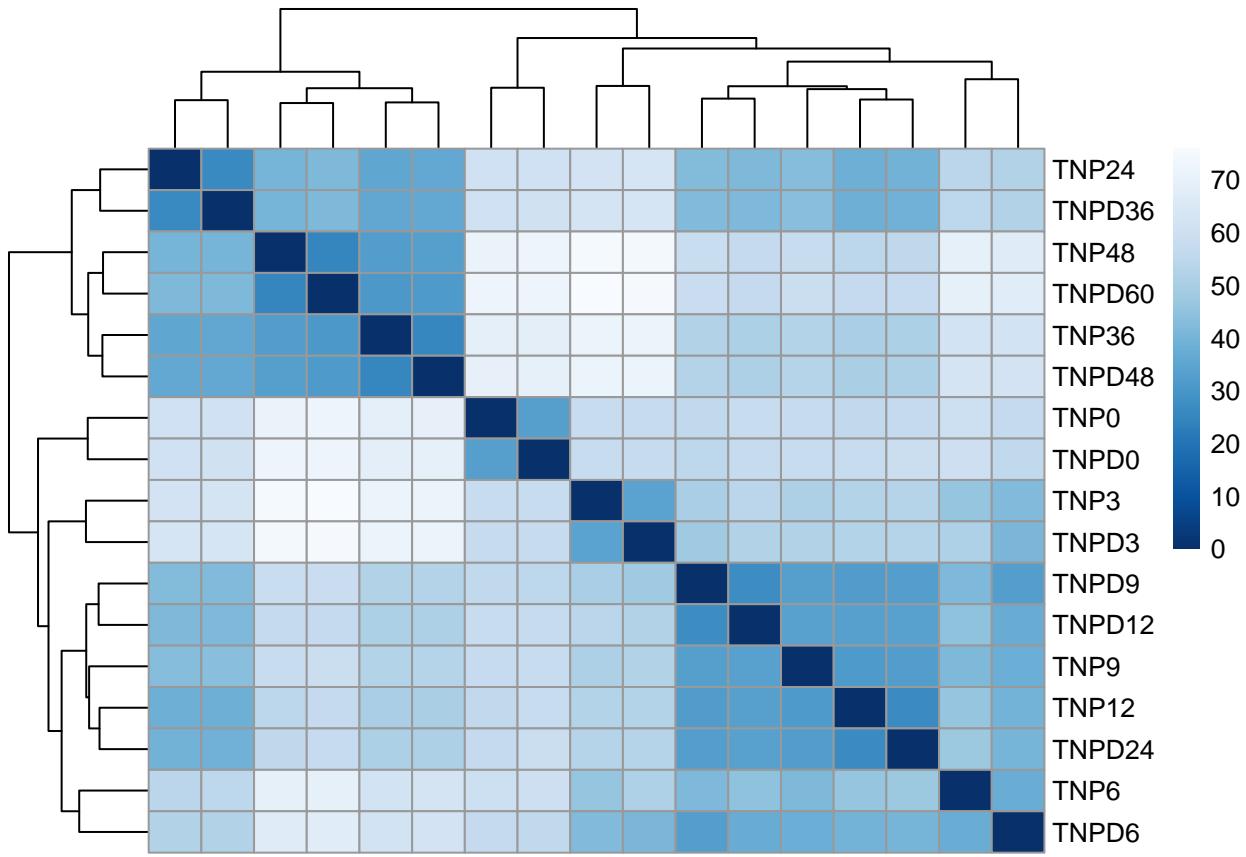
## The following object is masked from 'package:SeuratObject':
##
##      Assays

## The following object is masked from 'package:Seurat':
##
##      Assays

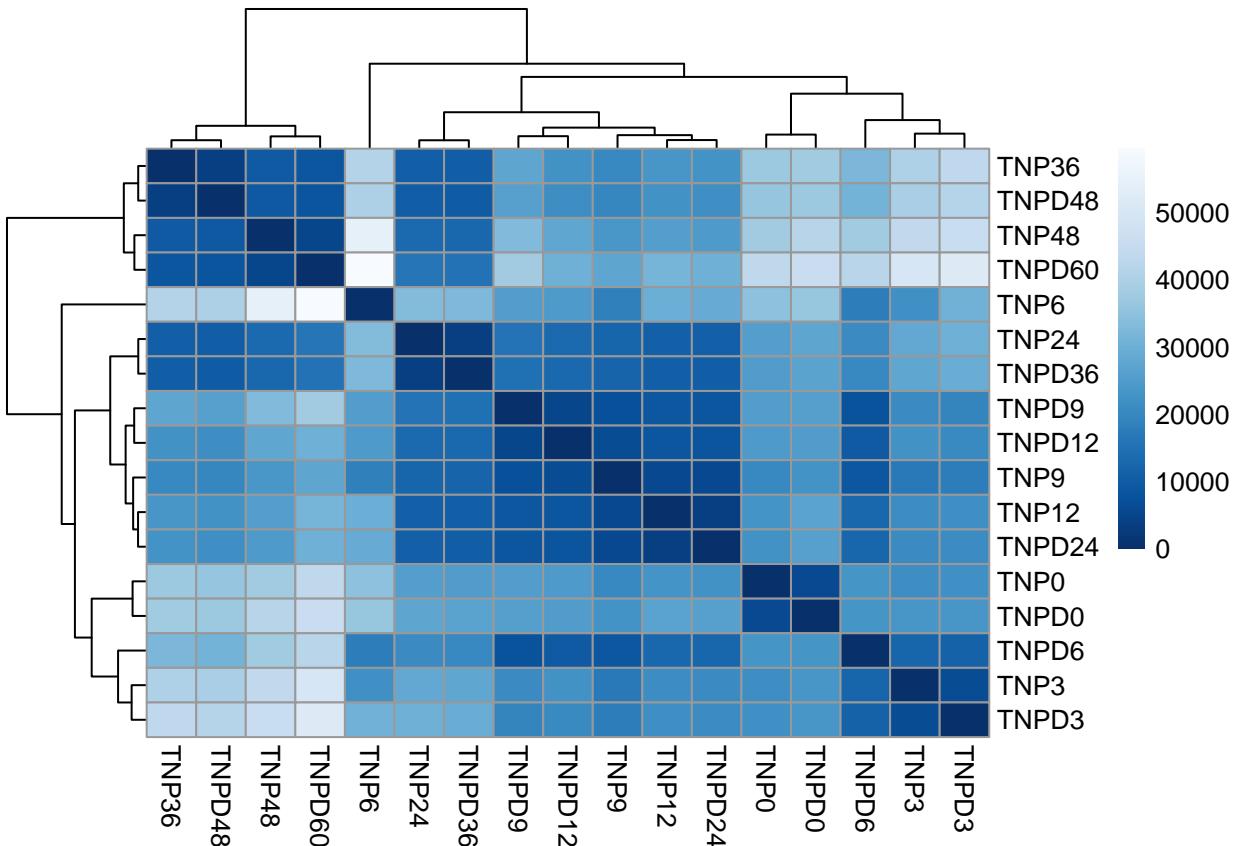
library("DESeq2")
if (!require("pheatmap")) BiocManager::install("pheatmap")
library("pheatmap")
dir = "D:\\Linlab\\OCT4 induction\\Function\\SEQ220117_TNP\\pca" ##this is the dir for saving data

NP_TNP<-read.csv("D:\\Linlab\\OCT4 induction\\Function\\SEQ220117_TNP\\matrix\\total_count.csv",row.names=1)
NP_TNP<-as.matrix(NP_TNP)
#We use the R function dist to calculate the Euclidean distance between samples.
#To ensure we have a roughly equal contribution from all genes, we use it on the VST data.
#We need to transpose the matrix of values using t, because the dist function expects the
#different samples to be rows of its argument, and different dimensions (here, genes) to be columns.
vsd <- vst(NP_TNP, blind = FALSE)
sampleDists <- dist(t(vsd))
library("pheatmap")
library("RColorBrewer")
sampleDistMatrix <- as.matrix( sampleDists )
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows = sampleDists,
         clustering_distance_cols = sampleDists,
         col = colors)

```



```
#Another option for calculating sample distances is to use the Poisson Distance (Witten 2011),
#implemented in the PoiClaClu package. This measure of dissimilarity between counts also
#takes the inherent variance structure of counts into consideration when calculating the
#distances between samples. The PoissonDistance function takes the original count matrix
#(not normalized) with samples as rows instead of columns, so we need to transpose the counts in dds.
library("PoiClaClu")
poisd <- PoissonDistance(t(NP_TNP))
samplePoisDistMatrix <- as.matrix(poisd$dd)
rownames(samplePoisDistMatrix) <- colnames(NP_TNP)
colnames(samplePoisDistMatrix) <- colnames(NP_TNP)
pheatmap(samplePoisDistMatrix,
         clustering_distance_rows = poisd$dd,
         clustering_distance_cols = poisd$dd,
         col = colors)
```



```
#####
#####read the csv count matrix and annotation matrix
#####
#####change it to DESeqDataSet

cts <- as.matrix(NP_TNP)
coldata <- read.csv("D:\\Linlab\\OCT4 induction\\Function\\SEQ220117_TNP\\pca\\anno_NP_all.csv")
rownames(coldata)<-coldata[,1]
coldata <- coldata[,c("condition","type")]
coldata$condition <- factor(coldata$condition)
coldata$type <- factor(coldata$type)

library("DESeq2")
minReplicateForReplace=Inf
dds <- DESeqDataSetFromMatrix(countData = cts,
                                colData = coldata,
                                design = ~ condition)
vsd <- vst(dds,blind = FALSE)
head(assay(vsd), 3)
```

```
##          TNPO    TNP3    TNP6    TNP9    TNP12    TNP24    TNP36
## ENSMUSG00000102693 10.7158 10.7158 10.7158 10.7158 10.7158 10.7158 10.71580
## ENSMUSG00000064842 10.7158 10.7158 10.7158 10.7158 10.7158 10.7158 10.71580
## ENSMUSG00000051951 10.7158 10.7158 10.7158 10.7158 10.7158 10.7158 10.76228
##          TNP48    TNPD0    TNPD3    TNPD6    TNPD9    TNPD12   TNPD24
## ENSMUSG00000102693 10.7158 10.7158 10.71580 10.71580 10.71580 10.7158 10.7158
## ENSMUSG00000064842 10.7158 10.7158 10.71580 10.71580 10.71580 10.7158 10.7158
```

```

## ENSMUSG00000051951 10.7158 10.7158 10.75462 10.76701 10.74587 10.7158 10.7158
##                                     TNPD36  TNPD48  TNPD60
## ENSMUSG00000102693 10.7158 10.7158 10.7158
## ENSMUSG00000064842 10.7158 10.7158 10.7158
## ENSMUSG00000051951 10.7158 10.7158 10.7158

```

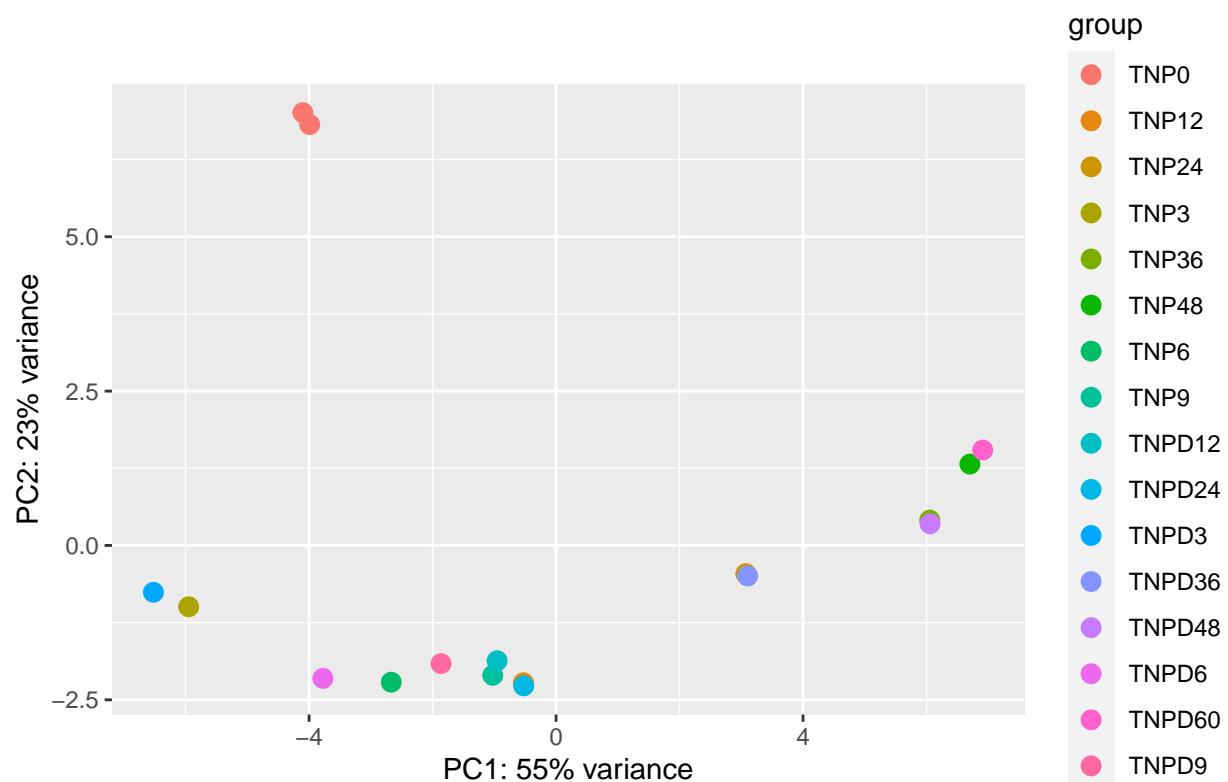
```
colData(vsd)
```

```

## DataFrame with 17 rows and 3 columns
##   condition      type sizeFactor
##   <factor> <factor> <numeric>
## TNP0       TNP0     PE150    0.738457
## TNP3       TNP3     PE150    0.771608
## TNP6       TNP6     PE150    1.544397
## TNP9       TNP9     PE150    0.699742
## TNP12      TNP12    PE150    1.018836
## ...        ...
## TNPD12     TNPD12   PE150    0.986349
## TNPD24     TNPD24   PE150    0.911123
## TNPD36     TNPD36   PE150    0.925719
## TNPD48     TNPD48   PE150    1.011904
## TNPD60     TNPD60   PE150    1.671248

```

```
plotPCA(vsd)
```



### 13. Correlation Heatmap

```
if (!require("DESeq2")) BiocManager::install("DESeq2")
library("DESeq2")
if (!require("pheatmap")) install.packages("pheatmap")

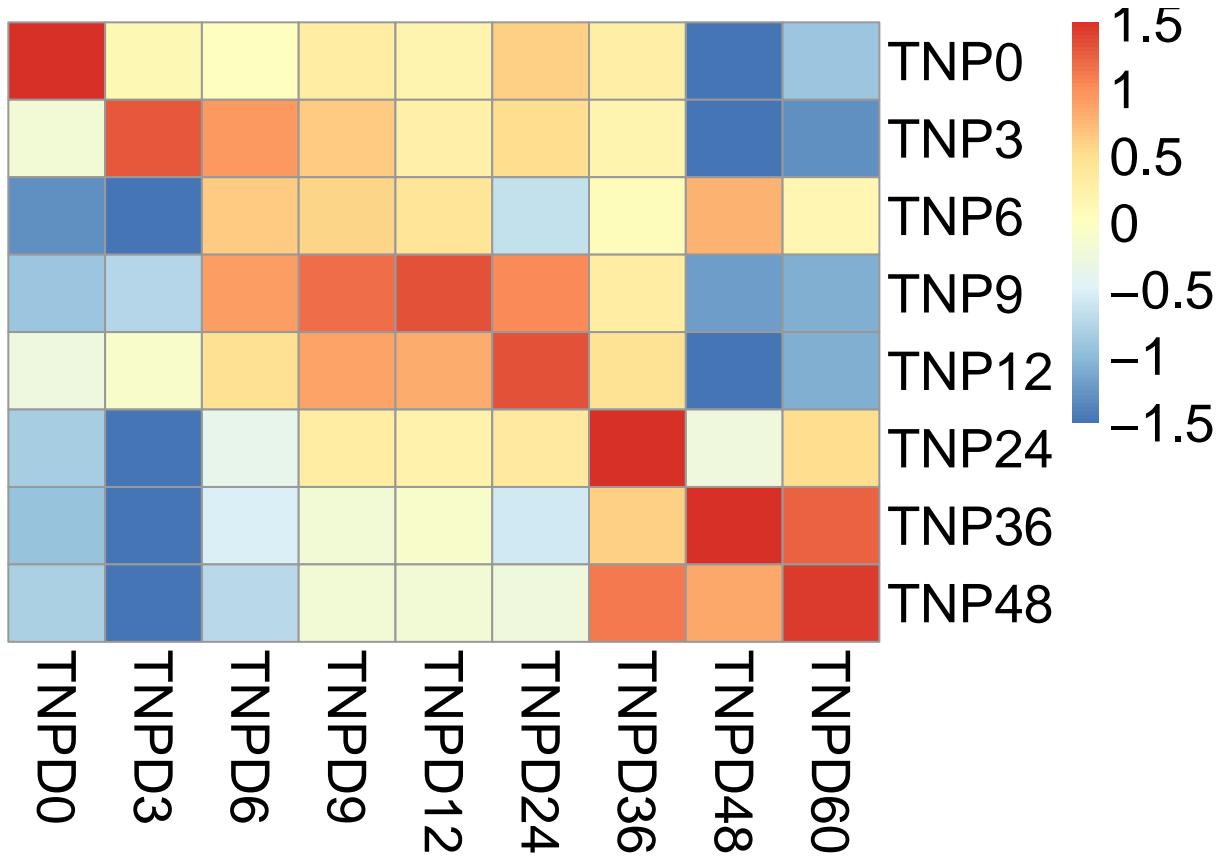
###calculate the pearson correlation of each condition:
all_marker_genes<-read.table("D:\\Linlab\\OCT4 induction\\Function\\SEQ220117_TNP\\pca\\marker_all.txt")
NP_TNP<-read.csv("D:\\Linlab\\OCT4 induction\\Function\\SEQ220117_TNP\\matrix\\total_matrix_gene_ID.csv")
NP_TNP<-as.data.frame(NP_TNP)
NP_TNP_marker<-na.omit(NP_TNP[,1])
cor_NP_TNP<-cor(NP_TNP_marker)
library("corrplot")

## corrplot 0.92 loaded

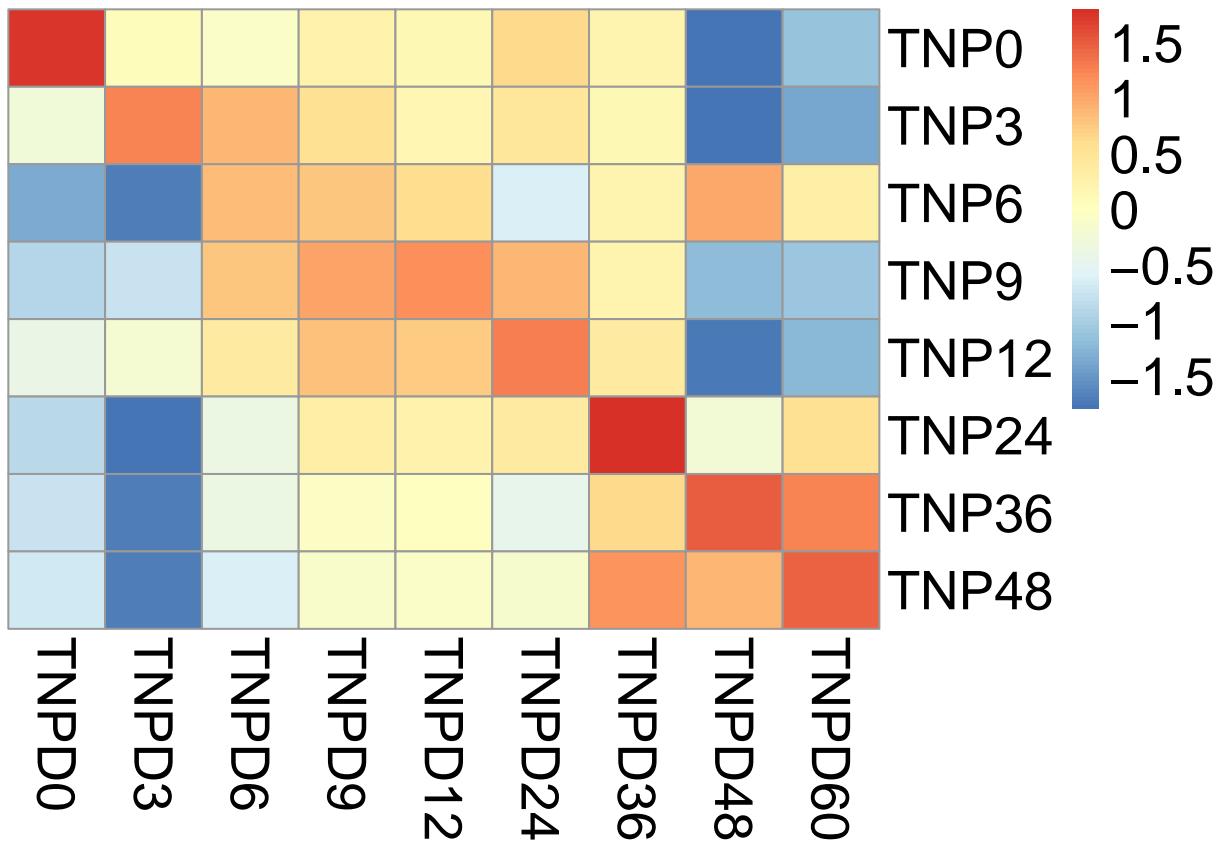
#corrplot corre_sc_gene_cor
#col=colorRampPalette(c("navy", "white", "firebrick3"))
#corrplot(corre_sc_gene_cor, add=TRUE, type="lower", method="number", cluster = TRUE, diag=FALSE, tl.pos="n")
#corrplot(corre_sc_gene_cor, type="upper", cluster = TRUE, col=col(10), tl.pos="d")
#library(RColorBrewer)square

#calculate the z-score for the correlation matrix:
scale_cor_NP_TNP <- round(t(apply(cor_NP_TNP, 1, scale)),2)
colnames(scale_cor_NP_TNP) <- colnames(cor_NP_TNP)

cor_NP_TNP_for_plot<-scale_cor_NP_TNP[c("TNPO", "TNP3", "TNP6", "TNP9", "TNP12", "TNP24", "TNP36", "TNP48"),
                                         c("TNP0", "TNP03", "TNP06", "TNP09", "TNP012", "TNP024", "TNP036", "TNP048"),
                                         c("TNP00", "TNP003", "TNP006", "TNP009", "TNP0012", "TNP0024", "TNP0036", "TNP0048")]
#corrplot(cor_NP_TNP_for_plot, title = "Correlation Plot", method = "circle", outline = T, addgrid.col =
bk = unique(c(seq(-1.5,1.5, length=100)))
#PDF:8:6
pheatmap(cor_NP_TNP_for_plot
         # annotation_col = annotation_col,
         ,breaks = bk
         # ,color = colorRampPalette(c("#ffffcc", "#41b6c4", "#225ea8"))(100)
         ,cluster_col= FALSE
         ,cluster_rows = FALSE
         ,show_colnames = T
         ,clustering_distance_rows = drows
         ,scale ="none"
         ,show_rownames= T,
         fontsize = 20) #change it to T if you want to show rownames
```



```
pheatmap(cor_NP_TNP_for_plot
  # annotation_col = annotation_col,
  #,breaks = bk
  # ,color = colorRampPalette(c("#ffffcc", "#41b6c4", "#225ea8"))(100)
  ,cluster_col= FALSE
  ,cluster_rows = FALSE
  ,show_colnames = T
  ,clustering_distance_rows = drows
  ,scale ="row"
  ,show_rownames= T,
  fontsize = 20) #change it to T if you want to show rownames
```



14. Pie Plot and Horizontal Bar Plot for GO Clusters

```

if (!require("ggplot2")) install.packages("ggplot2")
library(ggplot2)
if (!require("gplots")) install.packages("gplots")
library(gplots)
if (!require("vegan")) install.packages("vegan")
library(vegan)
if (!require("permute")) install.packages("permute")
library(permute)
if (!require("lattice")) install.packages("lattice")
library(lattice)
if (!require("limma")) install.packages("limma")
library(limma)
if (!require("ggdendro")) BiocManager::install("ggdendro", version = "3.8")
library("ggdendro")
if (!require("cowplot")) install.packages("cowplot")
library("cowplot")
if (!require("curl")) install.packages("curl")
library("curl")
if (!require("pheatmap")) install.packages("pheatmap")
library(dplyr)
if (!require("tidyverse")) install.packages("tidyverse")

```

```

library(tidyr)
if (!require("reshape2")) install.packages("reshape2")

## Loading required package: reshape2

##
## Attaching package: 'reshape2'

## The following objects are masked from 'package:reshape':
## 
##     colsplit, melt, recast

## The following object is masked from 'package:tidyr':
## 
##     smiths

library(reshape2)

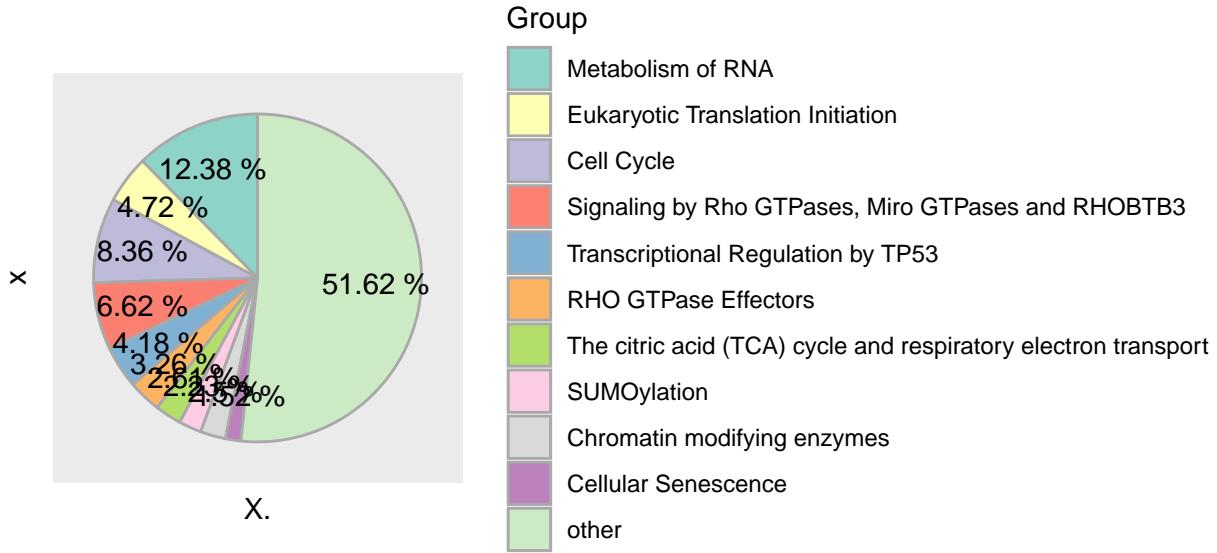
data<-read.csv("D:\\Linlab\\_article\\02_version1\\figure2\\04_go_plot\\go_2i_F48_P_allGENE_top10.csv")
data$Group <- factor(data>Description,levels = data>Description)
mylabel<-paste(data$X., "%")
mylabel<-rev(mylabel)
PercentS<-rev(data$X.)
#####size 8-6
p14_1<-ggplot(data,aes(x="",y=X.,fill=Group))+  

  geom_bar(stat = "identity",color="darkgrey")+
  scale_fill_brewer(palette="Set3")+
  coord_polar(theta = "y")+
  theme(axis.text.x = element_blank(),
        axis.ticks = element_blank(),
        panel.grid = element_blank())+
  #geom_text(aes(y=percent,x=1),label=mylabel)
  geom_text(aes(y=cumsum(PercentS)-PercentS/2,x=1.2),label=mylabel)

## Coordinate system already present. Adding new coordinate system, which will replace the existing one

plot(p14_1)

```



```
###12-5
data2<-data[order(data[, "Log10.P."], decreasing=T),]
data2$Description = factor(data2$Description, levels = data2$Description)
p14_2<-ggplot(data2[-1,], aes(x=Description, y=-as.numeric(as.character(Log10.P.)))) +
  geom_bar(stat="identity", position=position_dodge(), width = 0.8, fill="white", color="blue")+
  scale_fill_manual(values="#1a9850")+
  #scale_fill_brewer(palette="Set3")+
  # theme_minimal()+
  coord_flip()+
  labs(x = "Genes", y = "p-value (-log10 transformed)")+
  #scale_fill_manual(values="#1a9850")+
  theme_bw()+
  theme(
    text = element_text(size=22),
    panel.background = element_rect(fill = "transparent", colour = NA),
    panel.grid.minor = element_blank(),
    axis.text=element_text(color='black'),
    plot.title = element_text(hjust = 0.5),
    title=element_text(size = 30),
    axis.title.x = element_text(size=24),
    axis.title.y = element_text(size=24),
    axis.text.x = element_text(size=16),
    axis.text.y = element_text(size=22))
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```

```
plot(p14_2)
```

Metabolism  
Eukaryotic Translation Initiation by Rho GTPases, Miro GTPases and RHGTPases  
Transcriptional Regulation by RHO GTPase Effectors  
Citric acid (TCA) cycle and respiratory electron transport  
SUMOylation  
Chromatin modifying enzymes  
Cellular Senescence

p-value

---

Part II: Other refined pictures

1. Dot Edge Density Plot

```
if (!require("ggplot2")) install.packages("ggplot2")
library(ggplot2)
if (!require("gplots")) install.packages("gplots")
library(gplots)
if (!require("vegan")) install.packages("vegan")
library(vegan)
if (!require("permute")) install.packages("permute")
library(permute)
if (!require("lattice")) install.packages("lattice")
library(lattice)
if (!require("ggdendro")) BiocManager::install("ggdendro", version = "3.8")
library("ggdendro")
if (!require("cowplot")) install.packages("cowplot")
library("cowplot")
if (!require("curl")) install.packages("curl")
```

```

library("curl")
if (!require("ggdendro")) install.packages("ggdendro")
library("ggdendro")
if (!require("cowplot")) install.packages("cowplot")
library("cowplot")
if (!require("curl")) install.packages("curl")
library("curl")
if (!require("tidyverse")) install.packages("tidyverse")
library("tidyverse")
if (!require("ggpubr")) install.packages("ggpubr")
library("ggpubr")
if (!require("dplyr")) install.packages("dplyr")
library(dplyr)
if (!require("forcats")) install.packages("forcats")
library(forcats)
if (!require("reshape")) install.packages("reshape")
library(reshape)
if (!require("RColorBrewer")) install.packages("RColorBrewer")
library(RColorBrewer)

selected_MAP <- as.data.frame(read.csv("D:\\Linlab\\_article\\figure1\\dot_edge_density\\matrix_select"))
NP_total <- as.data.frame(read.csv("D:\\Linlab\\_article\\figure1\\dot_edge_density\\NP1106.csv",row.names=1))
L2i0a <- as.data.frame(read.csv("D:\\Linlab\\_article\\figure1\\dot_edge_density\\L2i0a.csv",row.names=1))

inte_genes<-Reduce(intersect, list(rownames(selected_MAP),
                                     rownames(NP_total),
                                     rownames(L2i0a)))

###make it a new matrix:x-y-tag
MAP<-selected_MAP[inte_genes,]
colnames(MAP)<-c("2i","LIF","mEpiLC")
MAP_new <- MAP %>% gather("2i","mEpiLC","LIF",
                           key='class',value='MAP')
MAP_new2 <-as.data.frame(MAP_new[,c("class","MAP")])
total_cpm <- cbind(NP_total[inte_genes,],L2i0a[inte_genes,])
colnames(total_cpm)<-c("2i","mEpiLC","LIF")
total_cpm_new <- total_cpm %>% gather(colnames(total_cpm),
                                         key='class',value='cpm_value')
total_cpm_new2 <-as.data.frame(total_cpm_new[,c("class","cpm_value")])
##integrate these two matrix by class:
matrix_total<-cbind(MAP_new2,total_cpm_new2)
matrix_total<-as.data.frame(matrix_total[,-1])

###the log(cpm)
attach(matrix_total)

## The following object is masked _by_ .GlobalEnv:
##
##      MAP

matrix_total$cpm_value <- log(cpm_value)
detach(matrix_total)
attach(matrix_total)

```

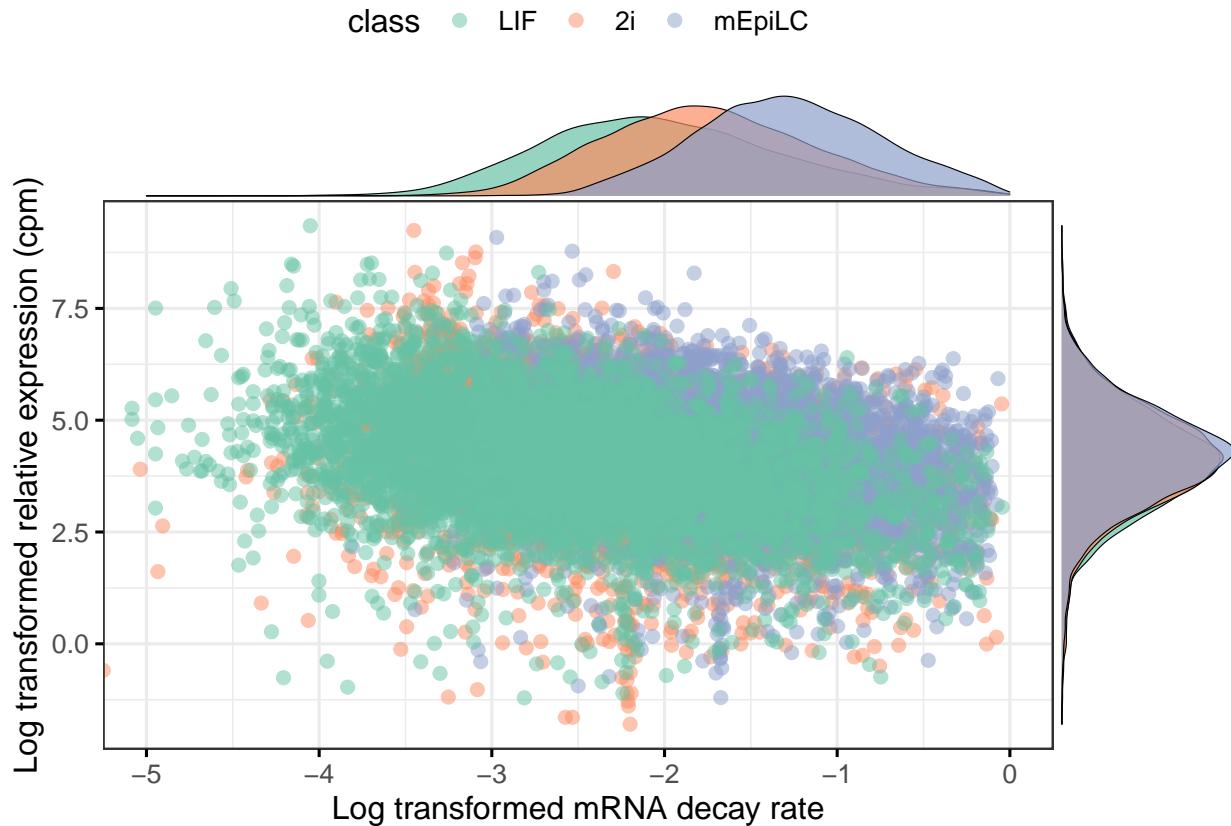
```

## The following object is masked _by_ .GlobalEnv:
##
##      MAP

matrix_total$MAP <- log(matrix_total$MAP)
detach(matrix_total)
##draw the density plot for three group:
matrix_total$class = with(matrix_total, reorder(class, MAP, median))
ggscatterhist(
  matrix_total, x = "MAP", y = "cpm_value",
  color = "class", size = 2, alpha = 0.5,
  palette = 'Set2',
  #palette = c("#00AFBB", "#E7B800", "#FC4E07"),
  margin.params = list(fill = "class", color = "black", size = 0.2),
  ggtheme = theme_bw(),
  xlab="Log transformed mRNA decay rate",
  ylab="Log transformed relative expression (cpm)",
  xlim=c(-5,0)
)

```

## Warning: Removed 1 rows containing non-finite values (stat\_density).



## 2. Bin Bar Plot

```
if (!require("ggplot2")) install.packages("ggplot2")
library(ggplot2)
if (!require("gplots")) install.packages("gplots")
library(gplots)
if (!require("vegan")) install.packages("vegan")
library(vegan)
if (!require("permute")) install.packages("permute")
library(permute)
if (!require("lattice")) install.packages("lattice")
library(lattice)
if (!require("ggdendro")) BiocManager::install("ggdendro", version = "3.8")
library("ggdendro")
if (!require("cowplot")) install.packages("cowplot")
library("cowplot")
if (!require("curl")) install.packages("curl")
library("curl")
if (!require("ggpubr")) install.packages("ggpubr")
library("ggpubr")
if (!require("dplyr")) install.packages("dplyr")
library(dplyr)
if (!require("forcats")) install.packages("forcats")
library(forcats)
if (!require("reshape")) install.packages("reshape")
library(reshape)
if (!require("kohonen")) install.packages("kohonen")
library(kohonen)
if (!require("RColorBrewer")) install.packages("RColorBrewer")
library(RColorBrewer)
summarySE_median <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                               conf.interval=.95, .drop=TRUE) {
  library(plyr)
  # New version of length which can handle NA's: if na.rm==T, don't count them
  length2 <- function (x, na.rm=FALSE) {
    if (na.rm) sum(!is.na(x))
    else      length(x)
  }
  # This does the summary. For each group's data frame, return a vector with
  dataac <- ddply(data, groupvars, .drop=.drop,
                  .fun = function(xx, col) {
                    c(N      = length2(xx[[col]]),
                      mean = mean(as.numeric(as.character(xx[[col]]))),    #replace the mean by median
                      sd    = sd(as.numeric(as.character(xx[[col]]))))
                  })
  },
  measurevar
)
# Rename the "mean" column
dataac <- rename(dataac, c("mean" = measurevar))
dataac$se <- dataac$sd / sqrt(dataac$N)  # Calculate standard error of the mean
ciMult <- qt(conf.interval/2 + .5, dataac$N-1)
dataac$ci <- dataac$se * ciMult
```

```

    return(dataac)
}

summarySE_mean <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                           conf.interval=.95, .drop=TRUE) {
  library(plyr)
  # New version of length which can handle NA's: if na.rm==T, don't count them
  length2 <- function (x, na.rm=FALSE) {
    if (na.rm) sum(!is.na(x))
    else      length(x)
  }
  # This does the summary. For each group's data frame, return a vector with
  dataac <- ddply(data, groupvars, .drop=.drop,
                  .fun = function(xx, col) {
                    c(N      = length2(xx[[col]]),
                      mean = mean(as.numeric(as.character(xx[[col]]))),   #replace the mean by median
                      sd    = sd(as.numeric(as.character(xx[[col]]))))
                  })
  },
  measurevar
)
# Rename the "mean" column
dataac <- rename(dataac, c("mean" = measurevar))
dataac$se <- dataac$sd / sqrt(dataac$N)  # Calculate standard error of the mean
ciMult <- qt(conf.interval/2 + .5, dataac$N-1)
dataac$ci <- dataac$se * ciMult
return(dataac)
}      #This is the basic function you need for summarizing the data, especially for error-bar plotting

integrate_all<-read.csv("D:\\Linlab\\OCT4 induction\\Function\\reanalysis_integrationALLE0_220908\\inte
nascent_cpm<-as.data.frame(cbind(integrate_all$s220128_E00.MAP*integrate_all$s220128_E00,
                                    integrate_all$s220128_E01.MAP*integrate_all$s220128_E01,
                                    integrate_all$s220128_E03.MAP*integrate_all$s220128_E03,
                                    integrate_all$s220128_E06.MAP*integrate_all$s220128_E06,
                                    integrate_all$s220128_E09.MAP*integrate_all$s220128_E09,
                                    integrate_all$s220128_E012.MAP*integrate_all$s220128_E012,
                                    integrate_all$s220716_EOPD_0.MAP*integrate_all$s220716_EOPD0,
                                    integrate_all$s220716_EOPD_1.MAP*integrate_all$s220716_EOPD1,
                                    integrate_all$s220716_EOPD_25.MAP*integrate_all$s220716_EOPD25,
                                    integrate_all$s220716_EOPD_6.MAP*integrate_all$s220716_EOPD6,
                                    integrate_all$s220716_EOPD_9.MAP*integrate_all$s220716_EOPD9,
                                    integrate_all$s220716_EOPD_12.MAP*integrate_all$s220716_EOPD12))
colnames(nascent_cpm)<-c("s220128_E00","s220128_E01","s220128_E03","s220128_E06","s220128_E09","s220128_E012","s220716_EOPD0","s220716_EOPD1","s220716_EOPD3","s220716_EOPD6","s220716_EOPD9")
rownames(nascent_cpm)<-rownames(integrate_all)
total_cpm2<-integrate_all[,c("s220128_E00","s220128_E01","s220128_E03","s220128_E06","s220128_E09","s220128_E012","s220716_EOPD0","s220716_EOPD1","s220716_EOPD25","s220716_EOPD6","s220716_EOPD9")]
colnames(total_cpm2)<-c("s220128_E00","s220128_E01","s220128_E03","s220128_E06","s220128_E09","s220128_E012","s220716_EOPD0","s220716_EOPD1","s220716_EOPD3","s220716_EOPD6","s220716_EOPD9")
#for the separated matrix:
total_E00128 <- total_cpm2[,c(1:6)]
nascent_E00128 <- nascent_cpm[,c(1:6)]

```

```

keep.exprs1 <- rowSums(nascent_E00128>1)>=2
nascent_E00128 <- nascent_E00128[keep.exprs1,]

total_EOPD <- total_cpm2[,c(7:12)]
nascent_EOPD <- nascent_cpm[,c(7:12)]
keep.exprs1 <- rowSums(nascent_EOPD>1)>=2
nascent_EOPD <- nascent_EOPD[keep.exprs1,]

genes<-intersect(rownames(nascent_EOPD),rownames(nascent_E00128))
total_E00128 <- total_E00128[genes,]
total_EOPD <- total_EOPD[genes,]
nascent_EOPD<-nascent_EOPD[genes,]
nascent_E00128<-nascent_E00128[genes,]
##replace 0 by 0.01
nascent_E00128<-replace(nascent_E00128, nascent_E00128==0, 0.01)
nascent_EOPD<-replace(nascent_EOPD, nascent_EOPD==0, 0.01)
total_EOPD<-replace(total_EOPD, total_EOPD==0, 0.01)
total_E00128<-replace(total_E00128, total_E00128==0, 0.01)

###calculate the fold change
##nascent E00128:
fcn_E00128_0<-log2(nascent_E00128[, "s220128_E00"] /nascent_E00128[, "s220128_E00"])
fcn_E00128_1<-log2(nascent_E00128[, "s220128_E01"] /nascent_E00128[, "s220128_E00"])
fcn_E00128_2<-log2(nascent_E00128[, "s220128_E03"] /nascent_E00128[, "s220128_E00"])
fcn_E00128_3<-log2(nascent_E00128[, "s220128_E06"] /nascent_E00128[, "s220128_E00"])
fcn_E00128_4<-log2(nascent_E00128[, "s220128_E09"] /nascent_E00128[, "s220128_E00"])
fcn_E00128_5<-log2(nascent_E00128[, "s220128_E012"] /nascent_E00128[, "s220128_E00"])
nascent_E00128_fc <- as.data.frame(cbind(fcn_E00128_0,fcn_E00128_1,fcn_E00128_2,fcn_E00128_3,fcn_E00128_4))

##draw the heatmap:
bk = unique(c(seq(-1,1, length=100)))
##nascent EOPD:
fcn_EOPD_0<-log2(nascent_EOPD[, "s220716_EOPD0"] /nascent_EOPD[, "s220716_EOPD0"])
fcn_EOPD_1<-log2(nascent_EOPD[, "s220716_EOPD1"] /nascent_EOPD[, "s220716_EOPD0"])
fcn_EOPD_2<-log2(nascent_EOPD[, "s220716_EOPD3"] /nascent_EOPD[, "s220716_EOPD0"])
fcn_EOPD_3<-log2(nascent_EOPD[, "s220716_EOPD6"] /nascent_EOPD[, "s220716_EOPD0"])
fcn_EOPD_4<-log2(nascent_EOPD[, "s220716_EOPD9"] /nascent_EOPD[, "s220716_EOPD0"])
fcn_EOPD_5<-log2(nascent_EOPD[, "s220716_EOPD12"] /nascent_EOPD[, "s220716_EOPD0"])
nascent_EOPD_fc <- as.data.frame(cbind(fcn_EOPD_0,fcn_EOPD_1,fcn_EOPD_2,fcn_EOPD_3,fcn_EOPD_4,fcn_EOPD_5))

##total E00128:
fc_E00128_0<-log2(total_E00128[, "s220128_E00"] /total_E00128[, "s220128_E00"])
fc_E00128_1<-log2(total_E00128[, "s220128_E01"] /total_E00128[, "s220128_E00"])
fc_E00128_2<-log2(total_E00128[, "s220128_E03"] /total_E00128[, "s220128_E00"])
fc_E00128_3<-log2(total_E00128[, "s220128_E06"] /total_E00128[, "s220128_E00"])
fc_E00128_4<-log2(total_E00128[, "s220128_E09"] /total_E00128[, "s220128_E00"])
fc_E00128_5<-log2(total_E00128[, "s220128_E012"] /total_E00128[, "s220128_E00"])
total_E00128_fc <- as.data.frame(cbind(fc_E00128_0,fc_E00128_1,fc_E00128_2,fc_E00128_3,fc_E00128_4,fc_E00128_5))

##total EOPD:
fc_EOPD_0<-log2(total_EOPD[, "s220716_EOPD0"] /total_EOPD[, "s220716_EOPD0"])
fc_EOPD_1<-log2(total_EOPD[, "s220716_EOPD1"] /total_EOPD[, "s220716_EOPD0"])
fc_EOPD_2<-log2(total_EOPD[, "s220716_EOPD3"] /total_EOPD[, "s220716_EOPD0"])
fc_EOPD_3<-log2(total_EOPD[, "s220716_EOPD6"] /total_EOPD[, "s220716_EOPD0"])

```

```

fc_EOPD_4<-log2(total_EOPD[, "s220716_EOPD9"]/total_EOPD[, "s220716_EOPD0"])
fc_EOPD_5<-log2(total_EOPD[, "s220716_EOPD12"]/total_EOPD[, "s220716_EOPD0"])
total_EOPD_fc <- as.data.frame(cbind(fc_EOPD_0,fc_EOPD_1,fc_EOPD_2,fc_EOPD_3,fc_EOPD_4,fc_EOPD_5))

total_fc<-cbind(nascent_E00128_fc, total_E00128_fc, nascent_EOPD_fc, total_EOPD_fc)

all_matrix_selected<-as.data.frame(cbind(fcn_E00128_1,fcn_EOPD_1))
##bin and draw the bar plot:
bin_E0<-all_matrix_selected %>% mutate(new_bin = cut(fcn_E00128_1, breaks=c(-11,-1,-0.5,0,0.5,1,20)))
bin_E0_sum<-summary_matrix1 <- as.data.frame(summarySE_mean(bin_E0, measurevar="fcn_E00128_1", groupvar=bin_E0))

##      new_bin     N fcn_E00128_1        sd        se        ci
## 1  (-11,-1]   123   -1.4134436 0.6688424 0.060307485 0.119384689
## 2  (-1,-0.5]   510   -0.6675374 0.1276110 0.005650710 0.011101587
## 3  (-0.5,0]   3447   -0.1894169 0.1290463 0.002197986 0.004309487
## 4   (0,0.5]   3056    0.1774189 0.1283172 0.002321177 0.004551226
## 5   (0.5,1]    340    0.6576570 0.1257240 0.006818339 0.013411580
## 6   (1,20]     57    1.3673229 0.8129706 0.107680580 0.215710123

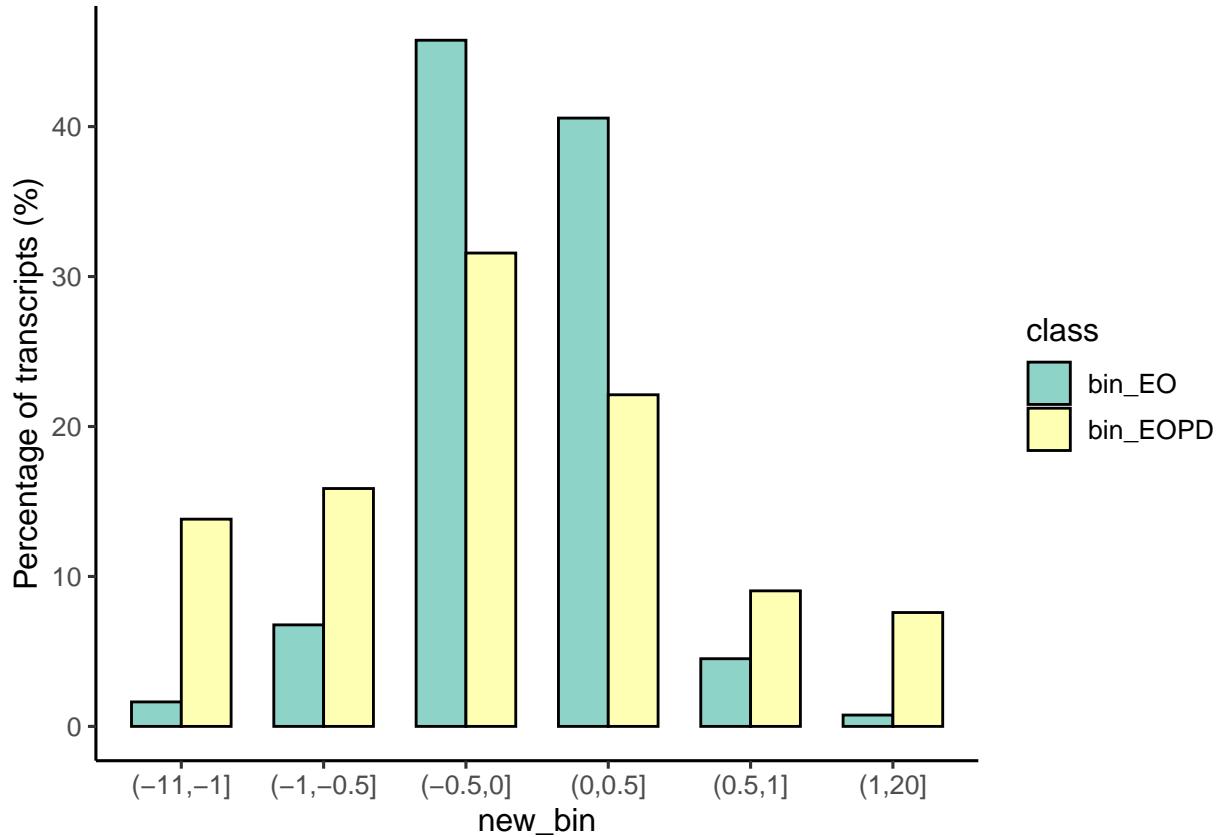
bin_EOPD<-all_matrix_selected %>% mutate(new_bin = cut(fcn_EOPD_1, breaks=c(-11,-1,-0.5,0,0.5,1,20)))
bin_EOPD_sum<-summary_matrix1 <- as.data.frame(summarySE_mean(bin_EOPD, measurevar="fcn_EOPD_1", groupvar=bin_EOPD))

##      new_bin     N fcn_EOPD_1        sd        se        ci
## 1  (-11,-1]  1041   -2.8053219 2.4321425 0.075381304 0.147916785
## 2  (-1,-0.5]  1195   -0.7134849 0.1426646 0.004126981 0.008096941
## 3  (-0.5,0]   2378   -0.2359149 0.1430127 0.002932707 0.005750929
## 4   (0,0.5]   1666    0.2171219 0.1421505 0.003482659 0.006830852
## 5   (0.5,1]    681    0.6994733 0.1423750 0.005455821 0.010712280
## 6   (1,20]    572    2.3555291 2.0164404 0.084311607 0.165598725

bin_final<-rbind(as.matrix(cbind(bin_E0_sum[,c(1,2)],"bin_E0")),
                  as.matrix(cbind(bin_EOPD_sum[,c(1,2)],"bin_EOPD")))
bin_final<-as.data.frame(bin_final)
bin_final[, "N"]<-c(bin_E0_sum[, "N"]*100/nrow(all_matrix_selected),bin_EOPD_sum[, "N"]*100/nrow(all_matrix_selected))
#bin_final2<-bin_final[c(1:5,8:19,22:28,)]
colnames(bin_final)<-c("new_bin","N","class")
bin_final$new_bin<-factor(bin_final$new_bin, levels = c('(-11,-1]', '(-1,-0.5]', '(-0.5,0]', '(0,0.5]', '(0.5,1]', '(1,20]'))
####save size 9-5
p22 <- ggplot(bin_final, aes( new_bin, weight=N,fill = class)) +
  #geom_hline(yintercept = seq(10, 50, 10), color = 'gray') +
  geom_bar(color = "black", width = .7, position = 'dodge') +
  labs( y = 'Percentage of transcripts (%)') +
  scale_fill_brewer(palette = "Set3")+
  #scale_y_continuous(expand = c(0,0)) +
  theme_classic()

plot(p22)

```



### 3. Bar Plot

```

library(ggplot2)
library(gplots)
library("ggdendro")
library("cowplot")
library("curl")

human<-as.data.frame(read.csv("D:\\Linlab\\DuPengLAB\\human_mouse_embryonic_early\\human.csv",row.names=1))
mouse<-as.data.frame(read.csv("D:\\Linlab\\DuPengLAB\\human_mouse_embryonic_early\\mouse.csv",row.names=1))

total_matrix_human<-cbind(t(human[,"PABPC1",]),colnames(human))
total_matrix_mouse<-cbind(t(mouse[,"Pabpc1",]),colnames(mouse))
colnames(total_matrix_mouse)<-colnames(total_matrix_human)
total_matrix_2_pab<-as.data.frame(rbind(total_matrix_human,total_matrix_mouse))
##draw the bar-plot
###draw the bar_plot
total_matrix_human<-as.data.frame(total_matrix_human)

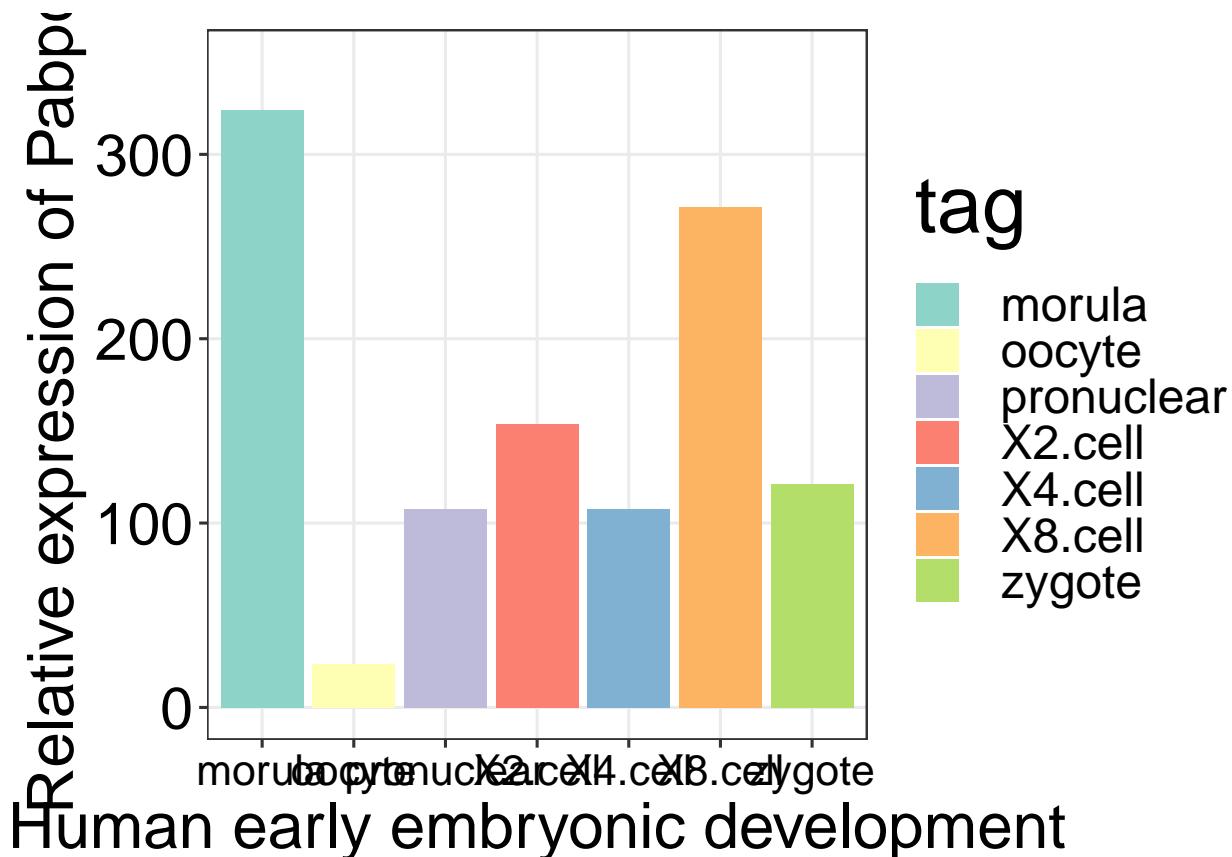
total_matrix_human<-as.data.frame(read.csv("D:\\Linlab\\DuPengLAB\\human_mouse_embryonic_early\\human_P",row.names=1))
total_matrix_mouse<-as.data.frame(read.csv("D:\\Linlab\\DuPengLAB\\human_mouse_embryonic_early\\mouse_P",row.names=1))
total_matrix_human$V2 = with(total_matrix_human, reorder(tag, PABPC1, mean)) ##order theclss by median
ggplot(data=total_matrix_human, aes(x=tag, y=as.numeric(as.character(PABPC1)), fill=tag)) +
  geom_bar(stat="identity", position=position_dodge())+

```

```

scale_fill_brewer(palette="Set3")+
ylim(0,350)+
# theme_minimal()+
labs(x = "Human early embryonic development", y = "Relative expression of Pabpc1")+
#scale_colour_manual(values=cc)+
theme_bw()+
theme(
  text = element_text(size=22),
  panel.background = element_rect(fill = "transparent", colour = NA),
  panel.grid.minor = element_blank(),
  axis.text=element_text(color='black'),
  plot.title = element_text(hjust = 0.5),
  title=element_text(size = 30),
  axis.title.x = element_text(size=24),
  axis.title.y = element_text(size=24),
  axis.text.x = element_text(size=16),
  axis.text.y = element_text(size=22))

```



#### 4. Violin Plot

```

library(ggplot2)
library(dplyr)
#library(easyGgplot2)

```

```

#library(easyGgplot2)
exon<-as.data.frame(read.csv("D:\\Linlab\\OCT4 induction\\database_datasets\\RNA-seq_INSPECT\\test_insp")
intron<-as.data.frame(read.csv("D:\\Linlab\\OCT4 induction\\database_datasets\\RNA-seq_INSPECT\\test_in
#rep1->d0_control rep2-->2i
#####for KO mESCs#####
rate_d0_DNMT1_ko<-intron[, "d0_DNMT1_ko"] / exon[, "d0_DNMT1_ko"]
rate_d0_DNMT1_ko[intron[, "d0_DNMT1_ko"] <= 0.2] <- NA
rate_d0_DNMT1_ko[exon[, "d0_DNMT1_ko"] <= 0.2] <- NA
rate_d0_DNMT1_ko<-as.data.frame(cbind(rate_d0_DNMT1_ko, "0"))
colnames(rate_d0_DNMT1_ko)<-c("value", "group")
rownames(rate_d0_DNMT1_ko)<-exon[, 1]

rate_d1_DNMT1_ko<-intron[, "d1_DNMT1_ko"] / exon[, "d1_DNMT1_ko"]
rate_d1_DNMT1_ko[intron[, "d1_DNMT1_ko"] <= 0.2] <- NA
rate_d1_DNMT1_ko[exon[, "d1_DNMT1_ko"] <= 0.2] <- NA
rate_d1_DNMT1_ko<-as.data.frame(cbind(rate_d1_DNMT1_ko, "1"))
colnames(rate_d1_DNMT1_ko)<-c("value", "group")
rownames(rate_d1_DNMT1_ko)<-exon[, 1]

rate_d3_DNMT1_ko<-intron[, "d3_DNMT1_ko"] / exon[, "d3_DNMT1_ko"]
rate_d3_DNMT1_ko[intron[, "d3_DNMT1_ko"] <= 0.2] <- NA
rate_d3_DNMT1_ko[exon[, "d3_DNMT1_ko"] <= 0.2] <- NA
rate_d3_DNMT1_ko<-as.data.frame(cbind(rate_d3_DNMT1_ko, "3"))
colnames(rate_d3_DNMT1_ko)<-c("value", "group")
rownames(rate_d3_DNMT1_ko)<-exon[, 1]

rate_d6_DNMT1_ko<-intron[, "d6_DNMT1_ko"] / exon[, "d6_DNMT1_ko"]
rate_d6_DNMT1_ko[intron[, "d6_DNMT1_ko"] <= 0.2] <- NA
rate_d6_DNMT1_ko[exon[, "d6_DNMT1_ko"] <= 0.2] <- NA
rate_d6_DNMT1_ko<-as.data.frame(cbind(rate_d6_DNMT1_ko, "6"))
colnames(rate_d6_DNMT1_ko)<-c("value", "group")
rownames(rate_d6_DNMT1_ko)<-exon[, 1]

rate_d9_DNMT1_ko<-intron[, "d9_DNMT1_ko"] / exon[, "d9_DNMT1_ko"]
rate_d9_DNMT1_ko[intron[, "d9_DNMT1_ko"] <= 0.2] <- NA
rate_d9_DNMT1_ko[exon[, "d9_DNMT1_ko"] <= 0.2] <- NA
rate_d9_DNMT1_ko<-as.data.frame(cbind(rate_d9_DNMT1_ko, "9"))
colnames(rate_d9_DNMT1_ko)<-c("value", "group")
rownames(rate_d9_DNMT1_ko)<-exon[, 1]

rate_d11_DNMT1_ko<-intron[, "d11_DNMT1_ko"] / exon[, "d11_DNMT1_ko"]
rate_d11_DNMT1_ko[intron[, "d11_DNMT1_ko"] <= 0.2] <- NA
rate_d11_DNMT1_ko[exon[, "d11_DNMT1_ko"] <= 0.2] <- NA
rate_d11_DNMT1_ko<-as.data.frame(cbind(rate_d11_DNMT1_ko, "99"))
colnames(rate_d11_DNMT1_ko)<-c("value", "group")
rownames(rate_d11_DNMT1_ko)<-exon[, 1]

rate_d17_DNMT1_ko<-intron[, "d17_DNMT1_ko"] / exon[, "d17_DNMT1_ko"]
rate_d17_DNMT1_ko[intron[, "d17_DNMT1_ko"] <= 0.2] <- NA
rate_d17_DNMT1_ko[exon[, "d17_DNMT1_ko"] <= 0.2] <- NA
rate_d17_DNMT1_ko<-as.data.frame(cbind(rate_d17_DNMT1_ko, "999"))
colnames(rate_d17_DNMT1_ko)<-c("value", "group")
rownames(rate_d17_DNMT1_ko)<-exon[, 1]

```

```

rate_d25_DNMT1_ko<-intron[, "d25_DNMT1_ko"] / exon[, "d25_DNMT1_ko"]
rate_d25_DNMT1_ko[intron[, "d25_DNMT1_ko"] <= 0.2] <- NA
rate_d25_DNMT1_ko[exon[, "d25_DNMT1_ko"] <= 0.2] <- NA
rate_d25_DNMT1_ko<-as.data.frame(cbind(rate_d25_DNMT1_ko, "9999"))
colnames(rate_d25_DNMT1_ko)<-c("value", "group")
rownames(rate_d25_DNMT1_ko)<-exon[,1]

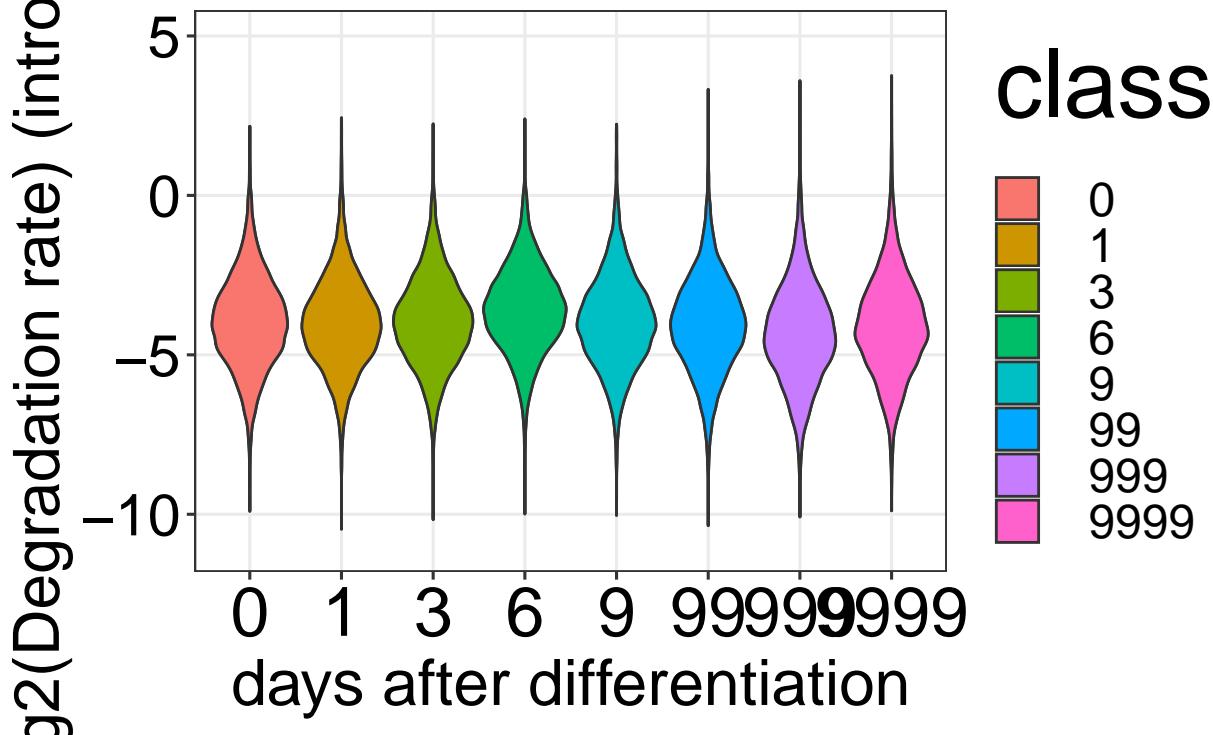
rate_total<-rbind(rate_d0_DNMT1_ko, rate_d1_DNMT1_ko, rate_d3_DNMT1_ko, rate_d6_DNMT1_ko, rate_d9_DNMT1_ko, rate_d12_DNMT1_ko)

rate_total2<-na.omit(rate_total)
rate_total2<-as.data.frame(rate_total2)
colnames(rate_total2)<-c("half_life", "class")

#violin plot:
#pdf(file="violin_plot_DNMT1_KO.pdf", width = 10, height = 10)
p24<-ggplot(rate_total2, aes(x = class, y = log2(as.numeric(as.character(half_life))))) +
  geom_violin() +
  ylim(-11,5) +
  ggtitle("DNMT1_KO mESCs") +
  geom_violin(aes(fill = class)) +
  labs(y = "log2(Degradation rate) (intron/exon)", x = "days after differentiation") +
  theme_bw() +
  theme(
    text = element_text(size=22),
    panel.background = element_rect(fill = "transparent", colour = NA),
    panel.grid.minor = element_blank(),
    axis.text=element_text(color='black'),
    plot.title = element_text(hjust = 0.5),
    title=element_text(size = 36),
    axis.title.x = element_text(size=24),
    axis.title.y = element_text(size=24),
    axis.text.x = element_text(size=26),
    axis.text.y = element_text(size=22))
plot(p24)

```

# DNMT1\_KO mESCs



```
#dev.off()
```

## 5. A Length Distribution and Uridylation

```
library("devtools")

## Loading required package: usethis

##
## Attaching package: 'devtools'

## The following object is masked from 'package:permute':
##       check

library("tidyverse")
library("ggplot2")

U_Oh<-read.table("D:\\Linlab\\OCT4 induction\\mechanism\\PIPA_seq\\PIPA-0916\\lyx1117\\ON68910_3Alength")
U_Oh<-as.data.frame(cbind(U_Oh,replace(U_Oh[,2], U_Oh[,2]>1, 1)))
colnames(U_Oh)<-c("A_length", "A_counts", "A_1")
U_6h<-read.table("D:\\Linlab\\OCT4 induction\\mechanism\\PIPA_seq\\PIPA-0916\\lyx1117\\ON68910_2Alength")
```

```

U_6h<-as.data.frame(cbind(U_6h,replace(U_6h[,2], U_6h[,2]>1, 1)))
colnames(U_6h)<-c("A_length","A_counts","A_1")

col_label=c("A_length","N","A_1","sd","se","ci","Condition")
###the A_1 is 100%
data_NP_0h <- summarySE_mean(U_0h, measurevar="A_1", groupvars=c("A_length"))

## Warning in qt(conf.interval/2 + 0.5, datac$N - 1):  NaNs

data_NP_0h1<-cbind(data_NP_0h,"txt_3")
colnames(data_NP_0h1)<-col_label
data_NP_6h <- summarySE_mean(U_6h, measurevar="A_1", groupvars=c("A_length"))

## Warning in qt(conf.interval/2 + 0.5, datac$N - 1):  NaNs

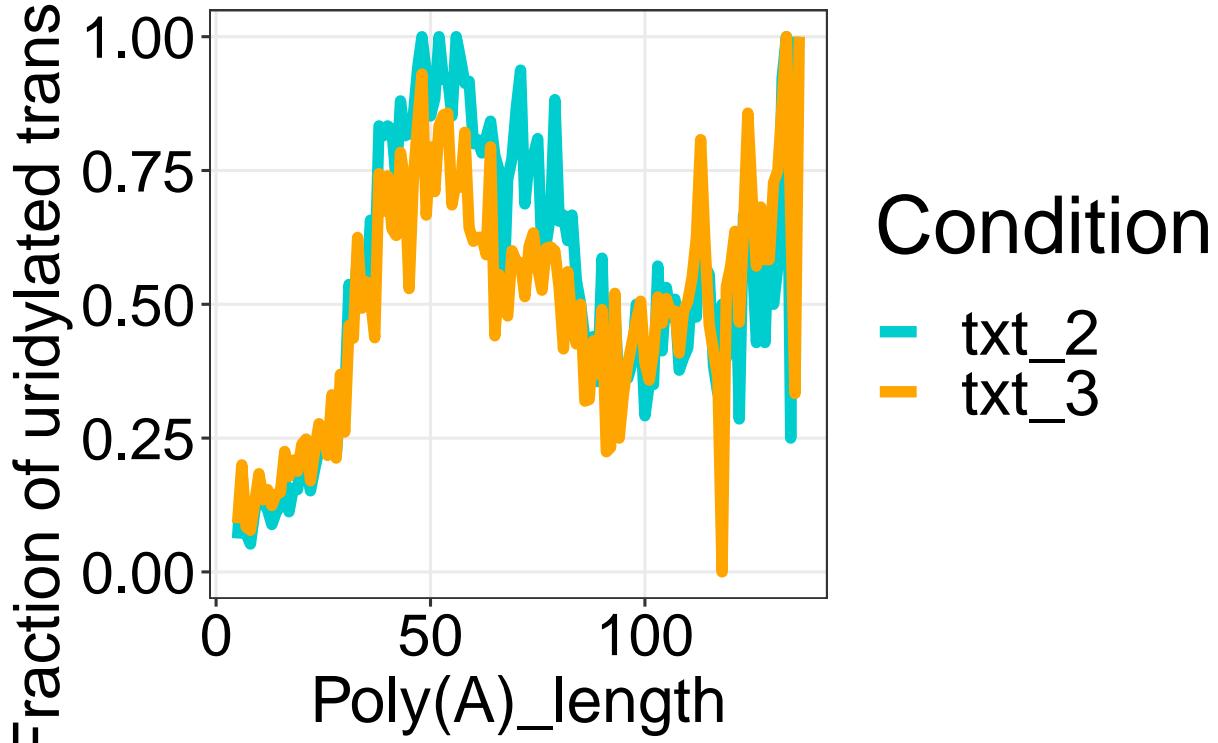
data_NP_6h1 <- cbind(data_NP_6h,"txt_2")
colnames(data_NP_6h1)<-col_label

data_msn2msn_forline<-as.data.frame(rbind(data_NP_0h1,data_NP_6h1))

cell_number = as.numeric(as.character(data_msn2msn_forline[1,"N"]))
#pdf(file =paste("Trajectory_",stress,"_",promoter,"_",strain,".pdf",collapse=NULL),width = 15,height =
cc<-c("cyan3","orange","Violet")
#ymaa=250
#ymii=min(as.numeric(as.character(data_msn2msn_forline[, "FP"])))
p25<-ggplot(data_msn2msn_forline, aes(x=A_length,y=A_1,group=Condition,col=Condition)) +
  geom_line(size=2)+
#  annotate("rect", xmin=responsetime, xmax=T_dur, ymin=ymii, ymax=ymaa, alpha=.1, fill="gray8")+
#  # annotate("rect", xmin=T_dur+0.5, xmax=movie_time, ymin=ymii, ymax=ymaa, alpha=.1, fill="dimgray")+
  scale_colour_manual(values=cc)+  #change the colour value of the curve
  labs(x = "Poly(A)_length", y = "Fraction of uridylated transcripts",
       title = "Uridylation frequency distribution")+
  theme_bw()+
  theme(
    text = element_text(size=30),
    panel.background = element_rect(fill = "transparent",colour = NA),
    panel.grid.minor = element_blank(),
    axis.text=element_text(color='black'),
    plot.title = element_text(hjust = 0.5),
    title=element_text(size = 30),
    axis.title.x = element_text(size=24),
    axis.title.y = element_text(size=24),
    axis.text.x = element_text(size=22),
    axis.text.y = element_text(size=22))
plot(p25)

```

# Transcription frequency distribution



6. Dot Plot for Last 6 Position

```

library("ggplot2")
less30nt_stat<-read.csv("D:\\Linlab\\OCT4 induction\\mechanism\\PIPA_seq\\PIPA-220214\\results\\last3\\")
less30nt_stat<-less30nt_stat[c("2iP2A","2iP2AD"),]
data_CFP_L = matrix(nrow=1,ncol=3)
for (n in 1:ncol(less30nt_stat)) {
  data_CFP_N = as.matrix(less30nt_stat[,n])
  data_CFP_NN = cbind(data_CFP_N,colnames(less30nt_stat)[n],rownames(less30nt_stat))
  data_CFP_L = rbind(data_CFP_L, data_CFP_NN)
}
data_CFP_L<-as.data.frame(data_CFP_L[-1,])
colnames(data_CFP_L)<-c("Ratio","Group","Condition")

ggplot(data=data_CFP_L) +
  geom_point(aes(x=Group, y=as.numeric(as.character(Ratio))*100,
                 shape=Condition, color=Condition,fill=Condition),size=6)+ 
  scale_shape_manual(values = c(15,16,17,18,19))+ 
  scale_fill_brewer(palette="Paired")+
  # theme_minimal()+
  labs(x = "Style", y = "Percent of uridylated transcripts")+
  #scale_colour_manual(values=cc)+
  theme_bw()

```

```

theme(
  text = element_text(size=22),
  panel.background = element_rect(fill = "transparent", colour = NA),
  panel.grid.minor = element_blank(),
  axis.text=element_text(color='black'),
  plot.title = element_text(hjust = 0.5),
  title=element_text(size = 30),
  axis.title.x = element_text(size=24),
  axis.title.y = element_text(size=24),
  axis.text.x = element_text(size=16),
  axis.text.y = element_text(size=22))

```

