**Bulk-correlation**

library(dplyr)

library(reshape2)

library(ggpubr)

library(extrafont)

######################################################

##########using t-SNE (non-linear) instead PCA(linear)

x <- read.table("Gene\_Expression\_CPM\_423.txt",header = T)

x2 <- read.table("Gene\_Expression\_CPM\_45.txt",header = T)

x3 <- merge(x,x2,by="gene",all = T)

x3[is.na(x3)] <- 0

geno1 <- read.table("genotype.txt")

geno2 <- read.table("genotype 2.txt")

geno <- rbind(geno1,geno2)

x4 <- melt(x3)

x5 <- merge(x4,geno,by.x="variable",by.y="V1")

x6 <- x5 %>% group\_by(V2,gene) %>% summarise(Mean\_CPM=mean(value))

x7 <- dcast(x6,gene~V2)

library(corrplot)

#col1 <- colorRampPalette(c("green","yellow","blue","white","red"))

corrplot(cor(x7[,c(2,3,6,15,16,7,9,14,5,11,8)],method = "pearson"),order = "original",cl.lim = c(0,1))

Bulk-Milk-markergenes

######################################################

##########using t-SNE (non-linear) instead PCA(linear)

x <- read.table("Count.txt",header = T)

rownames(x) <- x$geneid

#x <- x[,-1]

library(dplyr)

library(reshape2)

library(ggpubr)

library(extrafont)

x3 <- melt(x)

x4 <- x3 %>% group\_by(variable) %>% summarise(value2=sum(value))

x5 <- merge(x3,x4,by="variable")

x5$RPM <- x5$value \* 1000000 / x5$value2

x5 <- x5[,c(2,1,5)]

x5 <- dcast(x5, geneid~variable)

rownames(x5) <- x5$geneid

Gene\_Expression <- x5[,2:424]

Gene\_Expression <- Gene\_Expression[apply(Gene\_Expression, 1, mean) > 1,]

table1 <- Gene\_Expression[,c("B140024D0725","B110623D0725","B130244D0725.2","B140108D0725","B110530D0725","B111162D0725","B130296D0725","B130299D0725","B140113D0725","B121578D0725","B130082D0725","B130283D0725","B14133D1226","B180556D0831","B180504D0831","B160252D0831","B160009D0901","B160331D0901","B160249D0827","B180257D0827","B180490D0827","B180548D0827","B180791D0827","B180855D0827","B170224D0901",

"B170396D0901","B170431D0901","B170810D0901","B170954D0901","B180212D0901","B180309D0901","B180462D0901","B180522D0901","B160056D0729","B170682D0729","B180495D0729","B15109D1225","B17151D0106","B18134D0106","B18289D0106","B16139D0712","B17285D0712","B17186D0712","B15218D0712","B14147D0712","B17136D0712","B17199D0712","B17081D0712","B17254D0712","B17092D0712","B15274D0712",

"B16111D0712","B17190D0712","B15191D0712","B17258D0712","B17214D0712","B17233D0712","B17154D0712","B17219D0712","B17184D0712","B17091D0712","B17325D0712","B14124D0712","B16128D0712","B15108D0712","B17225D0712","B17172D0712","B15245D0712","B17276D0712","B17164D0712","B17167D0712","B17086D0712","B15097D0712","B15212D0712","B17189D0712","B15122D0712","B17185D0712","B17087D0712",

"B17339D0712","B17337D0712","B16054D0712","B16150D0712","B17150D0712","B17331D0712","B16050D0712","B74datD0712","B17201D0712","B17283D0712","B17266D0712","B16076D0712","B16132D0712","B17314D0712","B15107D0712","B17263D0712","B15286D0712","B17212D0712","B17192D0712","B16139D0523","B17285D0523","B17186D0523","B15218D0523","B14147D0523","B17136D0523","B17199D0523","B17081D0523",

"B17254D0523","B17195D0523","B15274D0523","B16111D0523","B17324D0523","B15273D0523","B15191D0523","B17258D0523","B17214D0523","B17233D0523","B17154D0523","B17219D0523","B17184D0523","B17091D0523","B17325D0523","B14124D0523","B15108D0523","B17326D0523","B17001D0523","B17225D0523","B17172D0523","B15245D0523","B17276D0523","B17167D0523","B17086D0523","B15097D0523","B17189D0523",

"B17185D0523","B17087D0523","B17339D0523","B17117D0523","B17337D0523","B16152D0523","B17309D0523","B16054D0523","B16150D0523","B17150D0523","B17187D0523","B17331D0523","B16244D0523","B17201D0523","B17283D0523","B17266D0523","B16076D0523","B16132D0523","B15258D0523","B17314D0523","B15286D0523","B17212D0523","B17362D0523","B17192D0523","B17190D0524","B16050D0524","B16128D0620m",

"B17326D0620m","B17164D0620m","B17164D0620n","B17192D0620m","B17192D0620e","B150195D0725","B150543D0725","B150625D0725","B160085D0725","B160205D0725","B160578D0725","B160847D0725","B170511D0725")]

table2 <- Gene\_Expression[,c("B170631D0725","B170678D0725","B170688D0725","B170821D0725","B180085D0725","B150235D0725","B150377D0725","B150420D0725","B160076D0725","B160254D0725","B160286D0725","B160544D0725","B160580D0725","B160601D0725","B160626D0725","B160720D0725","B160783D0725","B160858D0725","B160895D0725","B170012D0725","B170063D0725","B170506D0725","B170568D0725","B170620D0725","B170628D0725","B170718D0725",

"B170728D0725","B170738D0725","B170778D0725","B170855D0725","B170896D0725","B170916D0725","B170962D0725","B170985D0725","B180011D0725","B180015D0725","B160677D0725","B160796D0725","B160862D0725","B170519D0725","B170546D0725","B170658D0725","B17117D0620m","B17117D0620e","B17147","B17176","B17268","B17360","B18035","B18091","B18109","B15220D0625","B15274D0625","B17035D0625","B18035D0625",

"B19013D0625","B15273D1225","B16098D0106","B16139D0106","B16139D1225","B16252D0106","B17114D1225","B17131D1225","B17195D0106","B17354D1225","B18055D1225","B18091D1225","B18096D0106","B18112D0106","B18267D1225","B19101D0106","B19204D1225","B160206D0830","B160846D0830.1","B170033D0830","B170084D0830","B170455D0830","B170881D0830","B180160D0830","B180181D0830","B180229D0830","B180394D0830",

"B180396D0830","B180414D0830","B180432D0830","B180433D0830","B180466D0830","B160846D0830.2","B16139","B17186","B15218","B17081","B17254","B14156","B17195","B17092","B17205","B16111","B17190","B17258","B17233","B17184","B17172","B17167","B17086","B15097","B17189","B17185","B17337","B16152","B17309","B17187","B16050","B17283","B17266","B16132","B15258","B15286","B17212","B180057D0828","B180061D0828",

"B180075D0828","B180096D0828","B15322D0530","B16064D0530","B16173D0530","B17119D0530","B17129D0530","B17333D0530","B18015D0530","B16191D0530","B17103D0530","B18028D0530","B16125D0530","B18038D0530","B16020D0530","B16096D0530","B16265D0530","B17231D0530","B15181D0530","B15313D0530","B16101D0530","B16170D0530","B17030D0530","B17035D0530","B17101D0530","B150513D0725","B150524D0725","B160042D0725",

"B160173D0725","B160586D0725","B170374D0725","B170381D0725","B170683D0725","B170736D0725","B170823D0725","B170895D0725","B170948D0725","B170992D0725","B180008D0725","B180024D0725","B180038D0725","B180061D0725","B180096D0725","B180608D0725","B16257D0625","B17103D0625")]

table3 <- Gene\_Expression[,c("B17285D0620e","B17199D0620e","B17339D0620e","B17283D0620e","B16132D0620e","B150524D0828E","B160042D0828E","B160132D0828E","B160173D0828E","B170193D0828E","B170477D0828E","B170524D0828E","B170895D0828E","B170948D0828E","B170992D0828E","B180008D0828E","B180024D0828E","B180038D0828E","B180057D0828E","B180061D0828E","B150513D0729E","B150524D0729E","B160042D0729E","B170077D0729E","B170193D0729E",

"B170992D0729E","B180096D0729E","B180608D0729E","B17285D0620m","B17199D0620m","B17339D0620m","B17283D0620m","B16132D0620m","B150524D0828M","B160042D0828M","B160132D0828M","B160173D0828M","B170193D0828M","B170477D0828M","B170524D0828M","B170895D0828M","B170948D0828M","B170992D0828M","B180008D0828M","B180024D0828M","B180038D0828M","B180057D0828M","B180061D0828M","B150513D0729M","B150524D0729M",

"B160042D0729M","B170077D0729M","B170193D0729M","B170992D0729M","B180096D0729M","B180608D0729M","B17285D0620n","B17199D0620n","B17339D0620n","B17283D0620n","B16132D0620n","B150524D0828N","B160042D0828N","B160132D0828N","B160173D0828N","B170193D0828N","B170477D0828N","B170524D0828N","B170895D0828N","B170948D0828N","B170992D0828N","B180008D0828N","B180024D0828N","B180038D0828N","B180057D0828N",

"B180061D0828N","B150513D0729N","B150524D0729N","B160042D0729N","B170077D0729N","B170193D0729N","B170992D0729N","B180096D0729N","B180608D0729N")]

table\_group <- cbind(table1,table2,table3)

write.table(table\_group,"Gene\_Expression\_CPM\_423.txt",col.names = T,row.names = T,sep="\t",quote=F)

#################

Gene\_Expression\_log<-log2(Gene\_Expression+0.25)

write.table(Gene\_Expression\_log,"Gene\_Expression\_log.txt",col.names = T,row.names = T,sep="\t",quote=F)

#genotype <- read.table("genotype.txt")

table1 <- Gene\_Expression\_log[,c("B140024D0725","B110623D0725","B130244D0725.2","B140108D0725","B110530D0725","B111162D0725","B130296D0725","B130299D0725","B140113D0725","B121578D0725","B130082D0725","B130283D0725","B14133D1226","B180556D0831","B180504D0831","B160252D0831","B160009D0901","B160331D0901","B160249D0827","B180257D0827","B180490D0827","B180548D0827","B180791D0827","B180855D0827","B170224D0901",

"B170396D0901","B170431D0901","B170810D0901","B170954D0901","B180212D0901","B180309D0901","B180462D0901","B180522D0901","B160056D0729","B170682D0729","B180495D0729","B15109D1225","B17151D0106","B18134D0106","B18289D0106","B16139D0712","B17285D0712","B17186D0712","B15218D0712","B14147D0712","B17136D0712","B17199D0712","B17081D0712","B17254D0712","B17092D0712","B15274D0712",

"B16111D0712","B17190D0712","B15191D0712","B17258D0712","B17214D0712","B17233D0712","B17154D0712","B17219D0712","B17184D0712","B17091D0712","B17325D0712","B14124D0712","B16128D0712","B15108D0712","B17225D0712","B17172D0712","B15245D0712","B17276D0712","B17164D0712","B17167D0712","B17086D0712","B15097D0712","B15212D0712","B17189D0712","B15122D0712","B17185D0712","B17087D0712",

"B17339D0712","B17337D0712","B16054D0712","B16150D0712","B17150D0712","B17331D0712","B16050D0712","B74datD0712","B17201D0712","B17283D0712","B17266D0712","B16076D0712","B16132D0712","B17314D0712","B15107D0712","B17263D0712","B15286D0712","B17212D0712","B17192D0712","B16139D0523","B17285D0523","B17186D0523","B15218D0523","B14147D0523","B17136D0523","B17199D0523","B17081D0523",

"B17254D0523","B17195D0523","B15274D0523","B16111D0523","B17324D0523","B15273D0523","B15191D0523","B17258D0523","B17214D0523","B17233D0523","B17154D0523","B17219D0523","B17184D0523","B17091D0523","B17325D0523","B14124D0523","B15108D0523","B17326D0523","B17001D0523","B17225D0523","B17172D0523","B15245D0523","B17276D0523","B17167D0523","B17086D0523","B15097D0523","B17189D0523",

"B17185D0523","B17087D0523","B17339D0523","B17117D0523","B17337D0523","B16152D0523","B17309D0523","B16054D0523","B16150D0523","B17150D0523","B17187D0523","B17331D0523","B16244D0523","B17201D0523","B17283D0523","B17266D0523","B16076D0523","B16132D0523","B15258D0523","B17314D0523","B15286D0523","B17212D0523","B17362D0523","B17192D0523","B17190D0524","B16050D0524","B16128D0620m",

"B17326D0620m","B17164D0620m","B17164D0620n","B17192D0620m","B17192D0620e","B150195D0725","B150543D0725","B150625D0725","B160085D0725","B160205D0725","B160578D0725","B160847D0725","B170511D0725")]

table2 <- Gene\_Expression\_log[,c("B170631D0725","B170678D0725","B170688D0725","B170821D0725","B180085D0725","B150235D0725","B150377D0725","B150420D0725","B160076D0725","B160254D0725","B160286D0725","B160544D0725","B160580D0725","B160601D0725","B160626D0725","B160720D0725","B160783D0725","B160858D0725","B160895D0725","B170012D0725","B170063D0725","B170506D0725","B170568D0725","B170620D0725","B170628D0725","B170718D0725","B170728D0725","B170738D0725","B170778D0725","B170855D0725","B170896D0725","B170916D0725","B170962D0725","B170985D0725","B180011D0725","B180015D0725","B160677D0725","B160796D0725","B160862D0725","B170519D0725","B170546D0725","B170658D0725","B17117D0620m","B17117D0620e","B17147","B17176","B17268","B17360","B18035","B18091","B18109","B15220D0625","B15274D0625","B17035D0625","B18035D0625","B19013D0625","B15273D1225","B16098D0106","B16139D0106","B16139D1225","B16252D0106","B17114D1225","B17131D1225","B17195D0106","B17354D1225","B18055D1225","B18091D1225","B18096D0106","B18112D0106","B18267D1225","B19101D0106","B19204D1225","B160206D0830","B160846D0830.1","B170033D0830","B170084D0830","B170455D0830","B170881D0830","B180160D0830","B180181D0830","B180229D0830","B180394D0830","B180396D0830","B180414D0830","B180432D0830","B180433D0830","B180466D0830","B160846D0830.2","B16139","B17186","B15218","B17081","B17254","B14156","B17195","B17092","B17205","B16111","B17190","B17258","B17233","B17184","B17172","B17167","B17086","B15097","B17189","B17185","B17337","B16152","B17309","B17187","B16050","B17283","B17266","B16132","B15258","B15286","B17212","B180057D0828","B180061D0828","B180075D0828","B180096D0828","B15322D0530","B16064D0530","B16173D0530","B17119D0530","B17129D0530","B17333D0530","B18015D0530","B16191D0530","B17103D0530","B18028D0530","B16125D0530","B18038D0530","B16020D0530","B16096D0530","B16265D0530","B17231D0530","B15181D0530","B15313D0530","B16101D0530","B16170D0530","B17030D0530","B17035D0530","B17101D0530","B150513D0725","B150524D0725","B160042D0725","B160173D0725","B160586D0725","B170374D0725","B170381D0725","B170683D0725","B170736D0725","B170823D0725","B170895D0725","B170948D0725","B170992D0725","B180008D0725","B180024D0725","B180038D0725","B180061D0725","B180096D0725","B180608D0725","B16257D0625","B17103D0625")]

table\_group <- cbind(table1,table2)

table\_1 <- table\_group[,1:13]

table\_1 <- table\_1[,sample(1:13,13)]

table\_2 <- table\_group[,14:40]

table\_2 <- table\_2[,sample(1:27,27)]

table\_3 <- table\_group[,41:97]

table\_3 <- table\_3[,sample(1:57,57)]

table\_4 <- table\_group[,98:164]

table\_4 <- table\_4[,sample(1:67,67)]

table\_5 <- table\_group[,165:216]

table\_5 <- table\_5[,sample(1:52,52)]

table\_6 <- table\_group[,217:244]

table\_6 <- table\_6[,sample(1:28,28)]

table\_7 <- table\_group[,245:291]

table\_7 <- table\_7[,sample(1:47,47)]

table\_8 <- table\_group[,292:339]

table\_8 <- table\_8[,sample(1:48,48)]

table\_group <- cbind(table\_1,table\_2,table\_3,table\_4,table\_5,table\_6,table\_7,table\_8)

write.table(table\_group,"table\_group.txt",col.names = T,row.names = T,sep="\t",quote=F)

table\_jl <- Gene\_Expression\_log[,c("B17285D0620e","B17199D0620e","B17339D0620e","B17283D0620e","B16132D0620e","B150524D0828E","B160042D0828E","B160132D0828E","B160173D0828E","B170193D0828E","B170477D0828E","B170524D0828E","B170895D0828E","B170948D0828E","B170992D0828E","B180008D0828E","B180024D0828E","B180038D0828E","B180057D0828E","B180061D0828E","B150513D0729E","B150524D0729E","B160042D0729E","B170077D0729E","B170193D0729E","B170992D0729E","B180096D0729E","B180608D0729E","B17285D0620m","B17199D0620m","B17339D0620m","B17283D0620m","B16132D0620m","B150524D0828M","B160042D0828M","B160132D0828M","B160173D0828M","B170193D0828M","B170477D0828M","B170524D0828M","B170895D0828M","B170948D0828M","B170992D0828M","B180008D0828M","B180024D0828M","B180038D0828M","B180057D0828M","B180061D0828M","B150513D0729M","B150524D0729M","B160042D0729M","B170077D0729M","B170193D0729M","B170992D0729M","B180096D0729M","B180608D0729M","B17285D0620n","B17199D0620n","B17339D0620n","B17283D0620n","B16132D0620n","B150524D0828N","B160042D0828N","B160132D0828N","B160173D0828N","B170193D0828N","B170477D0828N","B170524D0828N","B170895D0828N","B170948D0828N","B170992D0828N","B180008D0828N","B180024D0828N","B180038D0828N","B180057D0828N","B180061D0828N","B150513D0729N","B150524D0729N","B160042D0729N","B170077D0729N","B170193D0729N","B170992D0729N","B180096D0729N","B180608D0729N")]

write.table(table\_jl,"table\_jl.txt",col.names = T,row.names = T,sep="\t",quote=F)

###################

table\_group <- read.table("table\_group.txt",header = T)

p1 <- function(x){

p <- wilcox.test(x[1:13],x[14:339])$p.value

return(p)

}

foldchange1 <- function(x){

foldchange <- mean(x[1:13])-mean(x[14:339])

return(foldchange)

}

age\_p <- apply(table\_group,1,p1)

age\_fc <- apply(table\_group,1,foldchange1)

p2 <- function(x){

p <- wilcox.test(x[14:40],c(x[1:13],x[41:339]))$p.value

return(p)

}

foldchange2 <- function(x){

foldchange <- mean(x[14:40])-mean(c(x[1:13],x[41:339]))

return(foldchange)

}

chu\_p <- apply(table\_group,1,p2)

chu\_fc <- apply(table\_group,1,foldchange2)

p3 <- function(x){

p <- wilcox.test(x[41:97],c(x[1:40],x[98:339]))$p.value

return(p)

}

foldchange3 <- function(x){

foldchange <- mean(x[41:97])-mean(c(x[1:40],x[98:339]))

return(foldchange)

}

heat\_p <- apply(table\_group,1,p3)

heat\_fc <- apply(table\_group,1,foldchange3)

p4 <- function(x){

p <- wilcox.test(x[98:164],c(x[1:97],x[165:339]))$p.value

return(p)

}

foldchange4 <- function(x){

foldchange <- mean(x[98:164])-mean(c(x[1:97],x[165:339]))

return(foldchange)

}

late\_p <- apply(table\_group,1,p4)

late\_fc <- apply(table\_group,1,foldchange4)

p5 <- function(x){

p <- wilcox.test(x[165:216],c(x[1:164],x[217:339]))$p.value

return(p)

}

foldchange5 <- function(x){

foldchange <- mean(x[165:216])-mean(c(x[1:164],x[217:339]))

return(foldchange)

}

low\_p <- apply(table\_group,1,p5)

low\_fc <- apply(table\_group,1,foldchange5)

p6 <- function(x){

p <- wilcox.test(x[217:244],c(x[1:216],x[245:339]))$p.value

return(p)

}

foldchange6 <- function(x){

foldchange <- mean(x[217:244])-mean(c(x[1:216],x[245:339]))

return(foldchange)

}

yan\_p <- apply(table\_group,1,p6)

yan\_fc <- apply(table\_group,1,foldchange6)

p7 <- function(x){

p <- wilcox.test(x[245:291],c(x[1:244],x[292:339]))$p.value

return(p)

}

foldchange7 <- function(x){

foldchange <- mean(x[245:291])-mean(c(x[1:244],x[292:339]))

return(foldchange)

}

zao\_p <- apply(table\_group,1,p7)

zao\_fc <- apply(table\_group,1,foldchange7)

p8 <- function(x){

p <- wilcox.test(x[292:339],c(x[1:291]))$p.value

return(p)

}

foldchange8 <- function(x){

foldchange <- mean(x[292:339])-mean(c(x[1:291]))

return(foldchange)

}

zhong\_p <- apply(table\_group,1,p8)

zhong\_fc <- apply(table\_group,1,foldchange8)

table\_group\_p\_fc <- data.frame(age\_p,age\_fc,chu\_p,chu\_fc,heat\_p,heat\_fc,late\_p,late\_fc,low\_p,low\_fc,yan\_p,yan\_fc,zao\_p,zao\_fc,zhong\_p,zhong\_fc)

gene\_age <- table\_group\_p\_fc[table\_group\_p\_fc$age\_p<0.01 & table\_group\_p\_fc$age\_fc>1,]

gene\_chu <- table\_group\_p\_fc[table\_group\_p\_fc$chu\_p<0.01 & table\_group\_p\_fc$chu\_fc>1,]

gene\_heat <- table\_group\_p\_fc[table\_group\_p\_fc$heat\_p<0.01 & table\_group\_p\_fc$heat\_fc>1,]

gene\_late <- table\_group\_p\_fc[table\_group\_p\_fc$late\_p<0.01 & table\_group\_p\_fc$late\_fc>1,]

gene\_low <- table\_group\_p\_fc[table\_group\_p\_fc$low\_p<0.01 & table\_group\_p\_fc$low\_fc>1,]

gene\_yan <- table\_group\_p\_fc[table\_group\_p\_fc$yan\_p<0.01 & table\_group\_p\_fc$yan\_fc>1,]

gene\_zao <- table\_group\_p\_fc[table\_group\_p\_fc$zao\_p<0.01 & table\_group\_p\_fc$zao\_fc>1,]

gene\_zhong <- table\_group\_p\_fc[table\_group\_p\_fc$zhong\_p<0.01 & table\_group\_p\_fc$zhong\_fc>1,]

gene\_age <- gene\_age[order(gene\_age$age\_fc,decreasing = T),]

gene\_chu <- gene\_chu[order(gene\_chu$chu\_fc,decreasing = T),]

gene\_heat <- gene\_heat[order(gene\_heat$heat\_fc,decreasing = T),]

gene\_late <- gene\_late[order(gene\_late$late\_fc,decreasing = T),]

gene\_low <- gene\_low[order(gene\_low$low\_fc,decreasing = T),]

gene\_yan <- gene\_yan[order(gene\_yan$yan\_fc,decreasing = T),]

gene\_zao <- gene\_zao[order(gene\_zao$zao\_fc,decreasing = T),]

gene\_zhong <- gene\_zhong[order(gene\_zhong$zhong\_fc,decreasing = T),]

gene\_age\_specific <- setdiff(rownames(gene\_age),rownames(gene\_chu))

gene\_age\_specific <- setdiff(gene\_age\_specific,rownames(gene\_heat))

gene\_age\_specific <- setdiff(gene\_age\_specific,rownames(gene\_late))

gene\_age\_specific <- setdiff(gene\_age\_specific,rownames(gene\_low))

gene\_age\_specific <- setdiff(gene\_age\_specific,rownames(gene\_yan))

gene\_age\_specific <- setdiff(gene\_age\_specific,rownames(gene\_zao))

gene\_age\_specific <- setdiff(gene\_age\_specific,rownames(gene\_zhong))

gene\_chu\_specific <- setdiff(rownames(gene\_chu),rownames(gene\_age))

gene\_chu\_specific <- setdiff(gene\_chu\_specific,rownames(gene\_heat))

gene\_chu\_specific <- setdiff(gene\_chu\_specific,rownames(gene\_late))

gene\_chu\_specific <- setdiff(gene\_chu\_specific,rownames(gene\_low))

gene\_chu\_specific <- setdiff(gene\_chu\_specific,rownames(gene\_yan))

gene\_chu\_specific <- setdiff(gene\_chu\_specific,rownames(gene\_zao))

gene\_chu\_specific <- setdiff(gene\_chu\_specific,rownames(gene\_zhong))

gene\_heat\_specific <- setdiff(rownames(gene\_heat),rownames(gene\_chu))

gene\_heat\_specific <- setdiff(gene\_heat\_specific,rownames(gene\_age))

gene\_heat\_specific <- setdiff(gene\_heat\_specific,rownames(gene\_late))

gene\_heat\_specific <- setdiff(gene\_heat\_specific,rownames(gene\_low))

gene\_heat\_specific <- setdiff(gene\_heat\_specific,rownames(gene\_yan))

gene\_heat\_specific <- setdiff(gene\_heat\_specific,rownames(gene\_zao))

gene\_heat\_specific <- setdiff(gene\_heat\_specific,rownames(gene\_zhong))

gene\_late\_specific <- setdiff(rownames(gene\_late),rownames(gene\_chu))

gene\_late\_specific <- setdiff(gene\_late\_specific,rownames(gene\_heat))

gene\_late\_specific <- setdiff(gene\_late\_specific,rownames(gene\_age))

gene\_late\_specific <- setdiff(gene\_late\_specific,rownames(gene\_low))

gene\_late\_specific <- setdiff(gene\_late\_specific,rownames(gene\_yan))

gene\_late\_specific <- setdiff(gene\_late\_specific,rownames(gene\_zao))

gene\_late\_specific <- setdiff(gene\_late\_specific,rownames(gene\_zhong))

gene\_low\_specific <- setdiff(rownames(gene\_low),rownames(gene\_chu))

gene\_low\_specific <- setdiff(gene\_low\_specific,rownames(gene\_heat))

gene\_low\_specific <- setdiff(gene\_low\_specific,rownames(gene\_late))

gene\_low\_specific <- setdiff(gene\_low\_specific,rownames(gene\_age))

gene\_low\_specific <- setdiff(gene\_low\_specific,rownames(gene\_yan))

gene\_low\_specific <- setdiff(gene\_low\_specific,rownames(gene\_zao))

gene\_low\_specific <- setdiff(gene\_low\_specific,rownames(gene\_zhong))

gene\_yan\_specific <- setdiff(rownames(gene\_yan),rownames(gene\_chu))

gene\_yan\_specific <- setdiff(gene\_yan\_specific,rownames(gene\_heat))

gene\_yan\_specific <- setdiff(gene\_yan\_specific,rownames(gene\_late))

gene\_yan\_specific <- setdiff(gene\_yan\_specific,rownames(gene\_low))

gene\_yan\_specific <- setdiff(gene\_yan\_specific,rownames(gene\_age))

gene\_yan\_specific <- setdiff(gene\_yan\_specific,rownames(gene\_zao))

gene\_yan\_specific <- setdiff(gene\_yan\_specific,rownames(gene\_zhong))

gene\_zao\_specific <- setdiff(rownames(gene\_zao),rownames(gene\_chu))

gene\_zao\_specific <- setdiff(gene\_zao\_specific,rownames(gene\_heat))

gene\_zao\_specific <- setdiff(gene\_zao\_specific,rownames(gene\_late))

gene\_zao\_specific <- setdiff(gene\_zao\_specific,rownames(gene\_low))

gene\_zao\_specific <- setdiff(gene\_zao\_specific,rownames(gene\_yan))

gene\_zao\_specific <- setdiff(gene\_zao\_specific,rownames(gene\_age))

gene\_zao\_specific <- setdiff(gene\_zao\_specific,rownames(gene\_zhong))

gene\_zhong\_specific <- setdiff(rownames(gene\_zhong),rownames(gene\_chu))

gene\_zhong\_specific <- setdiff(gene\_zhong\_specific,rownames(gene\_heat))

gene\_zhong\_specific <- setdiff(gene\_zhong\_specific,rownames(gene\_late))

gene\_zhong\_specific <- setdiff(gene\_zhong\_specific,rownames(gene\_low))

gene\_zhong\_specific <- setdiff(gene\_zhong\_specific,rownames(gene\_yan))

gene\_zhong\_specific <- setdiff(gene\_zhong\_specific,rownames(gene\_age))

gene\_zhong\_specific <- setdiff(gene\_zhong\_specific,rownames(gene\_zao))

gene\_all <- c(gene\_age\_specific[1:3],gene\_chu\_specific[1:3],gene\_heat\_specific[1:3],

gene\_late\_specific[1:3],gene\_low\_specific[1:3],gene\_yan\_specific[1:3],

gene\_zao\_specific[1:3],gene\_zhong\_specific[1:3]) #c(rownames(gene\_age\_specific))#,rownames(gene\_chu),rownames(gene\_heat),rownames(gene\_low),rownames(gene\_yan),rownames(gene\_zao),rownames(gene\_zhong))

gene\_all2 <- data.frame(Gene=gene\_all)

table\_group$Gene <- rownames(table\_group)

library(dplyr)

table\_show <- gene\_all2 %>% left\_join(table\_group,by="Gene")

table\_data <- table\_show[,2:340]

table\_data <- t(scale(t(table\_data)))

table\_data <- data.frame(table\_data)

table\_data2 <- cbind(table\_show[,c(1,1)],table\_data)

write.table(table\_data2,"table\_show.txt",col.names = T,row.names = T,sep="\t",quote=F)

library(pheatmap)

table\_show2 <- table\_show[,2:340]

rownames(table\_show2) <- table\_show$Gene

cat\_df <- data.frame("category" = c(rep("Senescence",13),rep("Colostrum",27),rep("Heat",57),rep("Late",67),rep("Low",52),rep("Mastitis",28),rep("Early",47),rep("Middle",48)))

rownames(cat\_df) <- colnames(table\_show2)

pheatmap(table\_show2,scale="row",cluster\_rows = F,border\_color = NA,cluster\_cols = F,annotation\_col = cat\_df,color = colorRampPalette(c("purple","black","yellow"))(101))

out\_specific\_genes <- data.frame(geneid=c(gene\_age\_specific,gene\_chu\_specific,gene\_heat\_specific,gene\_late\_specific,gene\_low\_specific,gene\_yan\_specific,gene\_zao\_specific,gene\_zhong\_specific),

category = c(rep("age",23),rep("chu",52),rep("heat",579),rep("late",102),rep("low",245),rep("yan",861),rep("zao",1613),rep("zhong",3)))

write.table(out\_specific\_genes,"group\_specific\_genes.txt",col.names = T,row.names = F,sep="\t",quote=F)

##############

table\_jl\_2 <- table\_jl[,c("B17285D0620m","B17285D0620n","B17285D0620e","B17199D0620m","B17199D0620n","B17199D0620e","B17339D0620m","B17339D0620n","B17339D0620e","B17283D0620m","B17283D0620n","B17283D0620e","B16132D0620m","B16132D0620n","B16132D0620e","B150524D0828M","B150524D0828N","B150524D0828E","B160042D0828M","B160042D0828N","B160042D0828E","B160132D0828M","B160132D0828N","B160132D0828E","B160173D0828M","B160173D0828N","B160173D0828E","B170193D0828M","B170193D0828N","B170193D0828E","B170477D0828M","B170477D0828N","B170477D0828E","B170524D0828M","B170524D0828N","B170524D0828E","B170895D0828M","B170895D0828N","B170895D0828E","B170948D0828M","B170948D0828N","B170948D0828E","B170992D0828M","B170992D0828N","B170992D0828E","B180008D0828M","B180008D0828N","B180008D0828E","B180024D0828M","B180024D0828N","B180024D0828E","B180038D0828M","B180038D0828N","B180038D0828E","B180057D0828M","B180057D0828N","B180057D0828E","B180061D0828M","B180061D0828N","B180061D0828E","B150513D0729M","B150513D0729N","B150513D0729E","B150524D0729M","B150524D0729N","B150524D0729E","B160042D0729M","B160042D0729N","B160042D0729E","B170077D0729M","B170077D0729N","B170077D0729E","B170193D0729M","B170193D0729N","B170193D0729E","B170992D0729M","B170992D0729N","B170992D0729E","B180096D0729M","B180096D0729N","B180096D0729E","B180608D0729M","B180608D0729N","B180608D0729E")]

chaN <- c()

for (i in 0:27) {

chaN <- cbind(chaN,table\_jl\_2[,(i\*3+2)]-table\_jl\_2[,(i\*3+1)])

}

chaN <- data.frame(geneid=rownames(table\_jl\_2),chaN)

chaE <- c()

for (i in 0:27) {

chaE <- cbind(chaE,table\_jl\_2[,(i\*3+3)]-table\_jl\_2[,(i\*3+1)])

}

chaE <- data.frame(geneid=rownames(table\_jl\_2),chaE)

colnames(chaN) <- c("GeneId","A.B17285D0620","A.B17199D0620","A.B17339D0620","A.B17283D0620","A.B16132D0620","A.B150524D0828","A.B160042D0828","A.B160132D0828","A.B160173D0828","A.B170193D0828","A.B170477D0828","A.B170524D0828","A.B170895D0828","A.B170948D0828","A.B170992D0828","A.B180008D0828","A.B180024D0828","A.B180038D0828","A.B180057D0828","A.B180061D0828","A.B150513D0729","A.B150524D0729","A.B160042D0729","A.B170077D0729","A.B170193D0729","A.B170992D0729","A.B180096D0729","A.B180608D0729")

colnames(chaE) <- c("GeneId","B.B17285D0620","B.B17199D0620","B.B17339D0620","B.B17283D0620","B.B16132D0620","B.B150524D0828","B.B160042D0828","B.B160132D0828","B.B160173D0828","B.B170193D0828","B.B170477D0828","B.B170524D0828","B.B170895D0828","B.B170948D0828","B.B170992D0828","B.B180008D0828","B.B180024D0828","B.B180038D0828","B.B180057D0828","B.B180061D0828","B.B150513D0729","B.B150524D0729","B.B160042D0729","B.B170077D0729","B.B170193D0729","B.B170992D0729","B.B180096D0729","B.B180608D0729")

table\_jl\_3 <- merge(chaN,chaE,by="GeneId")

chaNmean <- apply(chaN[,2:29], 1, mean)

chaEmean <- apply(chaE[,2:29], 1, mean)

data2 <- data.frame(geneid=(table\_jl\_3$GeneId),chaNmean,chaEmean)

write.table(data2,"Group\_diff\_jl.txt",col.names = T,row.names = F,sep="\t",quote = F)

#set.seed(10000)

#cls <- kmeans(data2[,2:3],centers = 20)

group\_out <- read.table("Group\_diff\_jl.out.txt",sep="\t",header = T,na.strings = "NA")

group\_out$type <- "1"

group\_out$type[group\_out$Profile %in% c(1,5,6)] <- "G1"

group\_out$type[group\_out$Profile %in% c(4)] <- "G2"

group\_out$type[group\_out$Profile %in% c(9,10,14)] <- "G3"

group\_out$type[group\_out$Profile %in% c(7)] <- "G4"

group\_out$type[group\_out$Profile %in% c(0,2,3)] <- "G5"

group\_out$type[group\_out$Profile %in% c(8)] <- "G6"

group\_out$type[group\_out$Profile %in% c(11)] <- "G7"

group\_out$type[group\_out$Profile %in% c(12,13,15)] <- "G8"

group\_out <- group\_out[,c(2,8)]

gene\_out1 <- table\_jl\_2[rownames(table\_jl\_2) %in% c("NBEAL1"),]

gene\_out1 <- data.frame(gene="NBEAL1",M=as.vector(t(gene\_out1[1,seq(1,84,by = 3)])),N=as.vector(t(gene\_out1[1,seq(2,84,by = 3)])),E=as.vector(t(gene\_out1[1,seq(3,84,by = 3)])))

boxplot(gene\_out1[2:4],outline = F)

s <- apply(gene\_out1[2:4], 2, mean)

##################

gene\_out1 <- merge(data2,group\_out,by="geneid")

gene\_out1$start <- 0

library(reshape2)

gene\_out1\_2 <- melt(gene\_out1)

library(ggplot2)

gene\_out1\_2$variable2 <- "Moring"

gene\_out1\_2$variable2[gene\_out1\_2$variable=="chaNmean"] <- "Noon"

gene\_out1\_2$variable2[gene\_out1\_2$variable=="chaEmean"] <- "Evening"

gene\_out1\_2$variable2 <- factor(gene\_out1\_2$variable2,levels = c("Moring","Noon","Evening"))

gene\_out1\_3 <- gene\_out1\_2 %>% group\_by(type,variable2) %>% summarise(value=mean(value))

gene\_out1\_2 <- gene\_out1\_2[,c("type","variable2","value","geneid")]

gene\_out1\_3$colr <- "red"

gene\_out1\_3$geneid <- "mean"

gene\_out1\_2$colr <- "grey"

gene\_out1\_4 <- rbind(gene\_out1\_2) #gene\_out1\_3

ggplot(gene\_out1\_4,aes(x=variable2,y=value,group=geneid)) + geom\_line(col="grey") + facet\_wrap(~type,nrow=2) + theme\_classic() + ylab("Log2 Foldchange to moring") + xlab("") + coord\_cartesian(ylim = c(-1,0.75))

ggplot(gene\_out1\_3,aes(x=variable2,y=value,group=geneid)) + geom\_line(col="red") + facet\_wrap(~type,nrow=2) + theme\_classic() + ylab("Log2 Foldchange to moring") + xlab("") + coord\_cartesian(ylim = c(-1,0.75))

Bulk-milk tsne

######################################################

##########using t-SNE (non-linear) instead PCA(linear)

x <- read.table("Count.txt",header = T)

rownames(x) <- x$geneid

#x <- x[,-1]

library(dplyr)

library(reshape2)

library(ggpubr)

library(extrafont)

x3 <- melt(x)

x4 <- x3 %>% group\_by(variable) %>% summarise(value2=sum(value))

x5 <- merge(x3,x4,by="variable")

x5$RPM <- x5$value \* 1000000 / x5$value2

x5 <- x5[,c(2,1,5)]

x5 <- dcast(x5, geneid~variable)

rownames(x5) <- x5$geneid

Gene\_Expression <- x5[,2:424]

Gene\_Expression\_log<-log2(Gene\_Expression+0.25)

Gene\_Expression\_log <- t(Gene\_Expression\_log)

library(Rtsne)

Train <- Gene\_Expression\_log

Texts <- rownames(Gene\_Expression\_log)

#new\_lables <- Smaple\_Summary$Tissue\_visa

#colors = rainbow(length(unique(new\_lables)))

#colors[4] <- "black"

#names(colors) = unique(new\_lables)

###clustering

set.seed(20002)

tsne <- Rtsne(Train, dims = 2, perplexity=30, verbose=TRUE, max\_iter = 5000,theta = 0.0)

set.seed(10000)

cl\_i <- kmeans(tsne$Y, centers = 10, nstart = 3,iter.max=1000) # k = 10

trainout <- cbind(rownames(Train),tsne$Y,cl\_i$cluster)

genotype <- read.table("genotype.txt")

trainout2 <- merge(trainout,genotype,by.x="V1",by.y="V1")

trainout2$V2.x <- as.numeric(trainout2$V2.x)

trainout2$V3 <- as.numeric(trainout2$V3)

library(ggplot2)

g1 <- ggplot(trainout2, aes(x=V2.x, y=V3,col=V4)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

#g <- g + geom\_text(aes(label=name), size=4, vjust = 0) #,nudge\_x = c(0,1,0,0,0,0,0,-1.5,1,0,1.5),nudge\_y = c(-0.5,0,0.5,1,0.5,-0.5,0,0,0,0.5,0))

g1 <- g1 + xlab("TSNE-1") + ylab("TSNE-2") + theme\_classic() + theme(legend.position = "none") + ggtitle("TSNE")

g1

#library(ggplot2)

#g <- ggplot(trainout2, aes(x=V2.x, y=V3,col=V2.y)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

#g <- g + geom\_text(aes(label=name), size=4, vjust = 0) #,nudge\_x = c(0,1,0,0,0,0,0,-1.5,1,0,1.5),nudge\_y = c(-0.5,0,0.5,1,0.5,-0.5,0,0,0,0.5,0))

#g <- g + xlab("PC1(16.2%)") + ylab("PC2(10.3%)")

#g

trainout2$label <- "other"

trainout2$label[trainout2$V2.y=="chu"] <- "chu"

table2\_1 <- trainout2[trainout2$label=="other",]

table2\_2 <- trainout2[trainout2$label=="chu",]

g2 <- ggplot(table2\_1, aes(x=V2.x, y=V3,col=label)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

g2 <- g2 + geom\_point(data=table2\_2,mapping = aes(x=V2.x, y=V3,col=label)) + scale\_color\_manual(values = c("red","grey")) + xlab("TSNE-1") + ylab("TSNE-2")

g2 <- g2 + theme\_classic() + theme(legend.position = "none") + ggtitle("colostrum")

trainout2$label <- "other"

trainout2$label[trainout2$V2.y=="low"] <- "low"

table2\_1 <- trainout2[trainout2$label=="other",]

table2\_2 <- trainout2[trainout2$label=="low",]

g3 <- ggplot(table2\_1, aes(x=V2.x, y=V3,col=label)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

g3 <- g3 + geom\_point(data=table2\_2,mapping = aes(x=V2.x, y=V3,col=label)) + scale\_color\_manual(values = c("red","grey")) + xlab("TSNE-1") + ylab("TSNE-2")

g3 <- g3 + theme\_classic() + theme(legend.position = "none") + ggtitle("low yield")

trainout2$label <- "other"

trainout2$label[trainout2$V2.y=="zao"] <- "zao"

table2\_1 <- trainout2[trainout2$label=="other",]

table2\_2 <- trainout2[trainout2$label=="zao",]

g4 <- ggplot(table2\_1, aes(x=V2.x, y=V3,col=label)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

g4 <- g4 + geom\_point(data=table2\_2,mapping = aes(x=V2.x, y=V3,col=label)) + scale\_color\_manual(values = c("grey","red")) + xlab("TSNE-1") + ylab("TSNE-2")

g4 <- g4 + theme\_classic() + theme(legend.position = "none") + ggtitle("early stage")

trainout2$label <- "other"

trainout2$label[trainout2$V2.y=="age"] <- "age"

table2\_1 <- trainout2[trainout2$label=="other",]

table2\_2 <- trainout2[trainout2$label=="age",]

g5 <- ggplot(table2\_1, aes(x=V2.x, y=V3,col=label)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

g5 <- g5 + geom\_point(data=table2\_2,mapping = aes(x=V2.x, y=V3,col=label)) + scale\_color\_manual(values = c("red","grey")) + xlab("TSNE-1") + ylab("TSNE-2")

g5 <- g5 + theme\_classic() + theme(legend.position = "none") + ggtitle("senescence")

trainout2$label <- "other"

trainout2$label[trainout2$V2.y=="zhong"] <- "zhong"

table2\_1 <- trainout2[trainout2$label=="other",]

table2\_2 <- trainout2[trainout2$label=="zhong",]

g6 <- ggplot(table2\_1, aes(x=V2.x, y=V3,col=label)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

g6 <- g6 + geom\_point(data=table2\_2,mapping = aes(x=V2.x, y=V3,col=label)) + scale\_color\_manual(values = c("grey","red")) + xlab("TSNE-1") + ylab("TSNE-2")

g6 <- g6 + theme\_classic() + theme(legend.position = "none") + ggtitle("medium stage")

trainout2$label <- "other"

trainout2$label[trainout2$V2.y=="late"] <- "late"

table2\_1 <- trainout2[trainout2$label=="other",]

table2\_2 <- trainout2[trainout2$label=="late",]

g7 <- ggplot(table2\_1, aes(x=V2.x, y=V3,col=label)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

g7 <- g7 + geom\_point(data=table2\_2,mapping = aes(x=V2.x, y=V3,col=label)) + scale\_color\_manual(values = c("red","grey")) + xlab("TSNE-1") + ylab("TSNE-2")

g7 <- g7 + theme\_classic() + theme(legend.position = "none") + ggtitle("late stage")

trainout2$label <- "other"

trainout2$label[trainout2$V2.y=="yan"] <- "yan"

table2\_1 <- trainout2[trainout2$label=="other",]

table2\_2 <- trainout2[trainout2$label=="yan",]

g8 <- ggplot(table2\_1, aes(x=V2.x, y=V3,col=label)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

g8 <- g8 + geom\_point(data=table2\_2,mapping = aes(x=V2.x, y=V3,col=label)) + scale\_color\_manual(values = c("grey","red")) + xlab("TSNE-1") + ylab("TSNE-2")

g8 <- g8 + theme\_classic() + theme(legend.position = "none") + ggtitle("mastitis")

trainout2$label <- "other"

trainout2$label[trainout2$V2.y=="m"] <- "m"

table2\_1 <- trainout2[trainout2$label=="other",]

table2\_2 <- trainout2[trainout2$label=="m",]

g9 <- ggplot(table2\_1, aes(x=V2.x, y=V3,col=label)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

g9 <- g9 + geom\_point(data=table2\_2,mapping = aes(x=V2.x, y=V3,col=label)) + scale\_color\_manual(values = c("red","grey")) + xlab("TSNE-1") + ylab("TSNE-2")

g9 <- g9 + theme\_classic() + theme(legend.position = "none") + ggtitle("Morning")

trainout2$label <- "other"

trainout2$label[trainout2$V2.y=="n"] <- "n"

table2\_1 <- trainout2[trainout2$label=="other",]

table2\_2 <- trainout2[trainout2$label=="n",]

g10 <- ggplot(table2\_1, aes(x=V2.x, y=V3,col=label)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

g10 <- g10 + geom\_point(data=table2\_2,mapping = aes(x=V2.x, y=V3,col=label)) + scale\_color\_manual(values = c("red","grey")) + xlab("TSNE-1") + ylab("TSNE-2")

g10 <- g10 + theme\_classic() + theme(legend.position = "none") + ggtitle("Noon")

trainout2$label <- "other"

trainout2$label[trainout2$V2.y=="e"] <- "e"

table2\_1 <- trainout2[trainout2$label=="other",]

table2\_2 <- trainout2[trainout2$label=="e",]

g11 <- ggplot(table2\_1, aes(x=V2.x, y=V3,col=label)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

g11 <- g11 + geom\_point(data=table2\_2,mapping = aes(x=V2.x, y=V3,col=label)) + scale\_color\_manual(values = c("red","grey")) + xlab("TSNE-1") + ylab("TSNE-2")

g11 <- g11 + theme\_classic() + theme(legend.position = "none") + ggtitle("Evening")

trainout2$label <- "other"

trainout2$label[trainout2$V2.y=="heat"] <- "heat"

table2\_1 <- trainout2[trainout2$label=="other",]

table2\_2 <- trainout2[trainout2$label=="heat",]

g13 <- ggplot(table2\_1, aes(x=V2.x, y=V3,col=label)) + geom\_point() #+ theme(legend.position = "none") #+ scale\_y\_continuous(limits = c(0,15)) + scale\_x\_continuous(limits = c(0,15))

g13 <- g13 + geom\_point(data=table2\_2,mapping = aes(x=V2.x, y=V3,col=label)) + scale\_color\_manual(values = c("red","grey")) + xlab("TSNE-1") + ylab("TSNE-2")

g13 <- g13 + theme\_classic() + theme(legend.position = "none") + ggtitle("heat stress")

ggarrange(g1,g2,g3,g4,g5,g6,g7,g8,g9,g10,g11,g13,

ncol = 4, nrow = 3)

trainoutput <- trainout2[,c(1,2,3)]

colnames(trainoutput) <- c("Sample","X","Y")

write.table(trainoutput,"Milk\_bulk\_TSNE\_cluster.txt",col.names = T,row.names = F,sep = "\t",quote=F)

##########################

table3 <- trainout2 %>% group\_by(`V4`,V2.y) %>% summarise(num=n())

table4 <- trainout2 %>% group\_by(`V4`) %>% summarise(num2=n())

table5 <- merge(table3,table4,by="V4")

table5$per <- table5$num / table5$num2 \* 100

write.table(table5,"TSNE-cluster.txt",col.names = T,row.names = F,sep="\t",quote=F)

allgenotype <- trainout2 %>% group\_by(V2.y) %>% summarise(num3=n())

allgenotype$per <- allgenotype$num3/423 \* 100

table5 <- table5[,c(1,2,5)]

colnames(table5) <- c("V4","V2","per")

allgenotype <- data.frame(`V4`="all",V2=allgenotype$V2.y,per=allgenotype$per)

table5 <- rbind(table5,allgenotype)

ggplot(table5,aes(x=`V4`,y=per,fill=V2)) + geom\_bar(stat="identity")

x2 <- dcast(table5,`V4`~V2)

write.table(x2,"TSNE-cluster.2.txt",col.names = T,row.names = F,sep="\t",quote = F)

Cow GO-analysis

---

title: "GO\_and\_KEGG\_enrichment\_analysis\_for\_DEGs\_after\_Nat10\_KO"

author: "HuangFei"

date: "2021/4/20"

output: html\_document

---

## Load dependence

```{r,warning=FALSE,results='hide'}

options(stringsAsFactors = F)

library(ggplot2)

library(ggpubr)

library(tibble)

library(data.table)

library(DT)

library(org.Bt.eg.db)

library(clusterProfiler, quietly = T)

library(DOSE, quietly = T)

```

### define functions to run GO and KEGG respectively

```{r}

set.seed(1000)

run\_KEGG\_mice = function(candidate\_gene, background\_gene = NULL, gene\_format = 'SYMBOL', cutoff=0.05,

showCategory = 20, font.size = 10,title = 'KEGG enrichment'){

diff\_gene\_ID<-clusterProfiler::bitr(candidate\_gene, fromType = gene\_format, toType ="ENTREZID", OrgDb="org.Bt.eg.db")

if(is.null(background\_gene)){

ekegg<-enrichKEGG(gene=diff\_gene\_ID$ENTREZID, organism = 'mmu',

pAdjustMethod = "BH",

pvalueCutoff =cutoff,

qvalueCutoff = cutoff)

}else{

background\_gene = clusterProfiler::bitr(background\_gene, fromType = gene\_format, toType ="ENTREZID", OrgDb="org.Bt.eg.db")

ekegg<-enrichKEGG(gene=diff\_gene\_ID$ENTREZID, organism = 'mmu',

universe = background\_gene$ENTREZID,

pAdjustMethod = "BH",

pvalueCutoff =cutoff,

qvalueCutoff = cutoff)

}

#ekegg <- setReadable(ekegg, OrgDb = org.Bt.eg.db, keytype = "ENTREZID")

ekegg.table = as.data.frame(ekegg)

if(nrow(ekegg.table)>0){

ekegg.table = ekegg.table[order(ekegg.table$p.adjust),]

print(ekegg.table$Description)

print(dotplot(ekegg, showCategory = showCategory, font.size = font.size, x='Count', title= title))

}

return(ekegg)

}

run\_GO\_mice = function(candidate\_gene, background\_gene=NULL, gene\_format = 'SYMBOL', ontology = 'BP', cutoff=0.05,

showCategory=10,font.size=10,title = 'GO enrichment'){

diff\_gene\_ID<-clusterProfiler::bitr(candidate\_gene, fromType = gene\_format, toType ="ENTREZID", OrgDb="org.Bt.eg.db")

if(is.null(background\_gene)){

ego <- simplify(enrichGO(gene = diff\_gene\_ID$ENTREZID, OrgDb = org.Bt.eg.db,

keyType = 'ENTREZID', ont = ontology, readable = T,

pAdjustMethod = "BH", qvalueCutoff = cutoff, pvalueCutoff = cutoff))

} else{

background\_gene = clusterProfiler::bitr(background\_gene, fromType = gene\_format, toType ="ENTREZID", OrgDb="org.Bt.eg.db")

ego <- simplify(enrichGO(gene = diff\_gene\_ID$ENTREZID, OrgDb = org.Bt.eg.db,

universe = background\_gene$ENTREZID,

keyType = 'ENTREZID', ont = ontology, readable = T,

pAdjustMethod = "BH", qvalueCutoff = cutoff, pvalueCutoff = cutoff))

}

if(nrow(ego@result)>0){

print(dotplot(ego, showCategory = showCategory, font.size = font.size, x='Count',title=title))

}

return(ego)

}

```

```{r}

ego = run\_GO\_mice(candidate\_gene = rownames(healthyhigh), background\_gene = NULL)

ggsave("healthyhigh.GO.pdf")

#ekegg = run\_KEGG\_mice(candidate\_gene = DEGs, background\_gene = NULL)

```

```{r}

ego = run\_GO\_mice(candidate\_gene = rownames(infecthigh), background\_gene = NULL)

ggsave("infecthigh.GO.pdf")

#ekegg = run\_KEGG\_mice(candidate\_gene = DEGs, background\_gene = NULL)

```

```{r}

ego = run\_GO\_mice(candidate\_gene = rownames(gene\_cool), background\_gene = NULL)

ggsave("Cool.GO.pdf")

#ekegg = run\_KEGG\_mice(candidate\_gene = DEGs, background\_gene = NULL)

```

```{r}

ego = run\_GO\_mice(candidate\_gene = rownames(gene\_heat), background\_gene = NULL)

ggsave("Heat.GO.pdf")

#ekegg = run\_KEGG\_mice(candidate\_gene = DEGs, background\_gene = NULL)

```

```{r}

table2 <- ego@result

table2 <- table2[1:10,]

table2 <- table2[order(table2$qvalue,decreasing = T),]

table2$Description <- factor(table2$Description,levels=table2$Description)

ggplot(table2,aes(x=Description,y=-log10(qvalue))) + geom\_bar(position = position\_dodge(),stat = "identity") +

theme\_classic() + coord\_flip() + xlab("")

#ggsave("GO-cluster3.pdf",width = 7,height=4)

```

```{r}

table2 <- ekegg@result

table2 <- table2[1:10,]

table2 <- table2[order(table2$qvalue,decreasing = T),]

table2$Description <- factor(table2$Description,levels=table2$Description)

ggplot(table2,aes(x=Description,y=-log10(qvalue))) + geom\_bar(position = position\_dodge(),stat = "identity") +

theme\_classic() + coord\_flip() + xlab("")

ggsave("KEGG-cluster2.pdf",width = 7,height=4)

```

getwd()

write.table(table2,"Nascent-ZGA-IAA.GO.txt",col.names = T,row.names = F,sep = "\t",quote = F)

**Deconvolution**

x <- read.csv("CIBERSORTx\_Job7\_Results.csv")

x1 <- x[,2:20]

x2 <- x1 / x$Absolute.score..sig.score.

xout <- cbind(x$Mixture,x2,x[,c(21:24)])

x3 <- cbind(x$Mixture,x2)

x3$`CD8+Tcell` <- x3$ClusterMilk0 + x3$ClusterMilk14 + x3$ClusterMilk15 + x3$ClusterMilk17

x3$`CD4+Tcell` <- x3$ClusterMilk1

x3$`Monocyte` <- x3$ClusterMilk2 + x3$ClusterMilk7

x3$`Macrophage` <- x3$ClusterMilk3 + x3$ClusterMilk9 + x3$ClusterMilk5

x3$`Bcell` <- x3$ClusterMilk4

x3$`Endothelialcell` <- x3$ClusterMilk6

x3$`Mammaryepithelialcellluminal` <- x3$ClusterMilk8

x3$`NKTcell` <- x3$ClusterMilk10

x3$`Dendriticcell` <- x3$ClusterMilk11 + x3$ClusterMilk12

x3$`MammaryepithelialcellAlveolar` <- x3$ClusterMilk13

x3$`GammaDeltaTcell` <- x3$ClusterMilk18 + x3$ClusterMilk16

x4 <- x3[,c(1,21,22,23,24,25,26,27,28,29,30,31)]

rownames(x4) <- x4$`x$Mixture`

x4 <- x4[,-1]

write.table(x4,"Milk-423-deconvolution.txt",col.names = T,row.names = T,sep="\t",quote = F)

x5 <- t(x4)

x5 <- data.frame(x5)

x5$celltype <- rownames(x5)

library(reshape2)

x6 <- melt(x5)

phe <- read.table("../../phenotype2.txt",sep="\t")

x7 <- merge(x6,phe,by.x="variable",by.y="V1")

library(ggplot2)

ggplot(x7,aes(x=variable,y=value,fill=celltype)) + geom\_bar(stat = "identity") + facet\_wrap(~V2,scales = "free") + xlab("") + ylab("Percentage") + theme\_classic() + scale\_fill\_brewer(palette="Paired")

x7$V2 <- factor(x7$V2,levels=c("Colostrum","Early","Middle","Late","Low","Senescence","Heat","Mastitis","Morning","Noon","Evening"))

ggplot(x7,aes(x=V2,y=value)) + geom\_boxplot(outlier.shape = NA) + geom\_point() + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("Percentage") + theme\_classic() + theme(axis.text.x = element\_text(angle = 45,hjust = 1))

library(ggpubr)

pheno <- read.table("../../pheno.txt",sep="\t",header = T)

x8 <- merge(x7,pheno,by.x="variable",by.y="ID")

x8 <- x8[!x8$group2 %in% c("m","n","e"),]

ggplot(x8,aes(x=value,y=Parity)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(celltype~V2,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.Parity.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=DIM)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(celltype~V2,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.DIM.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=Milk.yield)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(celltype~V2,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.Milk.yield.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=ruzhilv)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(celltype~V2,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.ruzhilv.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=danbailv)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(celltype~V2,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.danbailv.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=rutang)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(celltype~V2,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.rutang.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=zongguti)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(celltype~V2,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.zongguti.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=tixibao)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(celltype~V2,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.tixibao.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=niaosudan)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(celltype~V2,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.niaosudan.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=log10(SSC.C))) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(celltype~V2,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.SSC.C.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=DSCC)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(celltype~V2,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.DSCC.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=HIF)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(celltype~V2,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.HIF.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=Parity)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.Parity2.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=DIM)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.DIM2.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=Milk.yield)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.Milk.yield2.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=ruzhilv)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.ruzhilv2.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=danbailv)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.danbailv2.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=rutang)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.rutang2.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=zongguti)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.zongguti2.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=tixibao)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.tixibao2.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=niaosudan)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.niaosudan2.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=log10(SSC.C))) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.SSC.C2.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=DSCC)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.DSCC2.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=HIF)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.HIF2.pdf",width=25,height=25)

#######

x8 <- x8[x8$V2 %in% c("Early","Middle","Late"),]

ggplot(x8,aes(x=value,y=Parity)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.Parity3.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=DIM)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.DIM3.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=Milk.yield)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.Milk.yield3.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=ruzhilv)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.ruzhilv3.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=danbailv)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.danbailv3.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=rutang)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.rutang3.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=zongguti)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.zongguti3.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=tixibao)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.tixibao3.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=niaosudan)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.niaosudan3.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=log10(SSC.C))) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.SSC.C3.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=DSCC)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.DSCC3.pdf",width=25,height=25)

ggplot(x8,aes(x=value,y=HIF)) + geom\_point() + geom\_smooth(method="lm") + facet\_wrap(~celltype,scales = "free") + xlab("") + ylab("") + theme\_classic() + stat\_cor(method="pearson") #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

ggsave("Correlation.HIF3.pdf",width=25,height=25)

**ROC**

x <- read.table("Milk-423-deconvolution.txt")

table1 <- x[c("B140024D0725","B110623D0725","B130244D0725.2","B140108D0725","B110530D0725","B111162D0725","B130296D0725","B130299D0725","B140113D0725","B121578D0725","B130082D0725","B130283D0725","B14133D1226","B180556D0831","B180504D0831","B160252D0831","B160009D0901","B160331D0901","B160249D0827","B180257D0827","B180490D0827","B180548D0827","B180791D0827","B180855D0827","B170224D0901",

"B170396D0901","B170431D0901","B170810D0901","B170954D0901","B180212D0901","B180309D0901","B180462D0901","B180522D0901","B160056D0729","B170682D0729","B180495D0729","B15109D1225","B17151D0106","B18134D0106","B18289D0106","B16139D0712","B17285D0712","B17186D0712","B15218D0712","B14147D0712","B17136D0712","B17199D0712","B17081D0712","B17254D0712","B17092D0712","B15274D0712",

"B16111D0712","B17190D0712","B15191D0712","B17258D0712","B17214D0712","B17233D0712","B17154D0712","B17219D0712","B17184D0712","B17091D0712","B17325D0712","B14124D0712","B16128D0712","B15108D0712","B17225D0712","B17172D0712","B15245D0712","B17276D0712","B17164D0712","B17167D0712","B17086D0712","B15097D0712","B15212D0712","B17189D0712","B15122D0712","B17185D0712","B17087D0712",

"B17339D0712","B17337D0712","B16054D0712","B16150D0712","B17150D0712","B17331D0712","B16050D0712","B74datD0712","B17201D0712","B17283D0712","B17266D0712","B16076D0712","B16132D0712","B17314D0712","B15107D0712","B17263D0712","B15286D0712","B17212D0712","B17192D0712","B16139D0523","B17285D0523","B17186D0523","B15218D0523","B14147D0523","B17136D0523","B17199D0523","B17081D0523",

"B17254D0523","B17195D0523","B15274D0523","B16111D0523","B17324D0523","B15273D0523","B15191D0523","B17258D0523","B17214D0523","B17233D0523","B17154D0523","B17219D0523","B17184D0523","B17091D0523","B17325D0523","B14124D0523","B15108D0523","B17326D0523","B17001D0523","B17225D0523","B17172D0523","B15245D0523","B17276D0523","B17167D0523","B17086D0523","B15097D0523","B17189D0523",

"B17185D0523","B17087D0523","B17339D0523","B17117D0523","B17337D0523","B16152D0523","B17309D0523","B16054D0523","B16150D0523","B17150D0523","B17187D0523","B17331D0523","B16244D0523","B17201D0523","B17283D0523","B17266D0523","B16076D0523","B16132D0523","B15258D0523","B17314D0523","B15286D0523","B17212D0523","B17362D0523","B17192D0523","B17190D0524","B16050D0524","B16128D0620m",

"B17326D0620m","B17164D0620m","B17164D0620n","B17192D0620m","B17192D0620e","B150195D0725","B150543D0725","B150625D0725","B160085D0725","B160205D0725","B160578D0725","B160847D0725","B170511D0725"),]

table2 <- x[c("B170631D0725","B170678D0725","B170688D0725","B170821D0725","B180085D0725","B150235D0725","B150377D0725","B150420D0725","B160076D0725","B160254D0725","B160286D0725","B160544D0725","B160580D0725","B160601D0725","B160626D0725","B160720D0725","B160783D0725","B160858D0725","B160895D0725","B170012D0725","B170063D0725","B170506D0725","B170568D0725","B170620D0725","B170628D0725","B170718D0725",

"B170728D0725","B170738D0725","B170778D0725","B170855D0725","B170896D0725","B170916D0725","B170962D0725","B170985D0725","B180011D0725","B180015D0725","B160677D0725","B160796D0725","B160862D0725","B170519D0725","B170546D0725","B170658D0725","B17117D0620m","B17117D0620e","B17147","B17176","B17268","B17360","B18035","B18091","B18109","B15220D0625","B15274D0625","B17035D0625","B18035D0625",

"B19013D0625","B15273D1225","B16098D0106","B16139D0106","B16139D1225","B16252D0106","B17114D1225","B17131D1225","B17195D0106","B17354D1225","B18055D1225","B18091D1225","B18096D0106","B18112D0106","B18267D1225","B19101D0106","B19204D1225","B160206D0830","B160846D0830.1","B170033D0830","B170084D0830","B170455D0830","B170881D0830","B180160D0830","B180181D0830","B180229D0830","B180394D0830",

"B180396D0830","B180414D0830","B180432D0830","B180433D0830","B180466D0830","B160846D0830.2","B16139","B17186","B15218","B17081","B17254","B14156","B17195","B17092","B17205","B16111","B17190","B17258","B17233","B17184","B17172","B17167","B17086","B15097","B17189","B17185","B17337","B16152","B17309","B17187","B16050","B17283","B17266","B16132","B15258","B15286","B17212","B180057D0828","B180061D0828",

"B180075D0828","B180096D0828","B15322D0530","B16064D0530","B16173D0530","B17119D0530","B17129D0530","B17333D0530","B18015D0530","B16191D0530","B17103D0530","B18028D0530","B16125D0530","B18038D0530","B16020D0530","B16096D0530","B16265D0530","B17231D0530","B15181D0530","B15313D0530","B16101D0530","B16170D0530","B17030D0530","B17035D0530","B17101D0530","B150513D0725","B150524D0725","B160042D0725",

"B160173D0725","B160586D0725","B170374D0725","B170381D0725","B170683D0725","B170736D0725","B170823D0725","B170895D0725","B170948D0725","B170992D0725","B180008D0725","B180024D0725","B180038D0725","B180061D0725","B180096D0725","B180608D0725","B16257D0625","B17103D0625"),]

table\_group <- rbind(table1,table2)

Gene\_Expression\_log <- table\_group

#Gene\_Expression\_log <- t(Gene\_Expression\_log)

#set.seed(13031982)

#randomnumber <- matrix(exp(runif(7924045, log(1e-20),log(1e-10))),239,33155)

#Gene\_Expression\_log <- Gene\_Expression\_log + randomnumber

Gene\_Expression\_variance <- Gene\_Expression\_log[ , apply(Gene\_Expression\_log, 2, var) != 0]

pca <- prcomp((Gene\_Expression\_variance),scale. = TRUE)

pca$x

pca.var <- pca$sdev^2

pca.var.per <- round(pca.var/sum(pca.var)\*100,1)

sum(pca.var.per>0.1)

table <- pca$x

table <- table[,1:sum(pca.var.per>0.1)]

table <- data.frame(table)

table$name <- rownames(table)

#table$type <- c("blue","blue","blue","red","red","red","green","green","green")

genotype <- read.table("../../../423-2/genotype.txt")

table2 <- merge(table,genotype,by.x="name",by.y="V1")

table2 <- table2[!table2$V2 == "chu",]

rownames(table2) <- table2$name

table2 <- table2[,-1]

table2$V3 <- 0

table2$V3[table2$V2 %in% c("yan")] <- 1

table2$V3 <- factor(table2$V3)

dataset <- table2[,c(1:10,12)]

pvalues <- c()

for (i in 1:(ncol(dataset)-1)) {

#png(file = paste("var\_", i, ".png", sep=""))

a <- wilcox.test(dataset[, i] ~ dataset$V3)

pvalues <- c(pvalues,a$p.value)

#dev.off()

}

pvalues <- c(pvalues,0)

dataset <- dataset[,pvalues<0.05]

selectPC <- colnames(dataset)

train1 <- dataset[dataset$V3==0,]

train2 <- dataset[dataset$V3==1,]

trainsauc <- c()

testsauc <- c()

for (i in 1:100) {

train3 <- train1[sample(1:284,28),]

train <- rbind(train2,train3)

library(e1071)

library(caTools)

split <- sample.split(train$V3, SplitRatio = 0.5)

training\_set <- subset(train, split == TRUE)

test\_set <- subset(train, split == FALSE)

training\_scale <- training\_set

test\_scale <- test\_set

training\_scale[-4] = scale(training\_set[-4])

test\_scale[-4] = scale(test\_set[-4])

#install.packages('e1071')

classifier = svm(V3 ~ .,

data = training\_scale,

type = 'C-classification',

kernel = 'radial',

cost=1,

gamma=0.1,

probability=T)

par(mfrow=c(1,2))

y\_pred = predict(classifier, newdata = training\_scale[-4],probability=T)

probs <- attr(y\_pred,"probabilities")

library(pROC)

roc1<-roc(training\_scale$V3,probs[,2])

#plot(roc1,print.auc=TRUE,plot=TRUE,

# print.thres=TRUE)

trainsauc <- c(trainsauc,roc1$auc)

y\_pred = predict(classifier, newdata = test\_scale[-4],probability=T)

probs <- attr(y\_pred,"probabilities")

library(pROC)

roc1<-roc(test\_scale$V3,probs[,2])

#plot(roc1,print.auc=TRUE,plot=TRUE,

# print.thres=TRUE)

testsauc <- c(testsauc,roc1$auc)

}

x <- cbind(trainsauc,testsauc)

boxplot(x)

############################################################

############################################################

############################################################

x <- read.table("Milk-423-deconvolution.txt")

table1 <- x[c("B140024D0725","B110623D0725","B130244D0725.2","B140108D0725","B110530D0725","B111162D0725","B130296D0725","B130299D0725","B140113D0725","B121578D0725","B130082D0725","B130283D0725","B14133D1226","B180556D0831","B180504D0831","B160252D0831","B160009D0901","B160331D0901","B160249D0827","B180257D0827","B180490D0827","B180548D0827","B180791D0827","B180855D0827","B170224D0901",

"B170396D0901","B170431D0901","B170810D0901","B170954D0901","B180212D0901","B180309D0901","B180462D0901","B180522D0901","B160056D0729","B170682D0729","B180495D0729","B15109D1225","B17151D0106","B18134D0106","B18289D0106","B16139D0712","B17285D0712","B17186D0712","B15218D0712","B14147D0712","B17136D0712","B17199D0712","B17081D0712","B17254D0712","B17092D0712","B15274D0712",

"B16111D0712","B17190D0712","B15191D0712","B17258D0712","B17214D0712","B17233D0712","B17154D0712","B17219D0712","B17184D0712","B17091D0712","B17325D0712","B14124D0712","B16128D0712","B15108D0712","B17225D0712","B17172D0712","B15245D0712","B17276D0712","B17164D0712","B17167D0712","B17086D0712","B15097D0712","B15212D0712","B17189D0712","B15122D0712","B17185D0712","B17087D0712",

"B17339D0712","B17337D0712","B16054D0712","B16150D0712","B17150D0712","B17331D0712","B16050D0712","B74datD0712","B17201D0712","B17283D0712","B17266D0712","B16076D0712","B16132D0712","B17314D0712","B15107D0712","B17263D0712","B15286D0712","B17212D0712","B17192D0712","B16139D0523","B17285D0523","B17186D0523","B15218D0523","B14147D0523","B17136D0523","B17199D0523","B17081D0523",

"B17254D0523","B17195D0523","B15274D0523","B16111D0523","B17324D0523","B15273D0523","B15191D0523","B17258D0523","B17214D0523","B17233D0523","B17154D0523","B17219D0523","B17184D0523","B17091D0523","B17325D0523","B14124D0523","B15108D0523","B17326D0523","B17001D0523","B17225D0523","B17172D0523","B15245D0523","B17276D0523","B17167D0523","B17086D0523","B15097D0523","B17189D0523",

"B17185D0523","B17087D0523","B17339D0523","B17117D0523","B17337D0523","B16152D0523","B17309D0523","B16054D0523","B16150D0523","B17150D0523","B17187D0523","B17331D0523","B16244D0523","B17201D0523","B17283D0523","B17266D0523","B16076D0523","B16132D0523","B15258D0523","B17314D0523","B15286D0523","B17212D0523","B17362D0523","B17192D0523","B17190D0524","B16050D0524","B16128D0620m",

"B17326D0620m","B17164D0620m","B17164D0620n","B17192D0620m","B17192D0620e","B150195D0725","B150543D0725","B150625D0725","B160085D0725","B160205D0725","B160578D0725","B160847D0725","B170511D0725"),]

table2 <- x[c("B170631D0725","B170678D0725","B170688D0725","B170821D0725","B180085D0725","B150235D0725","B150377D0725","B150420D0725","B160076D0725","B160254D0725","B160286D0725","B160544D0725","B160580D0725","B160601D0725","B160626D0725","B160720D0725","B160783D0725","B160858D0725","B160895D0725","B170012D0725","B170063D0725","B170506D0725","B170568D0725","B170620D0725","B170628D0725","B170718D0725",

"B170728D0725","B170738D0725","B170778D0725","B170855D0725","B170896D0725","B170916D0725","B170962D0725","B170985D0725","B180011D0725","B180015D0725","B160677D0725","B160796D0725","B160862D0725","B170519D0725","B170546D0725","B170658D0725","B17117D0620m","B17117D0620e","B17147","B17176","B17268","B17360","B18035","B18091","B18109","B15220D0625","B15274D0625","B17035D0625","B18035D0625",

"B19013D0625","B15273D1225","B16098D0106","B16139D0106","B16139D1225","B16252D0106","B17114D1225","B17131D1225","B17195D0106","B17354D1225","B18055D1225","B18091D1225","B18096D0106","B18112D0106","B18267D1225","B19101D0106","B19204D1225","B160206D0830","B160846D0830.1","B170033D0830","B170084D0830","B170455D0830","B170881D0830","B180160D0830","B180181D0830","B180229D0830","B180394D0830",

"B180396D0830","B180414D0830","B180432D0830","B180433D0830","B180466D0830","B160846D0830.2","B16139","B17186","B15218","B17081","B17254","B14156","B17195","B17092","B17205","B16111","B17190","B17258","B17233","B17184","B17172","B17167","B17086","B15097","B17189","B17185","B17337","B16152","B17309","B17187","B16050","B17283","B17266","B16132","B15258","B15286","B17212","B180057D0828","B180061D0828",

"B180075D0828","B180096D0828","B15322D0530","B16064D0530","B16173D0530","B17119D0530","B17129D0530","B17333D0530","B18015D0530","B16191D0530","B17103D0530","B18028D0530","B16125D0530","B18038D0530","B16020D0530","B16096D0530","B16265D0530","B17231D0530","B15181D0530","B15313D0530","B16101D0530","B16170D0530","B17030D0530","B17035D0530","B17101D0530","B150513D0725","B150524D0725","B160042D0725",

"B160173D0725","B160586D0725","B170374D0725","B170381D0725","B170683D0725","B170736D0725","B170823D0725","B170895D0725","B170948D0725","B170992D0725","B180008D0725","B180024D0725","B180038D0725","B180061D0725","B180096D0725","B180608D0725","B16257D0625","B17103D0625"),]

table\_group <- rbind(table1,table2)

genotype <- read.table("../../../423-2/genotype.txt")

table\_group$name <- rownames(table\_group)

table\_group <- merge(table\_group,genotype,by.x="name",by.y="V1")

table\_group <- table\_group[!table\_group$V2 %in% c("chu"),]

train <- table\_group

train$V3 <- 0

train$V3[train$V2=="yan"] <- 1

train$V3 <- factor(train$V3)

rownames(train) <- train$name

train <- train[,c(2:12,14)]

train1 <- train[train$V3==0,]

train2 <- train[train$V3==1,]

trainsauc <- c()

testsauc <- c()

train3 <- train1

train <- rbind(train2,train3)

library(e1071)

library(caTools)

#for (i in 1:100) {

set.seed(92)

split <- sample.split(train$V3, SplitRatio = 0.75)

training\_set <- subset(train, split == TRUE)

test\_set <- subset(train, split == FALSE)

training\_scale <- training\_set

test\_scale <- test\_set

training\_scale[-12] = scale(training\_set[-12])

test\_scale[-12] = scale(test\_set[-12])

#install.packages('e1071')

classifier = svm(V3 ~ .,

data = training\_scale,

type = 'C-classification',

kernel = 'radial',

cost=1,

gamma=0.1,

probability=T)

pdf(paste("All-",i,".pdf",sep=""),width = 10,height = 5)

par(mfrow=c(1,2))

y\_pred = predict(classifier, newdata = training\_scale[-12],probability=T)

probs <- attr(y\_pred,"probabilities")

library(pROC)

roc1<-roc(training\_scale$V3,probs[,2])

plot(roc1,print.auc=TRUE,plot=TRUE,

print.thres=TRUE)

trainsauc <- c(trainsauc,roc1$auc)

y\_pred = predict(classifier, newdata = test\_scale[-12],probability=T)

probs <- attr(y\_pred,"probabilities")

library(pROC)

roc1<-roc(test\_scale$V3,probs[,2])

plot(roc1,print.auc=TRUE,plot=TRUE,

print.thres=TRUE)

dev.off()

testsauc <- c(testsauc,roc1$auc)

#}

x <- cbind(trainsauc,testsauc)

boxplot(x)

x <- data.frame(x)

x$mul <- x$trainsauc \* x$testsauc

s <- data.frame(cbind(test\_scale$V3,probs[,2]))

boxplot(s$X2[s$X1==1],s$X2[s$X1==2])

#############

**ROC2**

######################

library(ggplot2)

pheno <- read.table("../../pheno.txt",sep="\t",header = T)

pheno <- pheno[!pheno$group2 %in% c("m","n","e"),]

ggplot(pheno,aes(x=group2,y=(DSCC))) + geom\_boxplot(outlier.shape = NA) + xlab("") + ylab("DSCC") + theme\_classic() #+ theme(axis.text.x = element\_text(angle = 45,hjust = 1))

library(reshape2)

pheno2 <- melt(pheno)

ggplot(pheno2,aes(x=group2,y=log10(value))) + geom\_boxplot(outlier.shape = NA) + facet\_wrap(~variable,scales = "free") + xlab("") + ylab("DSCC") + theme\_classic() + theme(axis.text.x = element\_text(angle = 45,hjust = 1))

rownames(pheno) <- pheno$ID

pheno <- pheno[,c(11,14,3)]

pheno\_mas <- na.omit(pheno[pheno$group2 %in% c("Mastitis"),])

pheno\_chu <- na.omit(pheno[pheno$group2 %in% c("Colostrum"),])

pheno\_other <- na.omit(pheno[!pheno$group2 %in% c("Colostrum","Mastitis"),])

library(e1071)

library(caTools)

TPRs <- c()

FPRs <- c()

pheno\_mas$V2 <- 1

pheno\_other$V2 <- 0

pheno3 <- rbind(pheno\_mas,pheno\_other)

ggplot(pheno3,aes(x=tixibao,y=DSCC,col=V2)) + geom\_point() + theme\_classic()

par(mfrow=c(1,2))

roc1<-roc((pheno3$V2),(pheno3$DSCC))

plot(roc1,print.auc=TRUE,plot=TRUE,

print.thres=TRUE)

roc1<-roc((pheno3$V2),(pheno3$tixibao))

plot(roc1,print.auc=TRUE,plot=TRUE,

print.thres=TRUE)

scRNA-corelation

x1 <- read.table("Cow.CPM.100.Average.txt",header = T)

x2 <- read.table("Milk.CPM.100.Average.txt",header = T)

x <- merge(x1,x2,by="gene")

cor.test(log2(x$Endothelialcell.x+1),log2(x$Endothelialcell.y+1),method="spearman")

cor.test(log2(x$Macrophage.x+1),log2(x$Macrophage.y+1),method="spearman")

cor.test(log2(x$NKTcell.x+1),log2(x$NKTcell.y+1),method="spearman")

cor.test(log2(x$Dendriticcell.x+1),log2(x$Dendriticcell.y+1),method="spearman")

cor.test(log2(x$CD8Tcell.x+1),log2(x$CD8Tcell.y+1),method="spearman")

cor.test(log2(x$CD4Tcell.x+1),log2(x$CD4Tcell.y+1),method="spearman")

cor.test(log2(x$Mammaryepithelialcellluminal.x+1),log2(x$Mammaryepithelialcellluminal.y+1),method="spearman")

cor.test(log2(x$MammaryepithelialcellAlveolar.x+1),log2(x$MammaryepithelialcellAlveolar.y+1),method="spearman")

cor.test(log2(x$Monocyte.x+1),log2(x$Monocyte.y+1),method="spearman")

cor.test(log2(x$GammaDeltaTcell.x+1),log2(x$GammaDeltaTcell.y+1),method="spearman")

cor.test(log2(x$Endothelialcell.x+1),log2(x$Endothelialcell.y+1),method="pearson")

cor.test(log2(x$Macrophage.x+1),log2(x$Macrophage.y+1),method="pearson")

cor.test(log2(x$NKTcell.x+1),log2(x$NKTcell.y+1),method="pearson")

cor.test(log2(x$Dendriticcell.x+1),log2(x$Dendriticcell.y+1),method="pearson")

cor.test(log2(x$CD8Tcell.x+1),log2(x$CD8Tcell.y+1),method="pearson")

cor.test(log2(x$CD4Tcell.x+1),log2(x$CD4Tcell.y+1),method="pearson")

cor.test(log2(x$Mammaryepithelialcellluminal.x+1),log2(x$Mammaryepithelialcellluminal.y+1),method="pearson")

cor.test(log2(x$MammaryepithelialcellAlveolar.x+1),log2(x$MammaryepithelialcellAlveolar.y+1),method="pearson")

cor.test(log2(x$Monocyte.x+1),log2(x$Monocyte.y+1),method="pearson")

cor.test(log2(x$GammaDeltaTcell.x+1),log2(x$GammaDeltaTcell.y+1),method="pearson")

correlation <- cor(as.matrix(log2(x[,2:25]+1)),method = "spearman")

#correlation2 <- cor(as.matrix(log2(x[,2:25]+1)))

#write.table(correlation,"Correlation.txt",sep="\t",quote=F,row.names = T,col.names = T)

library(ggpubr)

g1 <- ggplot(x, aes(x=log2(Endothelialcell.x+1), y=log2(Endothelialcell.y+1))) + geom\_point() #+ theme(legend.position = "none") #+

g1 <- g1 + xlab("Endothelialcell-Cow") + ylab("Endothelialcell-Milk")

g1 <- g1 + theme\_classic() + scale\_y\_continuous(limits = c(0,10)) + scale\_x\_continuous(limits = c(0,10)) + stat\_cor(method="spearman")

g2 <- ggplot(x, aes(x=log2(Macrophage.x+1), y=log2(Macrophage.y+1))) + geom\_point() #+ theme(legend.position = "none") #+

g2 <- g2 + xlab("Macrophage-Cow") + ylab("Macrophage-Milk")

g2 <- g2 + theme\_classic() + scale\_y\_continuous(limits = c(0,9)) + scale\_x\_continuous(limits = c(0,9)) + stat\_cor(method="spearman")

g3 <- ggplot(x, aes(x=log2(NKTcell.x+1), y=log2(NKTcell.y+1))) + geom\_point() #+ theme(legend.position = "none") #+

g3 <- g3 + xlab("NKTcell-Cow") + ylab("NKTcell-Milk")

g3 <- g3 + theme\_classic() + scale\_y\_continuous(limits = c(0,10)) + scale\_x\_continuous(limits = c(0,10)) + stat\_cor(method="spearman")

g4 <- ggplot(x, aes(x=log2(Dendriticcell.x+1), y=log2(Dendriticcell.y+1))) + geom\_point() #+ theme(legend.position = "none") #+

g4 <- g4 + xlab("Dendriticcell-Cow") + ylab("Dendriticcell-Milk")

g4 <- g4 + theme\_classic() + scale\_y\_continuous(limits = c(0,9)) + scale\_x\_continuous(limits = c(0,9)) + stat\_cor(method="spearman")

g5 <- ggplot(x, aes(x=log2(CD8Tcell.x+1), y=log2(CD8Tcell.y+1))) + geom\_point() #+ theme(legend.position = "none") #+

g5 <- g5 + xlab("CD8Tcell-Cow") + ylab("CD8Tcell-Milk")

g5 <- g5 + theme\_classic() + scale\_y\_continuous(limits = c(0,10)) + scale\_x\_continuous(limits = c(0,10)) + stat\_cor(method="spearman")

g6 <- ggplot(x, aes(x=log2(CD4Tcell.x+1), y=log2(CD4Tcell.y+1))) + geom\_point() #+ theme(legend.position = "none") #+

g6 <- g6 + xlab("CD4Tcell-Cow") + ylab("CD4Tcell-Milk")

g6 <- g6 + theme\_classic() + scale\_y\_continuous(limits = c(0,10)) + scale\_x\_continuous(limits = c(0,10)) + stat\_cor(method="spearman")

g7 <- ggplot(x, aes(x=log2(Mammaryepithelialcellluminal.x+1), y=log2(Mammaryepithelialcellluminal.y+1))) + geom\_point() #+ theme(legend.position = "none") #+

g7 <- g7 + xlab("Mammaryepithelialcellluminal-Cow") + ylab("Mammaryepithelialcellluminal-Milk")

g7 <- g7 + theme\_classic() + scale\_y\_continuous(limits = c(0,10)) + scale\_x\_continuous(limits = c(0,10)) + stat\_cor(method="spearman")

g8 <- ggplot(x, aes(x=log2(MammaryepithelialcellAlveolar.x+1), y=log2(MammaryepithelialcellAlveolar.y+1))) + geom\_point() #+ theme(legend.position = "none") #+

g8 <- g8 + xlab("MammaryepithelialcellAlveolar-Cow") + ylab("MammaryepithelialcellAlveolar-Milk")

g8 <- g8 + theme\_classic() + scale\_y\_continuous(limits = c(0,12)) + scale\_x\_continuous(limits = c(0,12)) + stat\_cor(method="spearman")

g9 <- ggplot(x, aes(x=log2(Monocyte.x+1), y=log2(Monocyte.y+1))) + geom\_point() #+ theme(legend.position = "none") #+

g9 <- g9 + xlab("Monocyte-Cow") + ylab("Monocyte-Milk")

g9 <- g9 + theme\_classic() + scale\_y\_continuous(limits = c(0,9)) + scale\_x\_continuous(limits = c(0,9)) + stat\_cor(method="spearman")

g10 <- ggplot(x, aes(x=log2(GammaDeltaTcell.x+1), y=log2(GammaDeltaTcell.y+1))) + geom\_point() #+ theme(legend.position = "none") #+

g10 <- g10 + xlab("GammaDeltaTcell-Cow") + ylab("GammaDeltaTcell-Milk")

g10 <- g10 + theme\_classic() + scale\_y\_continuous(limits = c(0,9)) + scale\_x\_continuous(limits = c(0,9)) + stat\_cor(method="pearson")

ggarrange(g1,g2,g3,g4,g5,g6,g7,g8,g9,g10,

ncol = 5, nrow = 2)

#library(corrplot)

#col1 <- colorRampPalette(c("green","yellow","blue","white","red"))

#corrplot(cor(as.matrix(log2(x[,2:25]+1)),method = "spearman"),col=col1(51),method = "color",order = "original",cl.lim = c(0.5,1),is.corr=FALSE,tl.col = "black",addCoef.col="black")

**scRNA-cow**

library(Seurat)

library(dplyr)

library(AnnotationHub)

####################### Loading data ###################################################################################

projName='Cow1'

dataDir <- "Cow1\_filtered\_feature\_bc\_matrix/"

milk.data <- Read10X(data.dir = dataDir)

milk.data2 <- milk.data

mtgene <- read.table("transcriptID2GeneSymbol.MT.txt")

milk.data.name <- data.frame(gene=rownames(milk.data2))

suple <- milk.data.name[milk.data.name$gene %in% mtgene$V2,]

milk.data.name$gene[milk.data.name$gene %in% mtgene$V2] <- paste("MT-",suple,sep="")

rownames(milk.data2) <- milk.data.name$gene

####################### Create seurat object, and filter genes with min.cells = 10 and min.features = 200 ##############

milk <- CreateSeuratObject(counts = milk.data, project = "Milkcells", min.cells = 10, min.features = 200)

milk2 <- CreateSeuratObject(counts = milk.data2, project = "Milkcells", min.cells = 10, min.features = 200)

####################### Quality control ################################################################################

#number of genes detected per UMI

#milk$GenesPerUMI <- milk$nFeature\_RNA / milk$nCount\_RNA

#complexity

#milk$log10GenesPerUMI <- log10(milk$nFeature\_RNA) / log10(milk$nCount\_RNA)

#proportion of transcripts mapping to mitochondrial genes

milk$percent.mito <- PercentageFeatureSet(object = milk2, pattern = "^MT-")

milk$mitoRatio <- milk@meta.data$percent.mito / 100

#Plot number of genes, number of UMIs and mitoRatio distribution before QC

VlnPlot(milk, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mito"),ncol = 3, pt.size = 0.005, cols="dodgerblue")

#Filtering based on QC

nGene = 500

nUMI = 1000

mt = 20

milk = subset(milk, subset = nFeature\_RNA > nGene & nCount\_RNA > nUMI & nCount\_RNA < 5e4 & percent.mito < mt)

#Plot number of genes, number of UMIs and mitoRatio distribution after QC

VlnPlot(milk, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mito"),ncol = 3, pt.size = 0.005, cols="dodgerblue")

cow1 <- milk

####################### Loading data ###################################################################################

projName='Cow1'

dataDir <- "Cow2\_filtered\_feature\_bc\_matrix/"

milk.data <- Read10X(data.dir = dataDir)

milk.data2 <- milk.data

mtgene <- read.table("transcriptID2GeneSymbol.MT.txt")

milk.data.name <- data.frame(gene=rownames(milk.data2))

suple <- milk.data.name[milk.data.name$gene %in% mtgene$V2,]

milk.data.name$gene[milk.data.name$gene %in% mtgene$V2] <- paste("MT-",suple,sep="")

rownames(milk.data2) <- milk.data.name$gene

####################### Create seurat object, and filter genes with min.cells = 10 and min.features = 200 ##############

milk <- CreateSeuratObject(counts = milk.data, project = "Milkcells", min.cells = 10, min.features = 200)

milk2 <- CreateSeuratObject(counts = milk.data2, project = "Milkcells", min.cells = 10, min.features = 200)

####################### Quality control ################################################################################

#number of genes detected per UMI

#milk$GenesPerUMI <- milk$nFeature\_RNA / milk$nCount\_RNA

#complexity

#milk$log10GenesPerUMI <- log10(milk$nFeature\_RNA) / log10(milk$nCount\_RNA)

#proportion of transcripts mapping to mitochondrial genes

milk$percent.mito <- PercentageFeatureSet(object = milk2, pattern = "^MT-")

milk$mitoRatio <- milk@meta.data$percent.mito / 100

#Plot number of genes, number of UMIs and mitoRatio distribution before QC

VlnPlot(milk, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mito"),ncol = 3, pt.size = 0.005, cols="dodgerblue")

#Filtering based on QC

nGene = 500

nUMI = 1000

mt = 20

milk = subset(milk, subset = nFeature\_RNA > nGene & nCount\_RNA > nUMI & nCount\_RNA < 5e4 & percent.mito < mt)

#Plot number of genes, number of UMIs and mitoRatio distribution after QC

VlnPlot(milk, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mito"),ncol = 3, pt.size = 0.005, cols="dodgerblue")

cow2 <- milk

###############

#cow1 <- FindVariableFeatures(cow1, selection.method = "vst", nfeatures = 2000)

#cow2 <- FindVariableFeatures(cow2, selection.method = "vst", nfeatures = 2000)

cow1[["group"]] <- "Cow1"

cow2[["group"]] <- "Cow2"

#pbmc\_cca <- RunCCA(object1 = cow1, object2 = cow2)

cow1 <- NormalizeData(cow1)

cow1 <- FindVariableFeatures(cow1, selection.method = "vst", nfeatures = 2000)

cow2 <- NormalizeData(cow2)

cow2 <- FindVariableFeatures(cow2, selection.method = "vst", nfeatures = 2000)

features <- SelectIntegrationFeatures(object.list = c(cow1,cow2))

immune.anchors <- FindIntegrationAnchors(object.list = c(cow1,cow2), anchor.features = features)

# this command creates an 'integrated' data assay

immune.combined <- IntegrateData(anchorset = immune.anchors)

# specify that we will perform downstream analysis on the corrected data note that the

# original unmodified data still resides in the 'RNA' assay

DefaultAssay(immune.combined) <- "integrated"

# Run the standard workflow for visualization and clustering

immune.combined <- ScaleData(immune.combined)

immune.combined <- RunPCA(immune.combined)

immune.combined <- RunUMAP(immune.combined, dims = 1:20)

immune.combined <- RunTSNE(immune.combined, dims = 1:20)

immune.combined <- FindNeighbors(immune.combined, dims = 1:20)

immune.combined <- FindClusters(immune.combined, resolution = 0.8)

DimPlot(immune.combined, label = T)

DimPlot(immune.combined, label = T, reduction = 'tsne')

# Visualization

p1 <- DimPlot(immune.combined, reduction = "tsne", group.by = "group")

p2 <- DimPlot(immune.combined, reduction = "tsne", label = TRUE, repel = TRUE)

p1 + p2

DefaultAssay(immune.combined) <- "RNA"

save(immune.combined, file = "Cow.combined.RData")

features <- c("IL7R","RGCC","CXCR4","DUSP1","CD69","NR4A1")

features2 <- c("XCL2","XCL1","CRTAM","CD7","CTSW","PIK3R1","NKG7","CD96")

features3 <- c("CD7","CD96","CTSW","NKG7")

features4 <- c("IL7R","CD52","BCL11B","CD3E","GIMAP7","LTB")

features5 <- c("GZMK","CCL5","CD8B")

features6 <- c("APOE","C1QA","BOLA-DRA","C1QC","BOLA-DRB3","CFD","CXCL8","TYROBP","AIF1","RNASE6","CTSH","C1QB","TREM2")

features7 <- c("VWF","ACKR1","PECAM1","PLVAP","AQP1","EGFL7","CLEC3B","ENG","LIFR","CLDN5")

features8 <- c("CSN3","CSN2","CSN1S1","GLYCAM1","LTF","CSN1S2","LALBA","CD24","ELF5","KRT19","CLDN3")

features9 <- c("CD24","PRLR","KRT19","KRT7","KRT18","KRT8","CLDN3")

features10 <- c("CCL21","MMRN1","CAVIN2","STAB1","GNG11","PROX1","CAV1")

features11 <- c("LCN2","WFDC2","KRT19","KRT7","KRT15","CLDN3","CLDN4","EPCAM","LTF")

features12 <- c("BOLA-DQA2","PLD4","PKIB","BOLA-DMA","CST3","CD74","CD83")

features13 <- c("TAGLN","ACTA2","TPM2","MYH11","SPARCL1","MUSTN1","IGFBP6","DSTN","CALD1","MYL9","TPM1")

features14 <- c("BATF3","BASP1","LYZ","PLD4","LGALS1","CD74","CD83")

features15 <- c("NR4A2","PTPRC","FOSB","TRPM2","GPR174","SLC4A7","CD69","LCP2","ZAP70")

features16 <- c("SOX18","GPIHBP1","SEMA3G","CLEC14A","FBLN2","TCF4")

features17 <- c("COL3A1","CXCL12","COL1A1","COL1A2","DCN","COL6A1","AEBP1","SPARC","SERPING1","IGFBP7")

features18 <- c("ACTA2","IGFBP5","CNN1","MYLK","MYH11","TPM2","TAGLN","MYL9","DSTN","CALD1","TPM1")

features19 <- c("TOP2A","CENPF","MKI67","UBE2C","STMN1","TUBB","TUBA1B","HMGB2")

features20 <- c("MMP12","MMP9","CD68","RARRES1","TYROBP","GRN")

library(ggplot2)

FeaturePlot(immune.combined, features = features,reduction = "tsne")

ggsave("Cow0.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features2,reduction = "tsne")

ggsave("Cow1.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features3,reduction = "tsne")

ggsave("Cow2.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features4,reduction = "tsne")

ggsave("Cow3.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features5,reduction = "tsne")

ggsave("Cow4.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features6,reduction = "tsne")

ggsave("Cow5.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features7,reduction = "tsne")

ggsave("Cow6.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features8,reduction = "tsne")

ggsave("Cow7.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features9,reduction = "tsne")

ggsave("Cow8.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features10,reduction = "tsne")

ggsave("Cow9.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features11,reduction = "tsne")

ggsave("Cow10.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features12,reduction = "tsne")

ggsave("Cow11.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features13,reduction = "tsne")

ggsave("Cow12.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features14,reduction = "tsne")

ggsave("Cow13.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features15,reduction = "tsne")

ggsave("Cow14.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features16,reduction = "tsne")

ggsave("Cow15.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features17,reduction = "tsne")

ggsave("Cow16.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features18,reduction = "tsne")

ggsave("Cow17.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features19,reduction = "tsne")

ggsave("Cow18.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features20,reduction = "tsne")

ggsave("Cow19.pdf",width=10,height=10)

milk.markers <- FindAllMarkers(immune.combined, only.pos = TRUE, min.pct = 0.1, logfc.threshold = 0.25)

#Top10 cluster genes

top20 <- milk.markers %>% group\_by(cluster) %>% top\_n(n = 20, wt = avg\_log2FC)

#VlnPlot(immune.combined, features = c("CD7", "CD96","CD244","XCL2","XCL1","RGS2.1"))

write.table(top20,"Cow.marker.txt",col.names = T,row.names = F,sep="\t",quote=F)

#######

#data.raw <- data.frame(GetAssayData(immune.combined))

#data.cluster <- immune.combined@meta.data

DefaultAssay(immune.combined) <- "RNA"

#immune.combined2 <- NormalizeData(immune.combined,scale.factor = 1e6)

log\_data <- GetAssayData(immune.combined)

log2\_data <- expm1(log\_data)

log2\_data <- data.frame(log2\_data)

set.seed(10000)

log2\_data\_out <- log2\_data[,sample(1:15628,2000)]

data.cluster <- immune.combined@meta.data

data.cluster$sample <- gsub("-",".",rownames(data.cluster))

ss <- data.frame(sample=colnames(log2\_data\_out))

ss <- ss %>% left\_join(data.cluster,by="sample")

ss <- ss[,c(1,9)]

ss$seurat\_clusters <- paste("ClusterCow",ss$seurat\_clusters,sep="")

log2\_data\_out$gene <- rownames(log2\_data\_out)

log2\_data\_out <- log2\_data\_out[,c(2001,1:2000)]

ss2 <- c("gene",ss$seurat\_clusters)

log2\_data\_out <- rbind(ss2,log2\_data\_out)

write.table(log2\_data\_out,"Cow.CPM.100.2000.txt",col.names = F,row.names = F,sep="\t",quote=F)

#######

#data.raw <- data.frame(GetAssayData(immune.combined))

#data.cluster <- immune.combined@meta.data

DefaultAssay(immune.combined) <- "RNA"

#immune.combined2 <- NormalizeData(immune.combined,scale.factor = 1e6)

log\_data <- GetAssayData(immune.combined)

log2\_data <- expm1(log\_data)

log2\_data <- data.frame(log2\_data)

set.seed(10000)

log2\_data\_out <- log2\_data[,sample(1:15628,15628)]

data.cluster <- immune.combined@meta.data

data.cluster$sample <- gsub("-",".",rownames(data.cluster))

ss <- data.frame(sample=colnames(log2\_data\_out))

ss <- ss %>% left\_join(data.cluster,by="sample")

ss <- ss[,c(1,9)]

ss$seurat\_clusters <- paste("ClusterCow",ss$seurat\_clusters,sep="")

ss$celltype <- ""

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow4")] <- "CD8Tcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow14")] <- "CD4Tcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow13")] <- "Monocyte"

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow5","ClusterCow19")] <- "Macrophage"

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow15")] <- "Stemcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow6","ClusterCow9")] <- "Endothelialcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow8","ClusterCow10")] <- "Mammaryepithelialcellluminal"

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow0","ClusterCow1","ClusterCow2","ClusterCow3")] <- "NKTcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow11")] <- "Dendriticcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow7")] <- "MammaryepithelialcellAlveolar"

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow18")] <- "GammaDeltaTcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow12","ClusterCow17")] <- "Mammaryepithelialcellbasal"

ss$celltype[ss$seurat\_clusters %in% c("ClusterCow16")] <- "Stromalcells"

log2\_data\_out$gene <- rownames(log2\_data\_out)

log2\_data\_out <- log2\_data\_out[,c(15629,1:15628)]

ss2 <- c("gene",ss$celltype)

log2\_data\_out <- rbind(ss2,log2\_data\_out)

write.table(log2\_data\_out,"Cow.CPM.100.15628.txt",col.names = F,row.names = F,sep="\t",quote=F)

library(reshape2)

log2\_data\_out\_2 <- melt(log2\_data\_out)

**scRNA-milk**

library(Seurat)

library(dplyr)

library(AnnotationHub)

####################### Loading data ###################################################################################

projName='Cow1'

dataDir <- "milkcell1\_filtered\_feature\_bc\_matrix/"

milk.data <- Read10X(data.dir = dataDir)

milk.data2 <- milk.data

mtgene <- read.table("transcriptID2GeneSymbol.MT.txt")

milk.data.name <- data.frame(gene=rownames(milk.data2))

suple <- milk.data.name[milk.data.name$gene %in% mtgene$V2,]

milk.data.name$gene[milk.data.name$gene %in% mtgene$V2] <- paste("MT-",suple,sep="")

rownames(milk.data2) <- milk.data.name$gene

####################### Create seurat object, and filter genes with min.cells = 10 and min.features = 200 ##############

milk <- CreateSeuratObject(counts = milk.data, project = "Milkcells", min.cells = 10, min.features = 200)

milk2 <- CreateSeuratObject(counts = milk.data2, project = "Milkcells", min.cells = 10, min.features = 200)

####################### Quality control ################################################################################

#number of genes detected per UMI

#milk$GenesPerUMI <- milk$nFeature\_RNA / milk$nCount\_RNA

#complexity

#milk$log10GenesPerUMI <- log10(milk$nFeature\_RNA) / log10(milk$nCount\_RNA)

#proportion of transcripts mapping to mitochondrial genes

milk$percent.mito <- PercentageFeatureSet(object = milk2, pattern = "^MT-")

milk$mitoRatio <- milk@meta.data$percent.mito / 100

#Plot number of genes, number of UMIs and mitoRatio distribution before QC

VlnPlot(milk, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mito"),ncol = 3, pt.size = 0.005, cols="dodgerblue")

#Filtering based on QC

nGene = 500

nUMI = 1000

mt = 20

milk = subset(milk, subset = nFeature\_RNA > nGene & nCount\_RNA > nUMI & nCount\_RNA < 1e5 & percent.mito < mt)

#Plot number of genes, number of UMIs and mitoRatio distribution after QC

VlnPlot(milk, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mito"),ncol = 3, pt.size = 0.005, cols="dodgerblue")

cow1 <- milk

###############################

library(Seurat)

library(dplyr)

library(AnnotationHub)

####################### Loading data ###################################################################################

projName='Cow1'

dataDir <- "milkcell2\_filtered\_feature\_bc\_matrix/"

milk.data <- Read10X(data.dir = dataDir)

milk.data2 <- milk.data

mtgene <- read.table("transcriptID2GeneSymbol.MT.txt")

milk.data.name <- data.frame(gene=rownames(milk.data2))

suple <- milk.data.name[milk.data.name$gene %in% mtgene$V2,]

milk.data.name$gene[milk.data.name$gene %in% mtgene$V2] <- paste("MT-",suple,sep="")

rownames(milk.data2) <- milk.data.name$gene

####################### Create seurat object, and filter genes with min.cells = 10 and min.features = 200 ##############

milk <- CreateSeuratObject(counts = milk.data, project = "Milkcells", min.cells = 10, min.features = 200)

milk2 <- CreateSeuratObject(counts = milk.data2, project = "Milkcells", min.cells = 10, min.features = 200)

####################### Quality control ################################################################################

#number of genes detected per UMI

#milk$GenesPerUMI <- milk$nFeature\_RNA / milk$nCount\_RNA

#complexity

#milk$log10GenesPerUMI <- log10(milk$nFeature\_RNA) / log10(milk$nCount\_RNA)

#proportion of transcripts mapping to mitochondrial genes

milk$percent.mito <- PercentageFeatureSet(object = milk2, pattern = "^MT-")

milk$mitoRatio <- milk@meta.data$percent.mito / 100

#Plot number of genes, number of UMIs and mitoRatio distribution before QC

VlnPlot(milk, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mito"),ncol = 3, pt.size = 0.005, cols="dodgerblue")

#Filtering based on QC

nGene = 1000

nUMI = 1000

mt = 20

milk = subset(milk, subset = nFeature\_RNA > nGene & nCount\_RNA > nUMI & nCount\_RNA < 1e5 & percent.mito < mt)

#Plot number of genes, number of UMIs and mitoRatio distribution after QC

VlnPlot(milk, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mito"),ncol = 3, pt.size = 0.005, cols="dodgerblue")

cow2 <- milk

###############

cow1[["group"]] <- "Milk1"

cow2[["group"]] <- "Milk2"

#pbmc\_cca <- RunCCA(object1 = cow1, object2 = cow2)

cow1 <- NormalizeData(cow1)

cow1 <- FindVariableFeatures(cow1, selection.method = "vst", nfeatures = 2000)

cow2 <- NormalizeData(cow2)

cow2 <- FindVariableFeatures(cow2, selection.method = "vst", nfeatures = 2000)

features <- SelectIntegrationFeatures(object.list = c(cow1,cow2))

immune.anchors <- FindIntegrationAnchors(object.list = c(cow1,cow2), anchor.features = features)

# this command creates an 'integrated' data assay

immune.combined <- IntegrateData(anchorset = immune.anchors)

# specify that we will perform downstream analysis on the corrected data note that the

# original unmodified data still resides in the 'RNA' assay

DefaultAssay(immune.combined) <- "integrated"

# Run the standard workflow for visualization and clustering

immune.combined <- ScaleData(immune.combined)

immune.combined <- RunPCA(immune.combined)

immune.combined <- RunUMAP(immune.combined, dims = 1:20)

immune.combined <- RunTSNE(immune.combined, dims = 1:20)

immune.combined <- FindNeighbors(immune.combined, dims = 1:20)

immune.combined <- FindClusters(immune.combined, resolution = 0.8)

DimPlot(immune.combined, label = T)

DimPlot(immune.combined, label = T, reduction = 'tsne')

# Visualization

p1 <- DimPlot(immune.combined, reduction = "tsne", group.by = "group")

p2 <- DimPlot(immune.combined, reduction = "tsne", label = TRUE, repel = TRUE)

p1 + p2

DefaultAssay(immune.combined) <- "RNA"

save(immune.combined, file = "Milk.combined.RData")

features1 <- c("CCL5","GZMK","CD8A","CD8B","CD3E")

features2 <- c("CD4","CD52","TNFSF10","LTB","BCL11B","LEF1")

features3 <- c("BASP1","S100A9","IL1B","PLAUR")

features4 <- c("CTSB","CD36","LGALS3","LGMN","LIPA","CTSZ","CTSS","CD9","GRN","MMP19","APOE","CCL2","CXCL5","SPP1")

features5 <- c("IGHG1","IGLL1","MS4A1","TNFRSF13C","CD79A","TCF4","CD79B","MEF2C","PHGDH","LMO2","IRF8")

features6 <- c("FTL","FTH1","APOE","SPP1","CTSB","CSTB","GPX1","LIPA","CD9","CTSK")

features7 <- c("HSPA1A","EGR1","ADAMTS1","RHOB","SOX4","FOXQ1","NRARP")

features8 <- c("CCRL2","BATF3","GPR84","NFKBIA","SOD2","IL1RN","G0S2","BCL6","MCEMP1","PTX3")

features9 <- c("CLDN4","WFDC2","KRT7","CLDN3","KRT19","KRT8","EPCAM")

features10 <- c("FCN1","CD14","CYBB","SERPINB2","LYZ","LAMP1","RNASE6","CXCL3","S100A4","S100A12","CXCL2","CCL2","CXCL5")

features11 <- c("CD7","TCF7","KIT","XCL2","KLRD1","IKZF2","CRTAM","PIK3R1","GPR171","CTSW")

features12 <- c("CD1E","PLD4","CD74","AIF1","PKIB","IFI30","CD83","LYZ","CST3","GPR183","CCL17")

features13 <- c("CCL22","CCR7","LY75","PPP1R14A","CD83","CCL17","PKIB","LAMP3")

features14 <- c("LALBA","ELF5","SCD","GLYCAM1","CSN1S2","FASN","MFGE8","KRT19","CSN3","LTF","CSN1S1","CSN2")

features15 <- c("CD8A","CD8B","CCL5","GZMK","CD3E")

features16 <- c("CD3G","CD3E","CTSW","CXCR4","CCL5")

features17 <- c("UBE2C","TOP2A","CENPF","STMN1","TUBB","TUBA1B")

features18 <- c("CCL5","GZMK")

features19 <- c("MKI67","TOP2A","CENPF","UBE2C","STMN1","PCNA","TUBB","TUBA1B")

library(ggplot2)

FeaturePlot(immune.combined, features = features1,reduction = "tsne")

ggsave("Milk0.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features2,reduction = "tsne")

ggsave("Milk1.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features3,reduction = "tsne")

ggsave("Milk2.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features4,reduction = "tsne")

ggsave("Milk3.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features5,reduction = "tsne")

ggsave("Milk4.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features6,reduction = "tsne")

ggsave("Milk5.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features7,reduction = "tsne")

ggsave("Milk6.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features8,reduction = "tsne")

ggsave("Milk7.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features9,reduction = "tsne")

ggsave("Milk8.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features10,reduction = "tsne")

ggsave("Milk9.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features11,reduction = "tsne")

ggsave("Milk10.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features12,reduction = "tsne")

ggsave("Milk11.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features13,reduction = "tsne")

ggsave("Milk12.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features14,reduction = "tsne")

ggsave("Milk13.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features15,reduction = "tsne")

ggsave("Milk14.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features16,reduction = "tsne")

ggsave("Milk15.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features17,reduction = "tsne")

ggsave("Milk16.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features18,reduction = "tsne")

ggsave("Milk17.pdf",width=10,height=10)

FeaturePlot(immune.combined, features = features19,reduction = "tsne")

ggsave("Milk18.pdf",width=10,height=10)

milk.markers <- FindAllMarkers(immune.combined, only.pos = TRUE, min.pct = 0.1, logfc.threshold = 0.25)

#Top10 cluster genes

top20 <- milk.markers %>% group\_by(cluster) %>% top\_n(n = 20, wt = avg\_log2FC)

#VlnPlot(immune.combined, features = c("CD7", "CD96","CD244","XCL2","XCL1","RGS2.1"))

#FeaturePlot(immune.combined, features = c("CD3D", "GNLY", "IFI6"), split.by = "stim", max.cutoff = 3,cols = c("grey", "red"))

#FeaturePlot(immune.combined, features = c("CD3D", "SELL", "CREM", "CD8A", "GNLY", "CD79A", "FCGR3A","CCL2", "PPBP"), min.cutoff = "q9")

write.table(top20,"Milk.marker.txt",col.names = T,row.names = F,sep="\t",quote=F)

#######

#data.raw <- data.frame(GetAssayData(immune.combined))

#data.cluster <- immune.combined@meta.data

DefaultAssay(immune.combined) <- "RNA"

#immune.combined2 <- NormalizeData(immune.combined,scale.factor = 1e6)

log\_data <- GetAssayData(immune.combined)

log2\_data <- expm1(log\_data)

log2\_data <- data.frame(log2\_data)

set.seed(10000)

log2\_data\_out <- log2\_data[,sample(1:16469,2000)]

data.cluster <- immune.combined@meta.data

data.cluster$sample <- gsub("-",".",rownames(data.cluster))

ss <- data.frame(sample=colnames(log2\_data\_out))

ss <- ss %>% left\_join(data.cluster,by="sample")

ss <- ss[,c(1,9)]

ss$seurat\_clusters <- paste("ClusterMilk",ss$seurat\_clusters,sep="")

log2\_data\_out$gene <- rownames(log2\_data\_out)

log2\_data\_out <- log2\_data\_out[,c(2001,1:2000)]

ss2 <- c("gene",ss$seurat\_clusters)

log2\_data\_out <- rbind(ss2,log2\_data\_out)

write.table(log2\_data\_out,"Milk.CPM.100.2000.txt",col.names = F,row.names = F,sep="\t",quote=F)

#######

#data.raw <- data.frame(GetAssayData(immune.combined))

#data.cluster <- immune.combined@meta.data

DefaultAssay(immune.combined) <- "RNA"

#immune.combined2 <- NormalizeData(immune.combined,scale.factor = 1e6)

log\_data <- GetAssayData(immune.combined)

log2\_data <- expm1(log\_data)

log2\_data <- data.frame(log2\_data)

set.seed(10000)

log2\_data\_out <- log2\_data[,sample(1:16469,16469)]

data.cluster <- immune.combined@meta.data

data.cluster$sample <- gsub("-",".",rownames(data.cluster))

ss <- data.frame(sample=colnames(log2\_data\_out))

ss <- ss %>% left\_join(data.cluster,by="sample")

ss <- ss[,c(1,9)]

ss$seurat\_clusters <- paste("ClusterMilk",ss$seurat\_clusters,sep="")

ss$celltype <- ""

ss$celltype[ss$seurat\_clusters %in% c("ClusterMilk0","ClusterMilk14","ClusterMilk15","ClusterMilk17")] <- "CD8Tcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterMilk1")] <- "CD4Tcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterMilk2","ClusterMilk7")] <- "Monocyte"

ss$celltype[ss$seurat\_clusters %in% c("ClusterMilk3","ClusterMilk5","ClusterMilk9")] <- "Macrophage"

ss$celltype[ss$seurat\_clusters %in% c("ClusterMilk4")] <- "Bcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterMilk6")] <- "Endothelialcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterMilk8")] <- "Mammaryepithelialcellluminal"

ss$celltype[ss$seurat\_clusters %in% c("ClusterMilk10")] <- "NKTcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterMilk11","ClusterMilk12")] <- "Dendriticcell"

ss$celltype[ss$seurat\_clusters %in% c("ClusterMilk13")] <- "MammaryepithelialcellAlveolar"

ss$celltype[ss$seurat\_clusters %in% c("ClusterMilk16","ClusterMilk18")] <- "GammaDeltaTcell"

log2\_data\_out$gene <- rownames(log2\_data\_out)

log2\_data\_out <- log2\_data\_out[,c(16470,1:16469)]

ss2 <- c("gene",ss$celltype)

log2\_data\_out <- rbind(ss2,log2\_data\_out)

write.table(log2\_data\_out,"Milk.CPM.100.16469.txt",col.names = F,row.names = F,sep="\t",quote=F)

**LASSO**

setwd("F:/milk cell")

library(dplyr)

myexpr <- read.csv("423sample.csv",header = T,row.names = 1)

myexpr[1:3,1:4]

mysurv <- read.csv("423suv.csv",header = T,row.names = 1)

head(mysurv)

if (all(colnames(myexpr) %in% rownames(mysurv))){

warning("IDs of both files are the same")

} else{

warning("ids of both files are the same")

}

library("glmnet")

library("survival")

cvfit = cv.glmnet(t(myexpr), Surv(mysurv$months,mysurv$status),

#10fold

#nfold=10,

family = "cox"

)

plot(cvfit)

cvfit$lambda.min

cvfit$lambda.1se

fit <- glmnet(t(myexpr), Surv(mysurv$months,mysurv$status),

family = "cox")

plot(fit, label = TRUE)

mycol <- rep(c("#223D6C","#D20A13","#FFD121","#088247","#11AA4D","#58CDD9","#7A142C","#5D90BA","#431A3D","#91612D","#6E568C","#E0367A","#D8D155","#64495D","#7CC767"),2)

xmax <- 3.6

plotCoef\_plus <- function (beta, norm, lambda, df, dev, label = FALSE, legend = FALSE, xvar = c("norm",

"lambda", "dev"), xlab = iname, ylab = "Coefficients", ...)

{

which = nonzeroCoef(beta)

nwhich = length(which)

switch(nwhich + 1, `0` = {

warning("No plot produced since all coefficients zero")

return()

}, `1` = warning("1 or less nonzero coefficients; glmnet plot is not meaningful"))

beta = as.matrix(beta[which, , drop = FALSE])

xvar = match.arg(xvar)

switch(xvar, norm = {

index = if (missing(norm)) apply(abs(beta), 2, sum) else norm

iname = "L1 Norm"

approx.f = 1

}, lambda = {

index = log(lambda)

iname = "Log Lambda"

approx.f = 0

}, dev = {

index = dev

iname = "Fraction Deviance Explained"

approx.f = 1

})

dotlist = list(...)

type = dotlist$type

if (legend){

par(xpd = T, mar = par()$mar + c(0,0,0,6))

}

if (is.null(type))

matplot(index, t(beta), lty = 1, lwd = 2,

xlab = xlab, ylab = ylab,

xlim = c(0, xmax),

col = mycol,

type = "l", cex.lab=1.2, cex.axis=1,

bty="n", ...)

else matplot(index, t(beta), lty = 1, lwd = 2,

xlab = xlab, ylab = ylab,

xlim = c(0, xmax),

col = mycol,

type = "l", cex.lab=1.2, cex.axis=1,

bty="n", ...)

atdf = pretty(index)

prettydf = approx(x = index, y = df, xout = atdf, rule = 2,

method = "constant", f = approx.f)$y

axis(3, at = atdf, labels = prettydf, tcl = NA)

if (label) {

nnz = length(which)

xpos = max(index)

pos = 4

if (xvar == "lambda") {

xpos = min(index)

pos = 2

}

xpos = rep(xpos, nnz)

ypos = beta[, ncol(beta)]

text(xpos, ypos, paste(rownames(myexpr)[which]),

cex = 0.8,

col = mycol,

#col = "black",

pos = pos)

}

if (legend) {

legend("topright",

inset=c(-0.12,0),

legend = rownames(myexpr),

col = mycol,

lwd = 3,

cex = 1,

bty = "n")

}

par(xpd=FALSE)

}

plot.glmnet\_plus <- function (x, xvar = c("norm", "lambda", "dev"), label = FALSE, legend = FALSE,

...)

{

xvar = match.arg(xvar)

plotCoef\_plus(x$beta, lambda = x$lambda, df = x$df, dev = x$dev.ratio,

label = label, legend = legend, xvar = xvar, ...)

}

plot.glmnet\_plus(fit, label = TRUE,

legend = FALSE)

abline(v = cvfit$lambda.min, lty = 3,

lwd = 2,

col = "black")

coef.min = coef(cvfit, s = "lambda.min")

coef.min

active.min = which(coef.min != 0)

geneids <- rownames(myexpr)[active.min]

geneids

index.min = coef.min[active.min]

index.min

combine <- cbind(geneids, index.min)

write.csv(combine,"gene\_index\_mastitis423.csv")

signature <- as.matrix(t(myexpr[geneids,])) %\*% as.matrix(index.min)

summary(signature)

colnames(signature)[1] <- "lasso"

row.names = row.names(myexpr)

write.table(signature,"lasso\_output423.txt",row.names = T, quote = F)

expr\_matrix <- read.csv("68\_expr.csv", header = TRUE, row.names = 1)

coefficients <- read.csv("coefficients.csv", header = TRUE)

coefficients <- as.numeric(coefficients[1, ])

if (length(coefficients) != ncol(expr\_matrix)) {

stop("Error: Number of coefficients does not match number of genes in expression matrix.")

}

scores <- numeric(nrow(expr\_matrix))

for (i in 1:nrow(expr\_matrix)) {

score <- sum(expr\_matrix[i, ] \* coefficients)

scores[i] <- score

}

print(scores)

output\_file <- "scores.csv"

write.csv(scores, file = output\_file, row.names = FALSE)

cat("Scores have been successfully saved to", output\_file, "\n")