Smith PhD Thesis

Steven Smith

May 4, 2017

Table of Contents

# Analysis Script

**The following markdown serves as a record for analysis performed in Smith et al., 2017.**  
**Change the root directory to point to where "Script\_input" and "Scripts" directories reside.**  
This script will automatically create the following directories, if not present already:  
1. Tables- Tables associated with manuscript, in .csv or .txt  
2. Figures - Figures associated with manuscript, in .eps  
3. Script\_output - anything else output that is not a figure or table

A sessionsInfo log file will be included in Script\_output.

Script\_input contains static data used by script (previously generated and formatted for use as input). Also includes a data dictionary for sample metadata.

Some library installation may be required (e.g., install.packages("PACKAGE"))

timestamp()

## ##------ Wed May 31 10:12:36 2017 ------##

# Clear enviornment variables, set root   
rm(list=ls())  
  
## Set the root directory here:  
root\_directory<-"~/Dropbox (IGS)/Jacques\_Steve\_Shared/Thesis/Thesis\_pipeline/AnalysisPipeline/"  
  
## Supress warnings to make knit PDF shorter... but turn these back on since there may be some weird behaviors  
knitr::opts\_chunk$set(warning=FALSE, message=T,size=8)

# Prepare Enviornment

1. Setup environment, variables, etc
2. Define custom functions

## Load libraries   
## Note that some packages mask others. This might be a problem for, e.g., rename.   
  
## Load libraries.  
## Note that some override/mask functions from others. Had to explicitly use the "dplyr" package for all "select" statements  
library(reshape)  
library(ggbiplot)

## Loading required package: ggplot2

## Loading required package: plyr

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:reshape':  
##   
## rename, round\_any

## Loading required package: scales

## Loading required package: grid

library(plyr)  
library(grid)  
library(scales)  
library(gridExtra)  
library(edgeR)

## Loading required package: limma

library(Biobase)

## Loading required package: BiocGenerics

## Loading required package: parallel

##   
## Attaching package: 'BiocGenerics'

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

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

## The following object is masked from 'package:gridExtra':  
##   
## combine

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

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

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

library(RColorBrewer)  
library(psych)

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:scales':  
##   
## alpha, rescale

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library(randomForest)

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:psych':  
##   
## outlier

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

## The following object is masked from 'package:BiocGenerics':  
##   
## combine

## The following object is masked from 'package:gridExtra':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(rfPermute)  
library(tidyverse)

## Loading tidyverse: tibble  
## Loading tidyverse: tidyr  
## Loading tidyverse: readr  
## Loading tidyverse: purrr  
## Loading tidyverse: dplyr

## Conflicts with tidy packages ----------------------------------------------

## %+%(): ggplot2, psych  
## alpha(): ggplot2, psych, scales  
## arrange(): dplyr, plyr  
## col\_factor(): readr, scales  
## combine(): dplyr, randomForest, Biobase, BiocGenerics, gridExtra  
## compact(): purrr, plyr  
## count(): dplyr, plyr  
## discard(): purrr, scales  
## expand(): tidyr, reshape  
## failwith(): dplyr, plyr  
## filter(): dplyr, stats  
## id(): dplyr, plyr  
## lag(): dplyr, stats  
## margin(): ggplot2, randomForest  
## mutate(): dplyr, plyr  
## Position(): ggplot2, BiocGenerics, base  
## rename(): dplyr, plyr, reshape  
## summarise(): dplyr, plyr  
## summarize(): dplyr, plyr

library(squash)  
library(stringr)  
library(plotly)

##   
## Attaching package: 'plotly'

## The following objects are masked from 'package:plyr':  
##   
## arrange, mutate, rename, summarise

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

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

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

## The following object is masked from 'package:graphics':  
##   
## layout

library(gPCA)  
library(nlme)

##   
## Attaching package: 'nlme'

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

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

## The following object is masked from 'package:rfPermute':  
##   
## confusionMatrix

library(gplots)

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:squash':  
##   
## bluered

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

## Source the Random Forest wrapper script  
source(paste0(root\_directory,"Scripts/rfSubjectSpecific.R"))  
  
## Set output directories for tables, figures and data structures.   
if(!dir.exists(file.path(root\_directory, "Figures"))){  
 dir.create(file.path(root\_directory, "Figures"))   
}  
if(!dir.exists(file.path(root\_directory, "Tables"))){  
 dir.create(file.path(root\_directory, "Tables"))   
}  
if(!dir.exists(file.path(root\_directory, "Script\_output"))){  
 dir.create(file.path(root\_directory, "Script\_output"))   
}  
  
R\_script\_output\_directory<-paste0(root\_directory,"Script\_output/")  
R\_script\_input\_directory<-paste0(root\_directory,"Script\_input/") ## Contains Rdata files created outside of this script. These data are static.   
thesis\_tables\_directory<-paste0(root\_directory,"Tables/")  
thesis\_figures\_directory<-paste0(root\_directory,"Figures/")  
  
##Set global variables  
seed\_val<-4543 ## Needed for exact results obtained in manuscript  
pval\_threshold<-0.05 ## Signifgance value threshold  
npermutes<-500 ## Number of permutations to generate emperical null distribution in RF models  
nfolds<-10 #Number of cross-fold validation in customized RF script  
training\_prop<-0.7 ## Proportion of input data for use in training models  
nSpecies<-9 ## Max # of species to plot in Fig 1  
sizes<-1 ## Default sizes for plots  
raThreshold <-0.02 ## Min relative abundance threshold to plot in Fig 1  
margins<-unit(c(-2.5,40,-2.5,5),units="points") ## Default margins for plots  
ph\_normalization\_factor<-3 ## rescales y axis so that plot doesn't bein at 0  
alpha\_rect<-0.7 ## Fig 1 greyed out rectangle opacity  
rect\_fill<-"grey" ## Fig 1 greyed out color  
removed\_samples<-data.frame(Pre\_QC\_ID=NULL,QC\_removal\_stage=NULL)## Keep a list of removed samples, and the stage of removal  
  
## Set global plot theme  
mBio <- theme\_bw() + theme(text = element\_text(family = "Arial", colour = "black",size=12))  
  
## Set standardized color table  
load(file=paste0(R\_script\_input\_directory,"subject\_long\_taxa\_colors.Rdata"))  
color\_scheme\_BCS<- c("L. crispatus"= unname(subject\_long\_taxa\_colors["Lactobacillus\_crispatus"]),  
 "L. jensenii" = unname(subject\_long\_taxa\_colors["Lactobacillus\_jensenii"]),  
 "L. iners" = unname(subject\_long\_taxa\_colors["Lactobacillus\_iners"]),  
 "G. vaginalis" = unname(subject\_long\_taxa\_colors["Gardnerella\_vaginalis"]),  
 "Cell Culture Medium"='blue'  
) #c("L. crispatus"='red1',"G. vaginalis"='lightseagreen',"L. iners"='orange',"Cell Culture Medium"='blue',"L. jensenii"='#8c510a')  
  
cst.colors<-c("I-A"="red1",  
 "I-B"="#990000",  
 "I" = unname(subject\_long\_taxa\_colors["Lactobacillus\_crispatus"]),  
 "II"=unname(subject\_long\_taxa\_colors["Lactobacillus\_gasseri"]),  
 "III-A"="darkorange" ,  
 "III-B"="#cc7a00" ,  
 "III"=unname(subject\_long\_taxa\_colors["Lactobacillus\_iners"]),  
 "IV-A"="lightseagreen",  
 "IV-B"="mediumblue",  
 "V"=unname(subject\_long\_taxa\_colors["Lactobacillus\_jensenii"]),  
 "DUMMY"='grey'  
)  
  
## Table and figure names  
TABLE\_SEQSUMMARY<-"TABLE\_2\_1.csv"  
TABLE\_MIR\_TARGETS<-"TABLE\_2\_2.txt"  
TABLE\_TOPMIRS<-"TABLE\_2\_3.txt"  
TABLE\_QPCR\_TIMECOURSE<-"TABLE\_2\_4.csv"  
TABLE\_EDU\_SCRATCH\_QUANT<-"TABLE\_2\_5.csv"  
TABLE\_CT\_QUANT<-"TABLE\_2\_6.csv"  
  
TABLE\_TRL\_ALIGNSTATS<-"TABLE\_3\_1.csv"  
TABLE\_TRL\_NUMDEGENES<-"TABLE\_3\_2.csv"  
TABLE\_TRL\_SUMMARY\_PATHWAYS<-"TABLE\_3\_3.csv"  
  
TABLE\_COUNTS\_RAW<-"TABLE\_A4.csv"  
TABLE\_MODEL\_INPUT<-"TABLE\_A5.csv"  
TABLE\_SRL\_METADATA<-"TABLE\_A6.csv"  
TABLE\_PROXY\_AMSEL\_INPUT<-"TABLE\_A7.csv"  
TABLE\_RF\_SUMMARY.CV<-"TABLE\_A8.csv"  
TABLE\_PROXY\_AMSEL\_SRL<-"TABLE\_A9.csv"  
TABLE\_RF\_SUMMARY<-"TABLE\_A10.csv"  
  
TABLE\_TRL\_COUNTS\_RAW<-"TABLE\_A11.csv"  
TABLE\_EDGER\_RESULTS<-"TABLE\_A12.csv"  
  
  
FIGURE\_SUBJECT\_PLOTS<-"FIGURE\_2\_1\_"  
FIGURE\_QC<-"FIGURE\_2\_2\_"  
FIGURE\_QC\_RIN\_v\_READS.PROP<-paste0(FIGURE\_QC,"1.eps")  
FIGURE\_QC\_RIN\_v\_READS.ABS<-paste0(FIGURE\_QC,"2.eps")  
FIGURE\_QC\_PCA.PREQC.BYBATCH<-paste0(FIGURE\_QC,"3.eps")  
FIGURE\_QC\_PCA.PREQC.BYSUBJ<-paste0(FIGURE\_QC,"4.eps")  
FIGURE\_QC\_PCA.RMLOW.BYBATCH<-paste0(FIGURE\_QC,"5.eps")  
FIGURE\_QC\_PCA.RMLOW.BYSUBJ<-paste0(FIGURE\_QC,"6.eps")  
FIGURE\_QC\_PCA.NORMAL.BYBATCH<-paste0(FIGURE\_QC,"7.eps")  
FIGURE\_QC\_PCA.NORMAL.BYSUBJ<-paste0(FIGURE\_QC,"8.eps")  
FIGURE\_RF\_IMPORTANCE\_AMSEL<-"FIGURE\_2\_3\_A.eps"  
FIGURE\_RF\_IMPORTANCE\_NUGENT<-"FIGURE\_2\_3\_B.eps"  
FIGURE\_TOPMIRS<-"FIGURE\_2\_3\_C.eps"  
FIGURE\_QPCR\_TIMECOURSE<-"FIGURE\_2\_5.eps"  
FIGURE\_MIR\_TARGETS\_GO<-"FIGURE\_2\_4.eps"  
FIGURE\_DL\_LACTICACID\_QPCR<-"FIGURE\_2\_6.eps"  
FIGURE\_EDU\_QUANT<-"FIGURE\_2\_7\_D.eps"  
FIGURE\_SCRATCH\_QUANT<-"FIGURE\_2\_7\_C.eps"  
FIGURE\_CT\_INFECT\_QUANT<-"Figure\_2\_8\_G.eps"  
FIGURE\_CT\_EDU\_QUANT<-"Figure\_2\_8\_F.eps"  
FIGURE\_nyc\_v\_tsb<-"FIGURE\_2\_9.eps"  
  
FIGURE\_TRL\_RIN\_HIST<-"Figure\_3\_1.eps"  
FIGURE\_COMBINED\_PATHWAYS\_IMMUNE<-"FIGURE\_3\_2"  
FIGURE\_COMBINED\_PATHWAYS\_CYCLE<-"FIGURE\_3\_3"  
FIGURE\_LONGITDUINAL\_GENEEXP.immune<-"FIGURE\_3\_4.eps"  
FIGURE\_LONGITDUINAL\_GENEEXP.cycle<-"FIGURE\_3\_7.eps"  
  
setwd(paste0(R\_script\_output\_directory))  
  
## SRL\_meta and SRL\_counts were prepared in separate script-   
load(file=paste0(R\_script\_input\_directory,"SRL\_counts\_meta.RData"))  
  
## Extract counts data from ExpressionSet  
SRL\_counts\_table<-exprs(SRL\_counts\_meta)  
SRL\_meta\_table<-pData(SRL\_counts\_meta)  
  
##Write SRL Raw Counts to disk  
write.csv(SRL\_counts\_table,file=paste0(thesis\_tables\_directory,TABLE\_COUNTS\_RAW),row.names=T,quote=F)

Custom functions

###########  
## remove\_poorQC\_samples function  
###########  
  
## Adds samples in sample\_list to running removed\_samples data.frame. Tracks the reaon for removal  
remove\_poorQC\_samples<-function(removed\_samples=removed\_samples,  
 sample\_list=c(""),  
 reason=""){  
   
 removed\_samples<-unique(rbind(removed\_samples,data.frame(Pre\_QC\_ID=sample\_list,QC\_removal\_stage=reason)))  
 return(removed\_samples)  
}  
  
###########  
## End remove\_poorQC\_samples function  
###########  
  
  
###########  
## plot\_pca function  
###########  
  
## Plots PCA and calcualtes gPCA-based p value. Outputs a list with PCA plot object and p value (if plotly=F). Otherwise, just a plotly plot object   
  
plot\_pca<-function(ES, ## Expression Set object  
 plot\_title="", ## Title for PCA plot  
 center=TRUE, ## pcrcomp centering?  
 scale=TRUE, ##prcomp scaling?  
 color\_by='SID', ## What to color plots by  
 logt=TRUE, #Whether counts in ES should be log transformed  
 obs.scale = 1, ##prcomp scale factor  
 var.scale = 1, ##prcomp var scaling  
 ellipse = FALSE, ## draw ellipse around groups in PCA plot  
 circle = FALSE, ## or circle  
 var.axes=FALSE, ##option for prcomp  
 ploly=FALSE, ##whether to generate a ploty interactive plot. This option will not return gPCA p value  
 margins=unit(c(0,0,0,0),units = "points"),  
 seed\_val=4543, ## seed needed for gPCA  
 ...){   
   
 ## es = ExpressionSet object   
 ## \*\* Assumes ES counts data is log transformed. Set logt=FALSE to logt data first.   
 ## \*\*Assumes columns named "Batch", "SID" ,"BVGroup" and "NUGENT\_CLASS" exist in pData in ES.   
   
 set.seed(seed\_val) ## Repeatable results in Guided PCA  
 ##calcualte variance of counts  
 variable\_counts<-t(exprs(ES))[,apply(t(exprs(ES)), 2, var, na.rm=TRUE) != 0]  
   
 ## log transform data if needed  
 if(logt){  
 cnts <- log(variable\_counts+1,base = 2)  
 }else{  
 cnts<-variable\_counts  
 }  
   
 ## extract metadata from ES to decorate tree. Assumes these variables are present  
 batch <- as.numeric(pData(ES)[, 'Batch'])   
 sid<-pData(ES)[, 'SID']  
 bvgroup<-pData(ES)[,'BVGroup']  
 bvclass<-pData(ES)[,'NUGENT\_CLASS']  
 ##color mappings for different metadata  
 if(color\_by=="SID"){  
 group<-as.character(sid)  
 group.gpca<-as.numeric(as.factor(group))  
 }else if(color\_by=="Batch"){  
 group<-as.character(batch)  
 group.gpca<-batch  
 }else if (color\_by=="NugentC"){  
 group<-as.character(bvclass)  
 group.gpca<-as.numeric(as.factor(group))  
 }  
 else{  
 group<-as.character(bvgroup)  
 group.gpca<-as.numeric(as.factor(group))  
 }  
   
 ##PCA  
 counts.pca <- prcomp(cnts,center = center, scale. = scale)   
   
 #PCA plot  
 pca.p <- ggbiplot(counts.pca, groups = group,   
 obs.scale = obs.scale, var.scale = var.scale ,   
 ellipse = ellipse,   
 circle = circle,var.axes =var.axes, ...)  
 pca.p <- pca.p +   
 scale\_color\_discrete(name = '')+  
 mBio+  
 theme(plot.margin = margins)  
   
 gPCA.result<-gPCA.batchdetect(dist(cnts),group.gpca)  
   
 ##Optional plotly functionality... returns a list with plot object in an element otherwise (with gPCA p value)  
 if(ploly){ggplotly(pca.p)}else{return(list(pca.p=pca.p,gPCA.result=gPCA.result))}  
}  
  
###########  
## End plot\_pca definition  
###########  
  
###########  
## subset\_ExpressionSet function  
###########  
  
## Function to subset ExpressionSet objects based on vector of sample/rows names to remove. Returns an expression set without filterout samples  
  
subset\_ExpressionSet<-function(expSet, ##expression set  
 filterOut=c(""), ##vector of samples to drop from expSet  
 samples=TRUE){ ## filter samples names or row names  
   
 ## Drop sample (columns) from expSet  
 if(samples){  
 ## Remove count data in filterOut vector  
 counts\_meta<-exprs(expSet)[,!colnames(exprs(expSet)) %in% filterOut]  
 ## Remove metadata in filterOut vector  
 design.subset<-pData(expSet)[!row.names(pData(expSet)) %in% filterOut,]  
 ## Return a re-packaged filtered count and metadata into ExpressionSet  
 }  
 else{## Drop miRNAs (rows) from expSet  
 counts\_meta<-exprs(expSet)[!row.names(exprs(expSet)) %in% filterOut,]  
 design.subset<-pData(expSet)[,!colnames(pData(expSet)) %in% filterOut]  
   
 }  
 return(ExpressionSet(assayData = as.matrix(counts\_meta),  
 phenoData = AnnotatedDataFrame(design.subset)))   
}  
  
###########  
## End subset\_ExpressionSet   
###########  
  
###########  
## plot\_accuracy   
###########  
  
## Generates predicted vs acutal tables and plots the predicted vs actual values on plot  
  
plot\_accuracy<-function(rfp, ## Random Forest object  
 testing\_fullset, ## vector of sample names that were held out of training/used for testing  
 index\_of\_response=match("NUGENT\_SCORE",names(testing\_fullset)), ## index corresponding to response variable in input data  
 index\_of\_sid=match("SID",names(testing\_fullset)),## index corresponding to subject ID in input data (if subj\_spec=T)  
 subj\_spec=TRUE, ## whether RF was run with 'subject\_spec' option  
 nfold=10) # number of k folds in RF if subj\_spec=T  
{   
   
   
 accuracy\_table<-data.frame(fold=0,predicted=0,actual=0) ## hold values for actual, predicted values  
   
 ## Subject specific rfp have a model for each cross fold, so need to loop through and aggregate each fold  
 if(subj\_spec){  
   
 for(m in 1:nfold){  
 #m<-1  
 ## Compare predicted to hold out set  
 p1<-predict(rfp$mdl[[m]],   
 testing\_fullset[,-c(index\_of\_response,index\_of\_sid)], type='response')  
   
 accuracy\_table<-rbind(accuracy\_table,data.frame(fold=m,predicted=p1,actual=testing\_fullset[,index\_of\_response]))  
 }  
 }else{  
 p1<-predict(rfp,testing\_fullset[,-c(index\_of\_response,index\_of\_sid)], type='response')  
 accuracy\_table<-data.frame(fold="NA",predicted=p1,actual=testing\_fullset[,index\_of\_response])  
 }  
   
 accuracy\_table<-filter(accuracy\_table,!fold==0) ## drop the initialization row  
   
 plot\_a<-ggplot()+geom\_point(data=accuracy\_table,aes(y=predicted,x=actual,col=as.factor(fold)))+  
 ggtitle("Predicted vs Actual Values from plot\_accuracy")+  
 mBio+  
 scale\_x\_continuous(limits=c(0,10),breaks = 1:10)+  
 scale\_y\_continuous(limits=c(0,10),breaks = 1:10)  
 print(plot\_a)  
   
 return(accuracy\_table)  
   
}  
  
###########  
## End plot\_accuracy definition  
###########  
  
###########  
## run\_randomForest definition  
###########  
  
## Wrapper to run Random Forest model building. Can run RandomForest, rfPermute or rfSubjectSpecific. Takes care of subsetting data for training/testing and outputting accuracy/error/etc etstimates. Saves/loads previous models.   
#Note if load\_prev=T, most of the parameters are ignored  
run\_randomForest<-function(predictors\_response\_table,  
 response\_variable\_name,  
 nfold=10,  
 nreps=105,  
 permute=TRUE,  
 save\_model=FALSE,  
 load\_prev\_model=TRUE,  
 file\_n="rf\_model",  
 verbose=TRUE,  
 pval\_thres=pval\_threshold,  
 subj\_spec=TRUE,  
 importance\_thres=10,  
 training\_prop=0.7,  
 seed=seed\_val,  
 R\_script\_output\_directory,  
 ...)  
{  
   
 # predictors\_response\_table - data frame of predictors + response variable. If it contains a column called 'SID', sets subj\_spec=TRUE  
 # response\_variable\_name - column name as string of response variable (as found in predictors\_response\_table)  
 # nfold - number of k-fold validations to run  
 # nreps - number of permutations to run to compute null distribution. Ignored if permute=FALSE  
 # permute - whether or not to run rfPermute (generate null distribution permutation p-values for each feature)  
 # save\_model - save model as Rdata to outout directory? Uses 'file\_n' as file name  
 # load\_prev\_model - load model outout directory? Uses 'file\_n' as file name  
 # file\_n - file name to use when reading or writing a model to disk  
 # verbose - detailed output of model results, etc  
 # pval\_thres -p value threshold to call signifigant   
 # subj\_spec - whether or not to run rfSubjectSpecific.R. Looks for a column called 'SID' in predictors\_response\_table  
 # importance\_thres -  
 #training\_prop - proportion of input data to use as training. remaining is used as hold out set for testing  
 # seed -seed value to produce repeatable results  
 # R\_script\_output\_directory - root output directory for read/write model files  
   
 nrep<-nreps  
 set.seed(seed)  
 ## First major control point: loading from previously saved model, or generating a new trained model?   
   
 #### ///////  
 # Start New Model  
 #### //////  
   
 if(!load\_prev\_model){  
   
 #Find the column index corresponding to response & subject ID  
 index\_of\_response<-match(response\_variable\_name,names(predictors\_response\_table))  
 index\_of\_sid<-match("SID",names(predictors\_response\_table))  
 if(is.na(index\_of\_sid) & subj\_spec){  
 print("SID could not be found. Setting to non-subject specific")  
 subj\_spec<-FALSE  
 index\_of\_sid<-0  
 }else if (is.na(index\_of\_sid)){  
 index\_of\_sid<-0  
 } else if(index\_of\_sid>0 & !subj\_spec){  
 print("Found a SID column. Setting to subject-spefic. ")  
 subj\_spec<-TRUE  
 }  
   
 #Partition input data into training and testing  
   
 inTrain<-createDataPartition(y=predictors\_response\_table[,index\_of\_response],p =training\_prop,list = F)  
 if(is.character(predictors\_response\_table[,index\_of\_response])){  
 predictors\_response\_table[,index\_of\_response]<-as.factor(predictors\_response\_table[,index\_of\_response])  
 }  
   
 ##Subjet training and testing data  
 training\_fullset<-predictors\_response\_table[inTrain,]  
 testing\_fullset<-predictors\_response\_table[-inTrain,]  
 table(predictors\_response\_table[,index\_of\_response])  
 table(training\_fullset[,index\_of\_response])  
 table(testing\_fullset[,index\_of\_response])  
 response<-training\_fullset[,index\_of\_response]  
   
 # Determine whether to run classification or regression depending on response variable type  
 if(is.numeric(predictors\_response\_table[,index\_of\_response])){# regression  
 rf\_type<-"regression"  
 }else{  
 rf\_type<-"classification"  
 }  
   
 #Determine whether to run rfSubjectSpecific.R  
 if(subj\_spec){  
 print(paste0("Starting subject-specific rfSubjectSpecific with permute set to : ",permute))  
 rfp<-rfSubjectSpecific(training\_fullset[,-c(index\_of\_response,index\_of\_sid)],response,subjID = as.character(training\_fullset[,index\_of\_sid],nrep=nrep),nfolds = nfold,verbose=verbose,nrep = nreps,permute=permute) ## This will be sourced at the setup section. It is an external script.  
 }else{  
 print("Starting non subject-specific rfPermute")  
 rfp<-rfPermute(training\_fullset[,-c(index\_of\_response)],response,nrep = nreps,...)  
 }  
   
 ## The accuracy, etc output varies depending on whether which combination of RF were run  
   
 ## /////  
 ## Regression + Subject-Specific  
 ## ////  
   
 if(rf\_type=="regression" & subj\_spec){ ## If it's a regression model  
 accuracy\_table<-plot\_accuracy(rfp,testing\_fullset = testing\_fullset,index\_of\_response = index\_of\_response, index\_of\_sid = index\_of\_sid,subj\_spec = subj\_spec,nfold = nfold) ## see above for this function  
   
 ## /////  
 ## Classification + Non Subject-Specific  
 ## ////  
   
 }else if (rf\_type=="classification" & !subj\_spec){  
 p1<-predict(rfp, testing\_fullset[,-c(index\_of\_response,index\_of\_sid)], subj\_spec = subj\_spec,type='response')  
 (accuracy\_table<-table(Var1=p1,Var2=testing\_fullset[,index\_of\_response]))  
   
 ## /////  
 ## Classification + Subject-Specific  
 ## ////  
   
 }else if (rf\_type=="classification" & subj\_spec){  
 p1<-predict(rfp$mdl, testing\_fullset[,-c(index\_of\_response,index\_of\_sid)], subj\_spec = subj\_spec,type='response')  
 (accuracy\_table<-lapply(p1, function(x) table(x,testing\_fullset[,c(index\_of\_response)])))  
   
 ## /////  
 ## All else (Regression + Non Subject-Specific)  
 ## ////  
   
 }else{  
 accuracy\_table<-NULL  
 }  
   
 ## Write model to file + accuracy table and training/testing info  
   
 if(save\_model){  
 save(rfp,file=paste0(R\_script\_output\_directory,file\_n,".RData"))  
 save(accuracy\_table,file=paste0(R\_script\_output\_directory,file\_n,"\_accuracyTable.RData"))  
 training\_testing<-list(training\_fullset=training\_fullset,testing\_fullset=testing\_fullset)  
 save(training\_testing,file=paste0(R\_script\_output\_directory,file\_n,"\_training\_testing.RData"))  
 }  
   
 #### ///////  
 # Load Previous Model  
 #### //////  
 }else{  
   
 rfp.pointer<-load(file=paste0(R\_script\_output\_directory,file\_n,".RData"))  
 rfp<-get(rfp.pointer)  
   
 accuracy\_table.pointer<-load(file=paste0(R\_script\_output\_directory,file\_n,"\_accuracyTable.RData"))  
 accuracy\_table<-get(accuracy\_table.pointer)  
   
 training\_testing.pointer<-load(file=paste0(R\_script\_output\_directory,file\_n,"\_training\_testing.RData"))  
 training\_testing<-get(training\_testing.pointer)  
 training\_fullset<-training\_testing$training\_fullset  
 testing\_fullset<-training\_testing$testing\_fullset  
 }  
 if(verbose){  
 print(rfp)  
 print(rfp$mdl)  
 }  
   
 if(permute & subj\_spec){  
 importance<-data.frame(rfp$importance) ##only happens if rfSubjectSpecific is run with permute  
 importance.pval<-dplyr::select(importance,ends\_with("pval"))  
 }else{  
 if(!subj\_spec){  
 importance<-data.frame(rp.importance(rfp))  
 }else{  
 importance<-data.frame(rfp$imp)  
 }  
 }  
 return(list(rfp=rfp, ##model  
 accuracy\_table=accuracy\_table,  
 importance=importance,  
 training\_ids=row.names(training\_fullset),  
 testing\_ids=row.names(testing\_fullset)))   
}  
  
###########  
## END run\_randomForest function  
###########  
  
  
###########  
## plot\_RIN  
###########  
  
## Is there an effect due to RIN and number of reads miRNAs?  
### # Wrapper for plot of reads miRNAs vs RIN  
  
plot\_RIN\_meta<-function(SRL\_meta\_table, ## Table containing SRL metadata  
 y\_series="reads\_surviving.percent", ## y axis column name to be plotted  
 col\_by="Batch", ## color by  
 vjust = 0, ## vertical justification  
 nudge\_y = 0.05, ## nudge y by...  
 angle = 0, ## y axis angle  
 hjust = 0, ## hortizontal adjustment  
 nudge\_x = 0.05, ## nudge x by..  
 check\_overlap = FALSE, ## check for points overlap- try to minimize  
 y\_series\_label=y\_series # y label on plot  
){  
 ## Plot for RIN vs a measure of read mapping  
 p.RIN<-ggplot(SRL\_meta\_table,aes(x=RIN,y=SRL\_meta\_table[,y\_series],col=as.factor(SRL\_meta\_table[,col\_by]),label=SRL\_meta\_table$Pre\_QC\_ID),label = Pre\_QC\_ID)+  
 geom\_point()+  
 ylab(y\_series\_label)+  
 xlab("RINe")+  
 # ggtitle(paste(y\_series," vs RIN"))+  
 geom\_text(check\_overlap = check\_overlap,vjust = vjust, nudge\_y = nudge\_y,angle = angle,hjust = hjust, nudge\_x = nudge\_x)+  
 mBio+  
 guides(col=guide\_legend(title=col\_by))  
 return(p.RIN)  
}  
  
###########  
## end plot\_RIN  
###########  
  
###########  
## mapping\_stats  
###########  
  
## Generate simple mapping stats given an Expression Set pData object  
mapping\_stats<-function(column=pData(counts\_meta.qc)$number\_reads\_mirs){  
 return(data.frame(min=min(column,na.rm = T),median=median(column,na.rm = T),max=max(column,na.rm = T)))  
}  
  
###########  
## end mapping\_stats  
###########  
  
  
###########  
## plot\_importance  
###########  
  
## Plot RF Importance variables. Output is a plot  
  
plot\_importance<-function(importance\_df=Nugent\_RF$importance, ## Data frame holding importance results  
 ntopfeats=length(Nugent\_RF$top\_features$top\_features.all), ## number of features to plot  
 nfeats=25,#max # of features to plot  
 rankBy="IncMSE", ## Importance variable to rank by  
 model\_name="",  
 size\_font=12,  
 size\_points=5  
){  
 if(nfeats<ntopfeats){  
 stop("Number of top features less than total # of features plotted")  
 }  
 names(importance\_df)<-gsub(pattern = "X\\.",replacement = "",x = names(importance\_df))  
 importance\_df<-importance\_df[order(importance\_df[,rankBy],decreasing = T),]  
 importance\_df.cut<-importance\_df[1:nfeats,]  
 importance\_df.cut$features<-factor(row.names(importance\_df.cut),levels = rev(row.names(importance\_df.cut)),ordered = T)  
   
 importance\_df.cut.tmp<-separate(melt(importance\_df.cut,id.vars = c("features")),col = variable,into=c("metric","SUFFIX"),sep = "\\.",fill = "right")  
   
 importance\_df.cut.tmp[is.na(importance\_df.cut.tmp$SUFFIX),"SUFFIX"]<-"metric\_val"  
 importance\_df.cut<-spread(data=importance\_df.cut.tmp,key = SUFFIX,value = value)  
 cutoff<-nfeats+0.5-ntopfeats   
   
 ggplot(importance\_df.cut)+  
 geom\_point(aes(y=features,x=metric\_val,col=-log(pval,10),pch=metric),size=size\_points)+   
 xlab("Importance Metric")+  
 ylab("Predictor")+  
 scale\_colour\_gradient(high="red", low="blue",guide = guide\_legend(title = "-Log(p-value)"))+  
 mBio+  
 theme(text=element\_text(size = size\_font),panel.border = element\_rect(size=1))+  
 ggtitle(paste(model\_name," RF Importance Plot for the First",nfeats,"Ranked Features (by",rankBy,")"))  
   
}  
  
###########  
## end plot\_importance  
###########  
  
###########  
## t.test2  
###########  
  
## A t test using sumamry statistics ( and not the entire sample dataset as with t.test)  
## Returns p value, difference of means, standard error and t statistc  
t.test2 <- function(m1, ## mean of sample set 1  
 m2, ## mean of sample set 2  
 s1, ## standard dev of sample set 1  
 s2, ## standard dev of sample set 2  
 n1, ## number of samples in sample set 1  
 n2, ## number of samples in sample set 1  
 m0=0, ## the null for hypothesis to test (mean value)=  
 equal.variance=FALSE) #whether to assume equal variance between sample sets  
   
{  
 if( equal.variance==FALSE )   
 {  
 ## "normalize" standard deviations if unequal variance to compute standard error  
 se <- sqrt( (s1^2/n1) + (s2^2/n2) )  
 # welch-satterthwaite df  
 df <- ( (s1^2/n1 + s2^2/n2)^2 )/( (s1^2/n1)^2/(n1-1) + (s2^2/n2)^2/(n2-1) )  
 } else  
 {  
 # pooled standard deviation, scaled by the sample sizes  
 se <- sqrt( (1/n1 + 1/n2) \* ((n1-1)\*s1^2 + (n2-1)\*s2^2)/(n1+n2-2) )  
   
 df <- n1+n2-2  
 }   
 t <- (m1-m2-m0)/se## calcualte t statistic  
 dat <- c(m1-m2, se, t, 2\*pt(-abs(t),df)) ## calculate p value based on Student's t distribution  
 names(dat) <- c("Difference of means", "Std Error", "t", "p-value")  
 return(dat)   
}  
  
###########  
## end t.test2  
###########  
  
  
###########  
## apply\_ttest  
###########  
  
apply\_ttest<-function(sigtest\_results=proliferation\_sigtest,pval\_threshold=pval\_threshold,summary\_stats,test\_function="t.test",value="value",obs=""){  
   
 ## Loop through each test comparison and apply stat test. Store in dataframe  
 for(sigtest in 1:nrow(sigtest\_results)){  
 print(sigtest)  
 #sigtest<-1  
 x1<-as.character(sigtest\_results[sigtest,"xref"])   
 y1<-as.character(sigtest\_results[sigtest,"reference"])  
   
 ## Apply a t test to x1 vs y1  
 if(test\_function=="t.test"){  
 tes<-t.test(x=filter(summary\_stats,BCS==x1 )$percent\_cells,y=filter(summary\_stats,BCS==y1)$percent\_cells,alternative = "two.sided")   
 delta\_mean<-tes$estimate[2]-tes$estimate[1]  
 pval<-tes$p.value  
 }else if(test\_function=="t.test2"){  
 obs<-as.character(sigtest\_results[sigtest,"Observation"])  
 m1<-filter(summary\_stats,Observation==obs & Treatment==x1 )$grand\_mean  
 m2<-filter(summary\_stats,Observation==obs & Treatment==y1 )$grand\_mean  
 s1<-filter(summary\_stats,Observation==obs & Treatment==x1 )$grand\_sd  
 s2<-filter(summary\_stats,Observation==obs & Treatment==y1 )$grand\_sd  
 n1<-filter(summary\_stats,Observation==obs & Treatment==x1 )$n  
 n2<-filter(summary\_stats,Observation==obs & Treatment==y1 )$n  
 print(obs)  
 tes<-t.test2(m1 = m1,s1 = s1, m2=m2,s2=s2,n1=n1,n2=n2) #less  
 print(tes)  
 delta\_mean<-tes["Difference of means"]  
 pval<-tes["p-value"]  
   
   
 }else{  
 print(paste0(test\_function," not found"))  
 }  
   
 ## Store p value and difference of means  
 sigtest\_results[sigtest,"pval"]<-pval  
 sigtest\_results[sigtest,"mean\_diff"]<-delta\_mean  
 }  
   
 ## Denote signifigant tests with "\*"  
 sigtest\_results[sigtest\_results$pval<=pval\_threshold,"sig"]<-"\*"  
 return(sigtest\_results)  
}  
  
###########  
## end apply ttest  
###########  
  
###########  
## setup\_sigtest  
###########  
#pval\_threshold = pval\_threshold;raw\_data = proliferation.m;test\_function = "t.test";Experiment = "Scratch"  
  
setup\_sigtest<-function(pval\_threshold = pval\_threshold,raw\_data = raw\_data,test\_function = "t.test",Experiment="Scratch"){  
 ## Set up a significance test df to store inference testing data  
 if(Experiment=="Scratch"){  
 sigtest<-data.frame(xref=c(rep("L. crispatus",times=4),rep("L. jensenii",times=3),rep("L. iners",times=2),"G. vaginalis"),reference=c("L. jensenii","L. iners","G. vaginalis","Cell Culture Medium","L. iners","G. vaginalis","Cell Culture Medium","G. vaginalis","Cell Culture Medium","Cell Culture Medium"))  
   
 ## Factor labels  
 sigtest$xref<-factor(sigtest$xref,levels =c("L. crispatus","L. jensenii","L. iners","G. vaginalis","Cell Culture Medium"),ordered = T)  
 sigtest$reference<-factor(sigtest$reference,levels = c("L. jensenii","L. iners","G. vaginalis","Cell Culture Medium"),ordered = T)  
   
 sigtest<-apply\_ttest(sigtest\_results = sigtest,pval\_threshold = pval\_threshold,summary\_stats = raw\_data,test\_function = "t.test")  
 ## Summarize scratch assay data with mean and sd  
 summary\_stats<-ddply(dplyr::select(raw\_data,c(BCS,percent\_cells)),c("BCS"),summarise,mean=mean(percent\_cells),sd=sd(percent\_cells))  
   
 }else if(Experiment=="Infection"){  
   
 sigtest<-data.frame(Observation=rep(c("Proliferation","Infectivity"),each=3),xref=c(rep("CAS 546102-60-7",times=2),rep("Fascaplysin",times=1)),reference=c("Fascaplysin","Cell Culture Medium","Cell Culture Medium"))  
   
 sigtest$xref<-factor(sigtest$xref,ordered = T,levels = c("CAS 546102-60-7","Fascaplysin"))  
 sigtest$reference<-factor(sigtest$reference,ordered = T,levels = c("Fascaplysin","Cell Culture Medium"))  
   
   
 raw\_data.spread<-spread(raw\_data,key = Observation,value = percent\_cells)  
 plot(raw\_data.spread$Infectivity,raw\_data.spread$Proliferation,col=raw\_data.spread$Treatment)  
   
 summary\_stats.pre<-ddply(raw\_data,c("Observation","Treatment","Coverslip"),summarise,mean=mean(percent\_cells,na.rm = T),sd=sd(percent\_cells,na.rm = T))  
   
 summary\_stats<-ddply(summary\_stats.pre,c("Treatment","Observation"),summarise,grand\_mean=mean(mean,na.rm = T))  
   
 for(obs in c("Infectivity","Proliferation")){  
 for(treat in unique(summary\_stats$Treatment)){  
 #treat<-"Control"  
 gm<-summary\_stats[summary\_stats$Observation==obs & summary\_stats$Treatment==treat,"grand\_mean"]  
 means<-summary\_stats.pre[summary\_stats.pre$Treatment==treat & summary\_stats.pre$Observation==obs,"mean"]  
 n<- length(means[!is.na(means)])  
 summary\_stats[summary\_stats$Treatment==treat & summary\_stats$Observation==obs,"grand\_sd"]<-sqrt(sum((means-gm)^2,na.rm = T)/(n-1))  
 summary\_stats[summary\_stats$Observation==obs & summary\_stats$Treatment==treat,"n"]<-n  
 }  
 }  
   
 sigtest<-apply\_ttest(sigtest\_results = sigtest,pval\_threshold = pval\_threshold,summary\_stats = summary\_stats,test\_function = "t.test2")  
   
   
 }else{  
 print("Experiment must be Infection or Scratch")  
 }  
   
   
 statbars<-data.frame(xref=sigtest$xref,reference=sigtest$reference,  
 x=as.numeric(sigtest$xref)+.05,  
 xend=(as.numeric(sigtest$reference)+1)-.05)  
 if(Experiment=="Scratch"){  
   
 statbars[statbars$reference=="Cell Culture Medium","y"]<-rep(seq(100,91,-3),times=1)  
 statbars[statbars$reference=="G. vaginalis","y"]<-rep(seq(91,by=-3,length.out = 3),times=1)  
 statbars[statbars$reference=="L. iners","y"]<-rep(c(85,94),times=1)  
 statbars[statbars$reference=="L. jensenii","y"]<-rep(94,times=1)  
 statbars[statbars$reference=="Cell Culture Medium" & statbars$xref=="L. iners","y"]<-85  
 statbars[statbars$reference=="G. vaginalis" & statbars$xref=="L. iners","y"]<-94  
   
 }else if(Experiment=="Infection"){  
   
 statbars[statbars$reference=="Cell Culture Medium","y"]<-rep(seq(100,97,-3),times=2)  
 statbars[statbars$reference=="Fascaplysin","y"]<-rep(c(91,94),times=1)  
 statbars[statbars$reference=="CAS 546102-60-7","y"]<-rep(90,times=2)  
 statbars[statbars$xref=="CAS 546102-60-7" &statbars$reference=="Fascaplysin" ,"y"] <-97  
   
 }else{  
 print("Experiment must be Infection or Scratch")  
 }  
   
   
 statbars$yend<-statbars$y  
 statbars2<-data.frame(x=c(statbars$x,statbars$xend),  
 y=rep(statbars$y,times=2),  
 yend=rep(statbars$y-1.5,times=1))  
 statbars2$xend<-statbars2$x  
   
 sigtest<-merge(statbars,sigtest,by = c("xref","reference"))  
   
 sigtest$midpoints<-((sigtest$xend-sigtest$x)/2)+sigtest$x  
 sigtest$y.sig<-sigtest$y+1  
 sigtest$sig[is.na(sigtest$sig)]<-"N.S."  
   
 return(list(sigtest=sigtest,statbars=statbars,statbars2=statbars2,summary\_stats=summary\_stats))  
}  
  
###########  
## end setup\_sigtest  
###########  
  
###########  
## plot\_replicates  
###########  
  
  
plot\_replicates<-function(eset,BCS.selection=c(""),ExposureTime.selection=c(""),logt=F,rmlow=F,lowcnt=10){  
   
 if(logt){  
 cnts<-log(exprs(eset)+1,base = 2)  
 }else{  
 cnts<-exprs(eset)  
 }  
 if(rmlow){  
   
 cnts<-cnts[rowSums(cnts>lowcnt)==ncol(cnts),]  
 }  
   
 pairs.panels(cnts[, pData(eset)$BCS %in% c(BCS.selection) & pData(eset)$ExposureTime %in% c(ExposureTime.selection)])  
}  
  
  
###########  
## end plot\_replicates  
###########  
  
###########  
## subset\_ExpressionSet  
###########

# Summary Statistics & Quality Control

## Alignment Stats

Summarize the SmallRNA seq alignments

## Stats on # reads hg (% surviving), # reads miRNA (% survibing, hg)  
SRL\_meta\_table\_summary\_table<-data.frame(matrix(0,nrow=3,ncol=3));names(SRL\_meta\_table\_summary\_table)<-c("min","median","max");row.names(SRL\_meta\_table\_summary\_table)<-c("trimmed","hg19","miRNome")  
  
SRL\_meta\_table\_summary\_table["trimmed",]<-summarise(SRL\_meta\_table,min=min(reads\_surviving),median=median(reads\_surviving),max=max(reads\_surviving))  
SRL\_meta\_table\_summary\_table["hg19",]<-summarise(SRL\_meta\_table,min=min(hg19.mapped),median=median(hg19.mapped),max=max(hg19.mapped))  
SRL\_meta\_table\_summary\_table["miRNome",]<-summarise(SRL\_meta\_table,min=min(number\_reads\_mirs),median=median(number\_reads\_mirs),max=max(number\_reads\_mirs))  
print(SRL\_meta\_table\_summary\_table)

## min median max  
## trimmed 3225 27087178 107477660  
## hg19 607 4499861 17195821  
## miRNome 159 1102119 4642910

## Quality Control (QC) small RNA-seq count data

print(paste("Number of samples in inital counts table:",ncol(SRL\_counts\_table)))

## [1] "Number of samples in inital counts table: 113"

print(paste("Number of miRNAs in inital counts table:",nrow(SRL\_counts\_table)))

## [1] "Number of miRNAs in inital counts table: 1869"

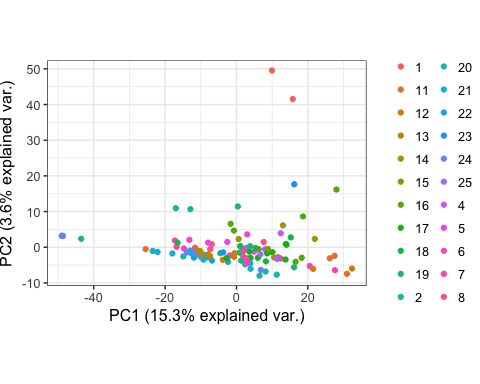
## Determine number of miRNAs without any counts across all samples   
missing<-rowSums(SRL\_counts\_table)==0 #rowSums(is.na(SRL\_counts\_table))==ncol(SRL\_counts\_table) |   
print(paste("Number of miRNAs without any counts across all",ncol(SRL\_counts\_table),"samples:",sum(missing)))

## [1] "Number of miRNAs without any counts across all 113 samples: 323"

print(paste0("which leaves ",100\*(1-sum(missing)/nrow(SRL\_counts\_table))," percent of all annotated miRNAs with at least 1 read"))

## [1] "which leaves 82.7180310326378 percent of all annotated miRNAs with at least 1 read"

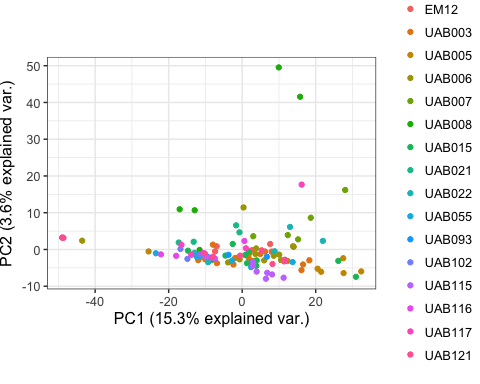
#print("...and the names of the miRNAs without any reads across all samples: ")  
#names(missing)[missing]  
  
plot\_QC.preQC.batch<-plot\_pca(SRL\_counts\_meta,"Log miRNA Raw Count Distance, PRE QC",color\_by = "Batch",ploly = F,seed\_val = seed\_val)##FIGURE  
plot(plot\_QC.preQC.batch$pca.p)



plot\_QC.preQC.batch$gPCA.result$p.val

## [1] 0.994

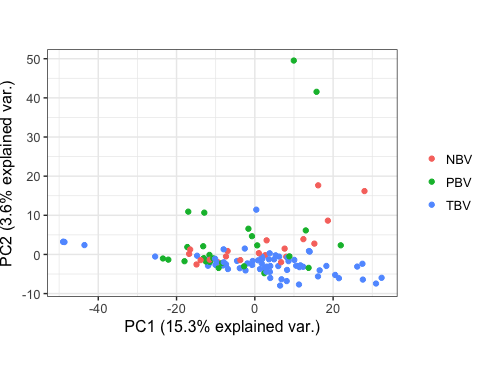
plot\_QC.preQC.sid<-plot\_pca(SRL\_counts\_meta,"Log miRNA Raw Count Distance, PRE QC",color\_by = "SID",ploly = F,seed\_val = seed\_val)##FIGURE  
plot(plot\_QC.preQC.sid$pca.p)



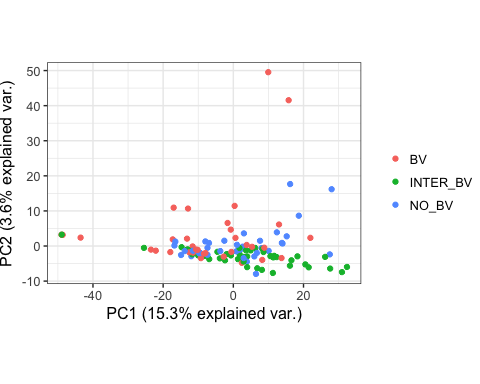
plot\_QC.preQC.sid$gPCA.result$p.val

## [1] 0.288

plot\_pca(SRL\_counts\_meta,"Log miRNA Raw Count Distance, PRE QC",color\_by = "BVGroup",seed\_val = seed\_val)$pca.p##FIGURE



plot\_pca(SRL\_counts\_meta,"Log miRNA Raw Count Distance, PRE QC",color\_by = "NugentC",seed\_val = seed\_val)$pca.p##FIGURE



## Get a \*ROUGH\* idea of miRNA coverage by dividing counts by # of annotated miRNAs in genome   
(number\_mirs\_genome<-nrow(SRL\_counts\_meta)) ## number of annotated miRNAs

## Features   
## 1869

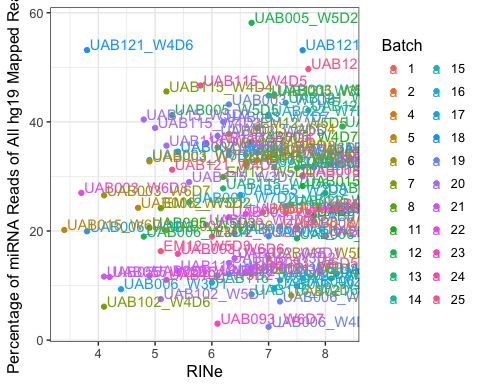
#SRL\_counts\_table<-SRL\_counts\_table[!missing,] ## non missing counts only   
SRL\_counts\_meta<-SRL\_counts\_meta[!missing]  
sum(missing)

## [1] 323

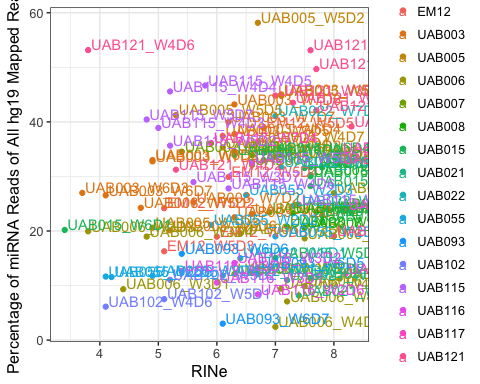
## Number of miRNAs completley removed due to non-detection  
nrow(SRL\_counts\_meta)

## Features   
## 1546

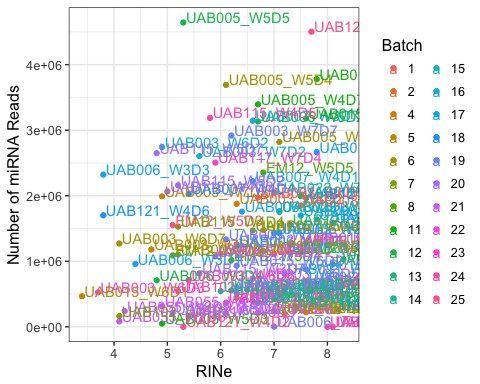
(plot\_QC.RIN.percent.batch<-plot\_RIN\_meta(SRL\_meta\_table,"number\_reads\_mirs.percent\_hgmapped",col\_by="Batch",y\_series\_label = "Percentage of miRNA Reads of All hg19 Mapped Reads"))##FIGURE



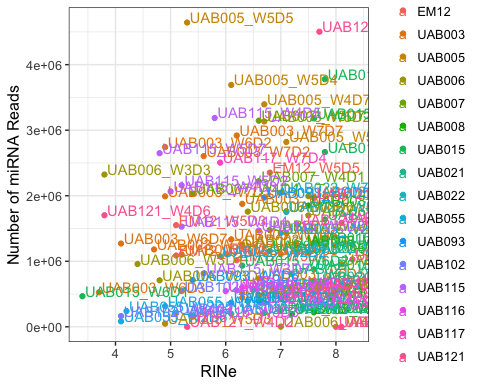
(plot\_QC.RIN.percent.sid<-plot\_RIN\_meta(SRL\_meta\_table,"number\_reads\_mirs.percent\_hgmapped",col\_by="SID",y\_series\_label = "Percentage of miRNA Reads of All hg19 Mapped Reads"))



(plot\_QC.RIN.number.batch<-plot\_RIN\_meta(SRL\_meta\_table,"number\_reads\_mirs",col\_by="Batch",y\_series\_label = "Number of miRNA Reads"))##FIGURE



(plot\_QC.RIN.number.sid<-plot\_RIN\_meta(SRL\_meta\_table,"number\_reads\_mirs",col\_by="SID",y\_series\_label = "Number of miRNA Reads"))



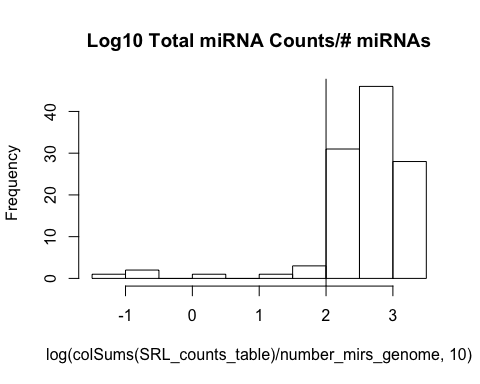
## Range of miRNA coverage based on simple reads/#annotated miRNAs  
print(summary(colSums(SRL\_counts\_table))/number\_mirs\_genome)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0851 258.0524 589.6201 684.3232 989.8341 2484.2162

## Avergae coverage  
data.frame(Pre\_QC\_ID=SRL\_meta\_table$Pre\_QC\_ID,avg\_coverage=SRL\_meta\_table$number\_reads\_mirs/number\_mirs\_genome)

## Pre\_QC\_ID avg\_coverage  
## 1 EM12\_W5D2 5.821814e+02  
## 2 EM12\_W5D3 8.298529e+02  
## 3 EM12\_W5D4 5.437319e+02  
## 4 EM12\_W5D5 1.258753e+03  
## 5 EM12\_W5D6 5.969540e+02  
## 6 UAB003\_W5D6 9.440787e+02  
## 7 UAB003\_W6D1 6.720284e+02  
## 8 UAB003\_W6D2 1.467211e+03  
## 9 UAB003\_W6D3 2.817180e+02  
## 10 UAB003\_W6D4 2.377742e+02  
## 11 UAB003\_W6D5 3.938844e+02  
## 12 UAB003\_W6D6 5.896838e+02  
## 13 UAB003\_W6D7 6.799176e+02  
## 14 UAB003\_W7D1 6.317940e+02  
## 15 UAB003\_W7D2 1.393483e+03  
## 16 UAB003\_W7D3 1.065232e+03  
## 17 UAB003\_W7D5 7.140310e+02  
## 18 UAB003\_W7D6 2.752729e+02  
## 19 UAB003\_W7D7 1.561850e+03  
## 20 UAB003\_W8D1 5.939187e+02  
## 21 UAB003\_W8D2 1.003773e+03  
## 22 UAB005\_W4D7 1.816241e+03  
## 23 UAB005\_W5D1 1.508920e+03  
## 24 UAB005\_W5D2 1.676461e+03  
## 25 UAB005\_W5D3 2.577314e+01  
## 26 UAB005\_W5D4 1.974515e+03  
## 27 UAB005\_W5D5 2.484168e+03  
## 28 UAB006\_W3D1 5.126613e+02  
## 29 UAB006\_W3D3 1.242925e+03  
## 30 UAB006\_W3D4 3.796479e+02  
## 31 UAB006\_W3D5 6.575725e+02  
## 32 UAB006\_W3D6 9.406057e+02  
## 33 UAB006\_W3D7 3.035163e+02  
## 34 UAB006\_W4D2 9.091423e+02  
## 35 UAB006\_W4D3 1.067289e+03  
## 36 UAB006\_W4D4 7.653917e+02  
## 37 UAB006\_W4D5 5.108480e+02  
## 38 UAB006\_W4D6 1.240235e+00  
## 39 UAB006\_W4D7 3.098208e+02  
## 40 UAB006\_W5D1 2.580749e+02  
## 41 UAB007\_W3D7 1.683195e+03  
## 42 UAB007\_W4D1 1.188735e+03  
## 43 UAB007\_W4D2 7.899422e+02  
## 44 UAB007\_W4D3 3.108406e+02  
## 45 UAB007\_W4D4 1.085003e+03  
## 46 UAB008\_W10D1 1.179684e+02  
## 47 UAB008\_W10D2 2.383264e+02  
## 48 UAB008\_W9D5 1.046533e+02  
## 49 UAB008\_W9D6 6.643826e+02  
## 50 UAB008\_W9D7 9.899674e+02  
## 51 UAB015\_W4D3 1.701653e+03  
## 52 UAB015\_W4D4 8.764815e+02  
## 53 UAB015\_W4D5 2.096003e+02  
## 54 UAB015\_W4D7 1.426799e+03  
## 55 UAB015\_W5D1 2.021888e+03  
## 56 UAB015\_W5D2 4.693895e+02  
## 57 UAB015\_W5D3 6.060877e+02  
## 58 UAB015\_W5D4 3.483291e+02  
## 59 UAB015\_W5D5 1.305746e+02  
## 60 UAB015\_W5D7 6.715677e+02  
## 61 UAB015\_W6D1 2.496645e+02  
## 62 UAB015\_W6D2 4.994072e+02  
## 63 UAB015\_W6D3 6.873087e+02  
## 64 UAB015\_W6D4 6.197250e+02  
## 65 UAB021\_W1D4 3.699508e+02  
## 66 UAB021\_W1D5 4.337148e+02  
## 67 UAB021\_W1D7 1.449508e+02  
## 68 UAB021\_W2D2 1.139957e+02  
## 69 UAB022\_W7D5 6.055661e+02  
## 70 UAB022\_W7D6 1.071701e+02  
## 71 UAB022\_W7D7 1.095501e+03  
## 72 UAB022\_W8D1 2.092547e+02  
## 73 UAB022\_W8D2 9.373751e+02  
## 74 UAB055\_W2D1 7.657517e+02  
## 75 UAB055\_W2D2 5.755693e+02  
## 76 UAB055\_W2D3 3.015522e+02  
## 77 UAB055\_W2D4 1.686811e+02  
## 78 UAB055\_W2D5 1.282814e+02  
## 79 UAB055\_W2D6 4.453451e+01  
## 80 UAB093\_W6D5 1.055765e+03  
## 81 UAB093\_W6D6 3.812033e+02  
## 82 UAB093\_W6D7 1.968737e+02  
## 83 UAB102\_W4D6 8.767737e+01  
## 84 UAB102\_W5D1 1.119551e+02  
## 85 UAB102\_W5D2 2.917887e+02  
## 86 UAB115\_W3D7 4.359042e+02  
## 87 UAB115\_W4D3 1.105002e+03  
## 88 UAB115\_W4D4 8.179583e+02  
## 89 UAB115\_W4D5 1.705125e+03  
## 90 UAB115\_W4D6 2.974179e+02  
## 91 UAB115\_W5D2 1.155358e+03  
## 92 UAB115\_W5D3 7.861445e+02  
## 93 UAB115\_W5D5 1.667892e+02  
## 94 UAB115\_W5D7 1.417144e+03  
## 95 UAB116\_W2D5 1.440792e+02  
## 96 UAB116\_W2D6 1.961594e+02  
## 97 UAB116\_W3D1 2.916645e+02  
## 98 UAB116\_W3D2 3.231621e+02  
## 99 UAB116\_W3D3 8.325147e+01  
## 100 UAB117\_W7D1 1.419449e+02  
## 101 UAB117\_W7D2 3.295152e+02  
## 102 UAB117\_W7D3 8.492167e+02  
## 103 UAB117\_W7D4 1.340304e+03  
## 104 UAB121\_W3D3 5.511017e+02  
## 105 UAB121\_W3D4 1.048689e-01  
## 106 UAB121\_W3D5 3.092622e+02  
## 107 UAB121\_W3D6 1.049983e+03  
## 108 UAB121\_W3D7 2.265420e+02  
## 109 UAB121\_W4D2 1.407170e-01  
## 110 UAB121\_W4D3 8.314880e+02  
## 111 UAB121\_W4D4 8.507223e-02  
## 112 UAB121\_W4D5 2.409049e+03  
## 113 UAB121\_W4D6 9.106602e+02

hist(log(colSums(SRL\_counts\_table)/number\_mirs\_genome,10),main="Log10 Total miRNA Counts/# miRNAs")  
abline(v = log(100,10))



.1\*number\_mirs\_genome

## Features   
## 186.9

## Samples with "low coverage", i.e., less than 1E4 reads/#miRs  
low\_coverage<-(colSums(SRL\_counts\_table))<=(125)\*number\_mirs\_genome ##125  
sum(low\_coverage)

## [1] 13

125 \* 1869

## [1] 233625

## An idea of the samples with "low coverage":  
print(head(SRL\_counts\_table[,low\_coverage]))

## UAB005\_W5D3 UAB006\_W4D6 UAB008\_W10D1 UAB008\_W9D5 UAB021\_W2D2  
## hsa-let-7a-2 602 36 2241 2463 2788  
## hsa-let-7a-3 618 37 2288 2529 2905  
## hsa-let-7b 1868 229 7186 11804 17191  
## hsa-let-7c 657 61 2766 5179 4186  
## hsa-let-7d 98 1 271 262 338  
## hsa-let-7e 10 2 88 81 36  
## UAB022\_W7D6 UAB055\_W2D6 UAB102\_W4D6 UAB102\_W5D1 UAB116\_W3D3  
## hsa-let-7a-2 3438 1652 2340 2646 1296  
## hsa-let-7a-3 3531 1621 2487 2713 1302  
## hsa-let-7b 20351 6993 19034 27855 11430  
## hsa-let-7c 11309 3736 6052 12083 5938  
## hsa-let-7d 792 206 305 413 180  
## hsa-let-7e 161 48 54 64 33  
## UAB121\_W3D4 UAB121\_W4D2 UAB121\_W4D4  
## hsa-let-7a-2 10 9 10  
## hsa-let-7a-3 9 15 10  
## hsa-let-7b 25 23 10  
## hsa-let-7c 5 9 10  
## hsa-let-7d 0 1 0  
## hsa-let-7e 0 0 0

(removed\_samples<-remove\_poorQC\_samples(removed\_samples = removed\_samples,sample\_list =unique(colnames(SRL\_counts\_table[,low\_coverage])),reason = "Low\_Coverage"))

## Pre\_QC\_ID QC\_removal\_stage  
## 1 UAB005\_W5D3 Low\_Coverage  
## 2 UAB006\_W4D6 Low\_Coverage  
## 3 UAB008\_W10D1 Low\_Coverage  
## 4 UAB008\_W9D5 Low\_Coverage  
## 5 UAB021\_W2D2 Low\_Coverage  
## 6 UAB022\_W7D6 Low\_Coverage  
## 7 UAB055\_W2D6 Low\_Coverage  
## 8 UAB102\_W4D6 Low\_Coverage  
## 9 UAB102\_W5D1 Low\_Coverage  
## 10 UAB116\_W3D3 Low\_Coverage  
## 11 UAB121\_W3D4 Low\_Coverage  
## 12 UAB121\_W4D2 Low\_Coverage  
## 13 UAB121\_W4D4 Low\_Coverage

## Remove those samples with low coverage   
counts\_meta.qc<-subset\_ExpressionSet(expSet = SRL\_counts\_meta,filterOut = c(as.character(removed\_samples$Pre\_QC\_ID)))  
  
nrow(pData(counts\_meta.qc))

## [1] 100

ncol(exprs(counts\_meta.qc))

## [1] 100

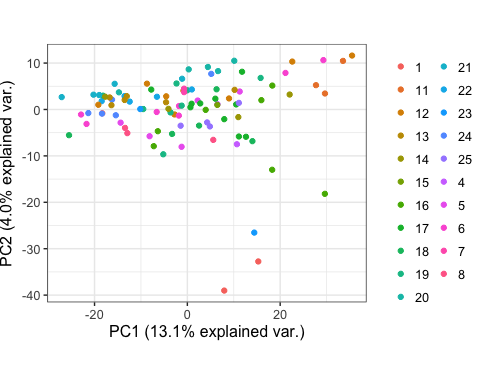
ncol(pData(counts\_meta.qc))

## [1] 50

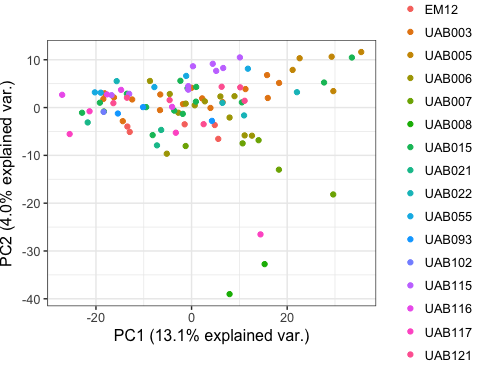
nrow(exprs(counts\_meta.qc))

## [1] 1546

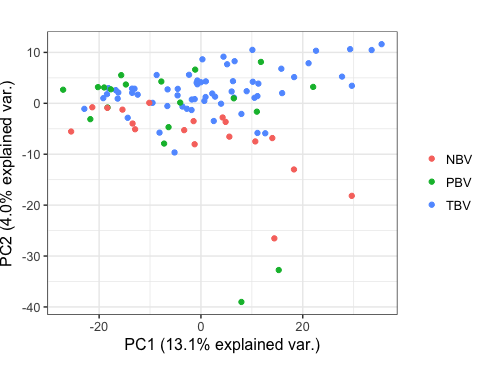
## Re-plot after QC  
plot\_QC.postQC.batch<-plot\_pca(counts\_meta.qc,"Log miRNA Raw Count Distance, POST QC",color\_by ="Batch",ploly = F,seed\_val = seed\_val)##FIGURE  
plot\_QC.postQC.batch$pca.p



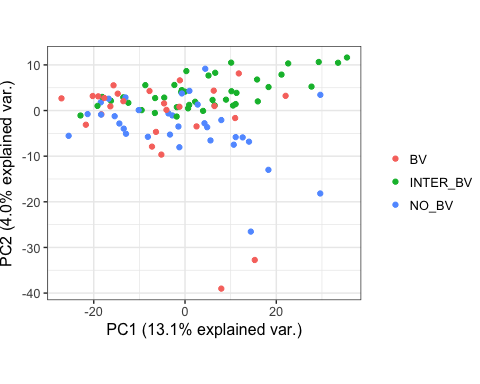
plot\_QC.postQC.sid<-plot\_pca(counts\_meta.qc,"Log miRNA Raw Count Distance, POST QC",color\_by = "SID",ploly = F,seed\_val = seed\_val)##FIGURE  
plot\_QC.postQC.sid$pca.p



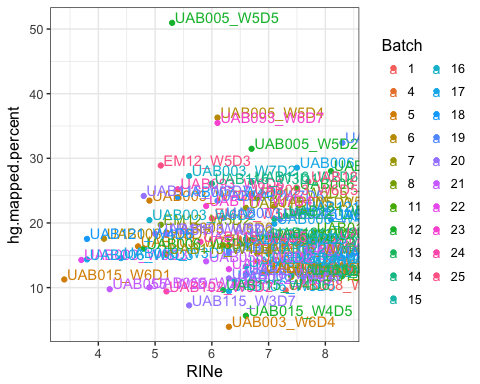
plot\_pca(counts\_meta.qc,"Log miRNA Raw Count Distance, POST QC",color\_by = "BVGroup",ploly = F,seed\_val = seed\_val)$pca.p##FIGURE



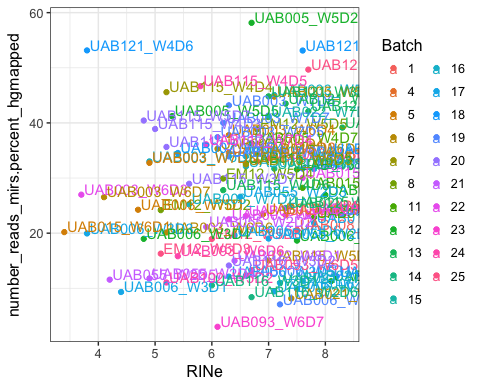
plot\_pca(counts\_meta.qc,"Log miRNA Raw Count Distance, POST QC",color\_by = "NugentC",ploly = F,seed\_val = seed\_val)$pca.p##FIGURE



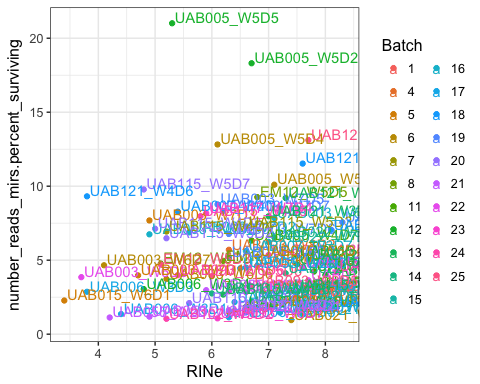
plot\_RIN\_meta(pData(counts\_meta.qc),"hg.mapped.percent",col\_by = "Batch")##FIGURE



plot\_RIN\_meta(pData(counts\_meta.qc),"number\_reads\_mirs.percent\_hgmapped",col\_by = "Batch")##FIGURE



plot\_RIN\_meta(pData(counts\_meta.qc),"number\_reads\_mirs.percent\_surviving",col\_by = "Batch")##FIGURE



dev.off()

## null device   
## 1

median\_RIN<-median(pData(counts\_meta.qc)$RIN,na.rm = T)  
  
print(paste0("The median RINe score was ",median\_RIN,", with ",round(100\*sum(pData((counts\_meta.qc))$RIN<7,na.rm = T)/length(pData((counts\_meta.qc))$RIN),digits = 1),"% of the samples having a RINe greater than 7."))

## [1] "The median RINe score was 6.6, with 60% of the samples having a RINe greater than 7."

number\_miRNAs<-mapping\_stats(pData(counts\_meta.qc)$number\_reads\_mirs)  
percent\_miRNAs.hg<-round(mapping\_stats(pData(counts\_meta.qc)$number\_reads\_mirs.percent\_hgmapped),1)  
percent\_miRNAs.oftrimmed<-round(mapping\_stats(pData(counts\_meta.qc)$number\_reads\_mirs.percent\_surviving),1)  
  
print(paste0("The median (minimum/maximum) percentage of post-QC miRNA reads relative to all hg19 mapped and total reads was ",percent\_miRNAs.hg$median,"% (",percent\_miRNAs.hg$min,"%/",percent\_miRNAs.hg$max,"%) and ",percent\_miRNAs.oftrimmed$median,"% (",percent\_miRNAs.oftrimmed$min,"%/",percent\_miRNAs.oftrimmed$max,"%), respectively. However, the median number of post-QC hg19 mapped miRNA reads was ",number\_miRNAs$median,", with a minimum ",number\_miRNAs$min," and maximum ",number\_miRNAs$max,". Thus, despite low relative miRNA read counts, the estimated coverage ranged from ",round(number\_miRNAs$min/number\_mirs\_genome,digits = 0),"X-",round(number\_miRNAs$max/number\_mirs\_genome,digits = 0),"X across the entire miRnome (",number\_mirs\_genome," annotated miRNAs)."))

## [1] "The median (minimum/maximum) percentage of post-QC miRNA reads relative to all hg19 mapped and total reads was 26.6% (3%/58.2%) and 4.2% (0.5%/21%), respectively. However, the median number of post-QC hg19 mapped miRNA reads was 1204913, with a minimum 239758 and maximum 4642910. Thus, despite low relative miRNA read counts, the estimated coverage ranged from 128X-2484X across the entire miRnome (1869 annotated miRNAs)."

sum(pData((counts\_meta.qc))$number\_reads\_mirs<10^6,na.rm = T)/length(pData((counts\_meta.qc))$number\_reads\_mirs)

## [1] 0.39

cairo\_ps(file = paste0(thesis\_figures\_directory,FIGURE\_QC\_PCA.PREQC.BYBATCH),width = 8,height = 5.5)  
plot(plot\_QC.preQC.batch$pca.p)  
dev.off()

## null device   
## 1

cairo\_ps(file = paste0(thesis\_figures\_directory,FIGURE\_QC\_PCA.PREQC.BYSUBJ),width = 8,height = 5.5)  
plot(plot\_QC.preQC.sid$pca.p)  
dev.off()

## null device   
## 1

cairo\_ps(file = paste0(thesis\_figures\_directory,FIGURE\_QC\_PCA.RMLOW.BYBATCH),width = 8,height = 5.5)  
plot(plot\_QC.postQC.batch$pca.p)  
dev.off()

## null device   
## 1

cairo\_ps(file = paste0(thesis\_figures\_directory,FIGURE\_QC\_PCA.RMLOW.BYSUBJ),width = 8,height = 5.5)  
plot(plot\_QC.postQC.sid$pca.p)  
dev.off()

## null device   
## 1

cairo\_ps(file = paste0(thesis\_figures\_directory,FIGURE\_QC\_RIN\_v\_READS.PROP),width = 8,height = 5.5)  
plot(plot\_QC.RIN.percent.batch)  
dev.off()

## null device   
## 1

cairo\_ps(file = paste0(thesis\_figures\_directory,FIGURE\_QC\_RIN\_v\_READS.ABS),width = 8,height = 5.5)  
plot(plot\_QC.RIN.number.batch)  
dev.off()

## null device   
## 1

# Discovery of miRNAs Associated w/ Lactobacillus spp. vs CST-IV/BV

## Build a proxy-Amsel model from Clinical Visits

Not all samples in the miRNA cohort have Amsel scores, but they do have 16S rRNA sequencing taxa assignments. Therefore:  
1. Build a RF model that uses a separate cohort to predict Amsel-16S rRNA associations.  
2. Assess accuracy, and tweak parameters or predictors.  
3. Using this model, assign proxy-Amsel scores to the miRNA cohort from the available 16S rRNA data.

## Generate a model using clinical visit data. The model uses 16S relative abundances to predict Amsel diagnosis.   
load(file=paste0(R\_script\_input\_directory,"CV\_16S\_AMSEL.Rdata"))  
load(file=paste0(R\_script\_input\_directory,"SRL\_16S.Rdata"))  
SRL\_16S<-SRL\_16S[!row.names(SRL\_16S) %in% as.character(removed\_samples$Pre\_QC\_ID),]   
  
ncol(CV\_16S\_AMSEL)

## [1] 204

setdiff(names(CV\_16S\_AMSEL),names(SRL\_16S))

## [1] "AMSEL" "Amsel\_BV"

setdiff(names(SRL\_16S),names(CV\_16S\_AMSEL))

## character(0)

table(CV\_16S\_AMSEL$Amsel\_BV)

##   
## NBV PBV   
## 210 71

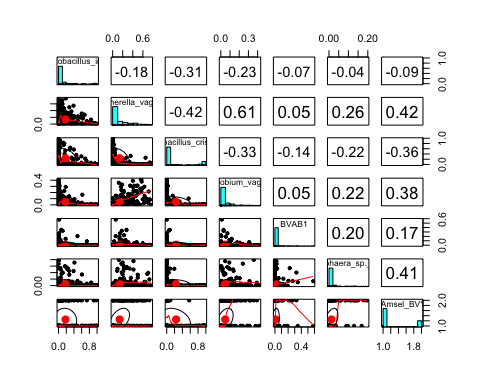
nuniq\_subjs\_amsel<-length(unique(gsub(row.names(CV\_16S\_AMSEL),pattern = "\_.\*",replacement = "")))  
  
print(paste0('Therefore, a proxy-Amsel diagnosis was developed by applying a Random Forest model trained with metataxonomic data and metadata from ',nuniq\_subjs\_amsel,' subjects of the parent cohort that included ',nrow(CV\_16S\_AMSEL),' samples for which both metataxonomic data and Amsel diagnosis was available (Amsel subset)(',TABLE\_PROXY\_AMSEL\_INPUT,').'))

## [1] "Therefore, a proxy-Amsel diagnosis was developed by applying a Random Forest model trained with metataxonomic data and metadata from 117 subjects of the parent cohort that included 281 samples for which both metataxonomic data and Amsel diagnosis was available (Amsel subset)(TABLE\_A7.csv)."

NBV.amsel<-table(CV\_16S\_AMSEL$Amsel\_BV)["NBV"]  
PBV.amsel<-table(CV\_16S\_AMSEL$Amsel\_BV)["PBV"]  
  
  
print(paste0("In the Amsel subset, asymptomatic or symptomatic BV diagnoses represented ",round(100\*PBV.amsel/(NBV.amsel+PBV.amsel),digits = 1),"% (",PBV.amsel,"/",NBV.amsel+PBV.amsel,") of the Amsel diagnoses (",TABLE\_PROXY\_AMSEL\_INPUT,"), a figure closely matching the reported prevalence of 29.2% for BV in similar populations [REF]."))

## [1] "In the Amsel subset, asymptomatic or symptomatic BV diagnoses represented 25.3% (71/281) of the Amsel diagnoses (TABLE\_A7.csv), a figure closely matching the reported prevalence of 29.2% for BV in similar populations [REF]."

##Sanity check for Amsel  
pairs.panels(dplyr::select(CV\_16S\_AMSEL,Lactobacillus\_iners,Gardnerella\_vaginalis,Lactobacillus\_crispatus,Atopobium\_vaginae,BVAB1,Megasphaera\_sp.\_type\_1,Amsel\_BV),scale = F,density = F)



CV\_16S\_AMSEL <- CV\_16S\_AMSEL %>% dplyr::select(-c(AMSEL))  
min\_CV\_16S\_AMSEL<-min(dplyr::select(CV\_16S\_AMSEL,-Amsel\_BV)[dplyr::select(CV\_16S\_AMSEL,-Amsel\_BV)>0],na.rm = T)  
CV\_16S\_AMSEL[is.na(CV\_16S\_AMSEL)]<-0  
CV\_16S\_AMSEL\_IDs=row.names(CV\_16S\_AMSEL)  
  
  
## See run\_randomForest function above for paramaters. Note that you can load a previously-built model to save time.   
  
Clinical\_Visit\_RF<-run\_randomForest(predictors\_response\_table = CV\_16S\_AMSEL,  
 response\_variable\_name = "Amsel\_BV",  
 subj\_spec = FALSE,  
 nfold = nfolds,  
 nreps = npermutes,  
 save\_model = F,  
 file\_n = "Clinical\_Visit\_RF",  
 verbose = F,  
 permute = T,  
 load\_prev\_model = T,  
 R\_script\_output\_directory=R\_script\_output\_directory)   
  
Clinical\_Visit\_RF$accuracy\_table

## Var2  
## Var1 NBV PBV  
## NBV 55 5  
## PBV 8 16

(Clinical\_Visit\_RF.accuracy<-c(diag(Clinical\_Visit\_RF$accuracy\_table)/rowSums(Clinical\_Visit\_RF$accuracy\_table)))

## NBV PBV   
## 0.9166667 0.6666667

(Clinical\_Visit\_RF$top\_features$top\_features.all<-row.names(Clinical\_Visit\_RF$importance[Clinical\_Visit\_RF$importance$MeanDecreaseAccuracy.pval<=pval\_threshold | Clinical\_Visit\_RF$importance$MeanDecreaseGini.pval<=pval\_threshold,]))

## [1] "Parvimonas\_micra" "Megasphaera\_sp.\_type\_1"   
## [3] "BVAB2" "Eggerthella"   
## [5] "Dialister\_sp.\_type\_2" "Leptotrichia\_amnionii"   
## [7] "BVAB3" "BVAB1"   
## [9] "PH" "Prevotella\_genogroup\_3"   
## [11] "Prevotella\_genogroup\_1" "Mobiluncus\_mulieris"   
## [13] "Porphyromonas\_sp.\_type\_1" "Gardnerella\_vaginalis"   
## [15] "Prevotella\_genogroup\_2" "Peptoniphilus\_lacrimalis"   
## [17] "Atopobium\_vaginae" "MENSTRUATION\_NORMALIZED\_PHASED"  
## [19] "Prevotella\_genogroup\_4" "Candidate\_Division\_TM7\_vaginal"  
## [21] "Lactobacillus\_crispatus" "Prevotella\_genogroup\_5"   
## [23] "Porphyromonas\_uenonis" "Gemella"   
## [25] "Anaerococcus\_vaginalis" "Lactobacillus\_helveticus"   
## [27] "Prevotella\_melaninogenica" "Lactobacillus\_vaginalis"   
## [29] "Porphyromonas\_endodontalis" "Sutterella\_stercoricanis"   
## [31] "Firmicutes" "Anaerococcus"

print(paste0("Multiple (",length(Clinical\_Visit\_RF$top\_features$top\_features.all),") important microbial (taxa and their relative abundance) or metadata features were predictive of the Amsel diagnosis: ",str\_c(sort(Clinical\_Visit\_RF$top\_features$top\_features.all),collapse = ", ")," (",TABLE\_RF\_SUMMARY.CV,")"))

## [1] "Multiple (32) important microbial (taxa and their relative abundance) or metadata features were predictive of the Amsel diagnosis: Anaerococcus, Anaerococcus\_vaginalis, Atopobium\_vaginae, BVAB1, BVAB2, BVAB3, Candidate\_Division\_TM7\_vaginal, Dialister\_sp.\_type\_2, Eggerthella, Firmicutes, Gardnerella\_vaginalis, Gemella, Lactobacillus\_crispatus, Lactobacillus\_helveticus, Lactobacillus\_vaginalis, Leptotrichia\_amnionii, Megasphaera\_sp.\_type\_1, MENSTRUATION\_NORMALIZED\_PHASED, Mobiluncus\_mulieris, Parvimonas\_micra, Peptoniphilus\_lacrimalis, PH, Porphyromonas\_endodontalis, Porphyromonas\_sp.\_type\_1, Porphyromonas\_uenonis, Prevotella\_genogroup\_1, Prevotella\_genogroup\_2, Prevotella\_genogroup\_3, Prevotella\_genogroup\_4, Prevotella\_genogroup\_5, Prevotella\_melaninogenica, Sutterella\_stercoricanis (TABLE\_A8.csv)"

print(paste0("The Amsel Random Forest model accuracy was tested using a hold-out set and found to be ",round(100\*Clinical\_Visit\_RF.accuracy["NBV"],1),"% accurate in correctly assigning NBV diagnosis and ",round(100\*Clinical\_Visit\_RF.accuracy["PBV"],1),"% accurate in correctly assigning PBV diagnosis."))

## [1] "The Amsel Random Forest model accuracy was tested using a hold-out set and found to be 91.7% accurate in correctly assigning NBV diagnosis and 66.7% accurate in correctly assigning PBV diagnosis."

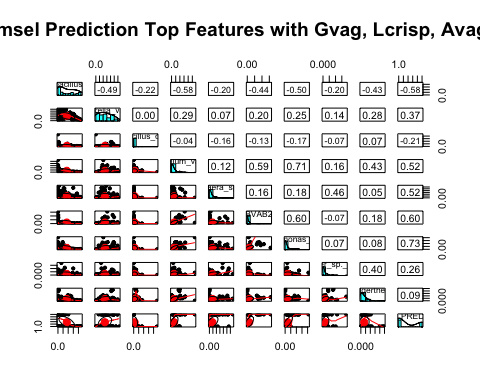
proxy\_amsel\_predictions.classprobs<-predict(Clinical\_Visit\_RF$rfp, SRL\_16S, type = "prob")  
(proxy\_amsel\_predictions<-predict(Clinical\_Visit\_RF$rfp, SRL\_16S, type = "response"))

## UAB003\_W5D6 UAB003\_W6D1 UAB003\_W6D2 UAB003\_W6D3 UAB003\_W6D4 UAB003\_W6D5   
## NBV NBV NBV NBV NBV NBV   
## UAB003\_W6D6 UAB003\_W6D7 UAB003\_W7D1 UAB003\_W7D2 UAB003\_W7D3 UAB003\_W7D5   
## NBV NBV NBV NBV NBV NBV   
## UAB003\_W7D6 UAB003\_W7D7 UAB003\_W8D1 UAB003\_W8D2 UAB005\_W4D7 UAB005\_W5D1   
## NBV NBV NBV NBV NBV NBV   
## UAB005\_W5D2 UAB005\_W5D4 UAB005\_W5D5 UAB006\_W3D3 UAB006\_W3D4 UAB006\_W3D5   
## NBV NBV NBV NBV NBV NBV   
## UAB006\_W3D6 UAB006\_W3D7 UAB006\_W4D2 UAB006\_W4D3 UAB006\_W4D4 UAB006\_W4D5   
## NBV NBV PBV PBV PBV PBV   
## UAB006\_W4D7 UAB006\_W5D1 UAB007\_W3D7 UAB007\_W4D1 UAB007\_W4D2 UAB007\_W4D3   
## PBV PBV NBV NBV NBV NBV   
## UAB007\_W4D4 UAB008\_W9D6 UAB015\_W4D3 UAB015\_W4D4 UAB015\_W4D5 UAB015\_W4D7   
## NBV PBV PBV PBV PBV PBV   
## UAB015\_W5D1 UAB015\_W5D2 UAB015\_W5D3 UAB015\_W5D4 UAB015\_W5D5 UAB015\_W5D7   
## PBV PBV PBV NBV PBV PBV   
## UAB015\_W6D1 UAB015\_W6D2 UAB015\_W6D3 UAB015\_W6D4 UAB021\_W1D5 UAB021\_W1D7   
## NBV NBV PBV NBV PBV PBV   
## UAB022\_W8D1 UAB055\_W2D1 UAB055\_W2D2 UAB055\_W2D3 UAB055\_W2D4 UAB055\_W2D5   
## PBV PBV PBV PBV PBV PBV   
## UAB093\_W6D6 UAB115\_W3D7 UAB115\_W4D3 UAB115\_W4D4 UAB115\_W4D6 UAB115\_W5D2   
## NBV NBV NBV NBV NBV NBV   
## UAB115\_W5D3 UAB115\_W5D5 UAB115\_W5D7 UAB116\_W2D5 UAB116\_W2D6 UAB116\_W3D1   
## NBV NBV NBV PBV PBV PBV   
## UAB116\_W3D2 UAB117\_W7D2 UAB117\_W7D4 UAB121\_W3D6 UAB121\_W4D3 UAB121\_W4D5   
## PBV NBV NBV NBV NBV NBV   
## EM12\_W5D2 EM12\_W5D3 EM12\_W5D4 EM12\_W5D5 EM12\_W5D6   
## NBV NBV NBV NBV NBV   
## Levels: NBV PBV

CV\_16S\_AMSEL.out<-CV\_16S\_AMSEL  
CV\_16S\_AMSEL.out$Amsel\_BV\_trainingTestingSet<-""  
CV\_16S\_AMSEL.out[row.names(CV\_16S\_AMSEL.out) %in% Clinical\_Visit\_RF$testing\_ids,"Amsel\_BV\_trainingTestingSet"]<-"testing"  
CV\_16S\_AMSEL.out[row.names(CV\_16S\_AMSEL.out) %in% Clinical\_Visit\_RF$training\_ids,"Amsel\_BV\_trainingTestingSet"]<-"training"  
  
write.csv(CV\_16S\_AMSEL.out,file=paste0(thesis\_tables\_directory,TABLE\_PROXY\_AMSEL\_INPUT),row.names=T,quote=F)  
  
SRL\_16S\_AMSEL<-cbind(SRL\_16S,AMSEL\_PREDICTION=proxy\_amsel\_predictions)  
table(SRL\_16S\_AMSEL$AMSEL\_PREDICTION)

##   
## NBV PBV   
## 54 29

write.csv(SRL\_16S\_AMSEL,file=paste0(thesis\_tables\_directory,TABLE\_PROXY\_AMSEL\_SRL),row.names=T,quote=F)  
  
##Sanity Check  
nfeats\_plot<-9  
pairs\_panel\_data<-SRL\_16S\_AMSEL[,names(SRL\_16S\_AMSEL) %in% c(unique(c("Lactobacillus\_iners","Gardnerella\_vaginalis","Lactobacillus\_crispatus","Atopobium\_vaginae",Clinical\_Visit\_RF$top\_features$top\_features.all)[1:nfeats\_plot]),"AMSEL\_PREDICTION")]  
pairs.panels(pairs\_panel\_data,main="proxy-Amsel Prediction Top Features with Gvag, Lcrisp, Avag, Liners")

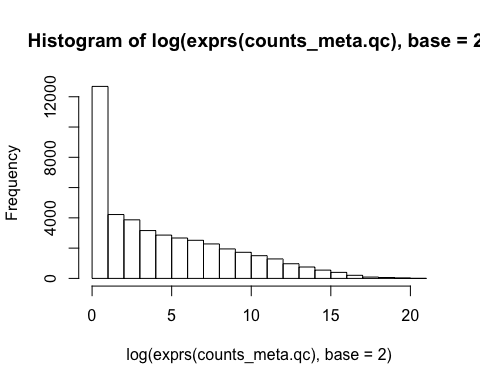


dev.off()

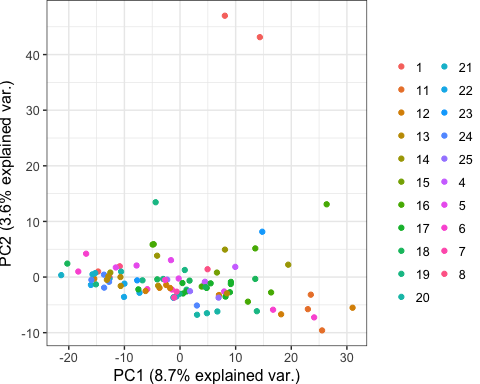
## null device   
## 1

## Normalize & log2 transform small RNA-Seq counts for use in models

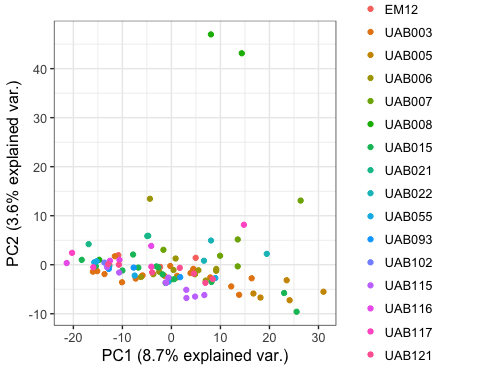
## Calcualte size factors for normalization using calcNormFactors function  
hist(log(exprs(counts\_meta.qc),base = 2))



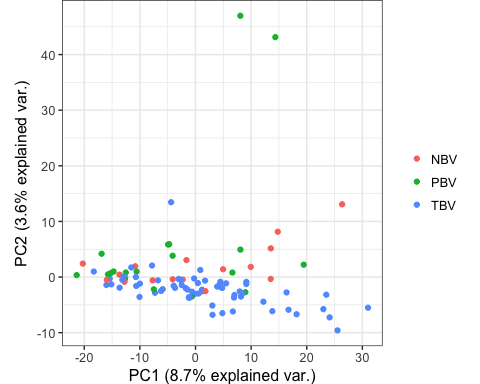
expSet.sizeDisp<-calcNormFactors(method = "TMM",DGEList(counts = exprs(counts\_meta.qc)),logratioTrim = 0.10,sumTrim = 0.05)   
  
## Normalize counts using size dispersions and norm factors, store as Expression Set  
expSet.normalized<-ExpressionSet(assayData = as.matrix(1E6\*expSet.sizeDisp$counts\*expSet.sizeDisp$samples$norm.factors/expSet.sizeDisp$samples$lib.size),phenoData = AnnotatedDataFrame(pData(counts\_meta.qc)))  
  
## Store log-transformed QC \*NON NORMALIZED\* count table for PCA. This is not used for anything but PCA.   
expSet\_log<-ExpressionSet(assayData = log(exprs(counts\_meta.qc),base=2),phenoData = AnnotatedDataFrame(pData(counts\_meta.qc)))  
  
## Store log-transformed QC normalized count table. This is the small RNAseq count table used in all downstream analysis  
expSet\_log.normalized<-ExpressionSet(assayData = log(exprs(expSet.normalized)+1,base=2),  
 phenoData = AnnotatedDataFrame(pData(expSet.normalized))) ## add pseuo counts... so that when take log(), all 0 reads become 0 normalized reads log(raw + 1)=log(1)=0. LIttle effect on other counts. This makes it esier to drop raw 0 counts later. Also store count table annotation from original Expression Set  
  
## PCA plots after normalization, colored by batch, SID and "BV group"  
plot\_QC.normalized.batch<-plot\_pca(expSet.normalized,color\_by = "Batch",margins = unit(c(0,-200,0,-200),units = "points"),seed\_val = seed\_val)  
plot\_QC.normalized.batch$pca.p



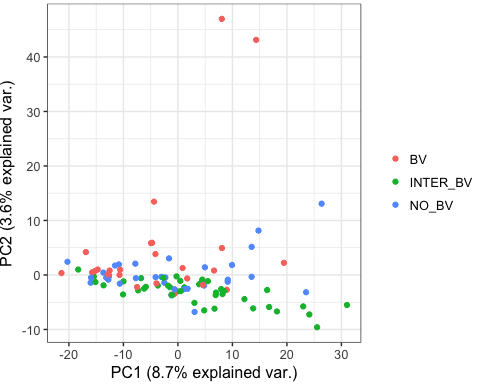
plot\_QC.normalized.sid<-plot\_pca(expSet\_log.normalized,color\_by = "SID",ploly = F,logt = F,margins = unit(c(0,30,0,5),units = "pt"),seed\_val = seed\_val)  
plot\_QC.normalized.sid$pca.p



plot\_pca(expSet\_log.normalized,color\_by = "BVGroup",logt = F)$pca.p



plot\_pca(expSet\_log.normalized,color\_by = "NugentC",logt = F)$pca.p



## Print for text  
print(paste0("PCA plots before and after sample removal and after normalization do not support batch effects (guided PCA p-values ",plot\_QC.preQC.batch$gPCA.result$p.val,", ",plot\_QC.postQC.batch$gPCA.result$p.val,", and ",plot\_QC.normalized.batch$gPCA.result$p.val,", respectively), or subject-specific effects (guided PCA p-values ",plot\_QC.preQC.sid$gPCA.result$p.val,", ",plot\_QC.postQC.sid$gPCA.result$p.val,", and ",plot\_QC.normalized.sid$gPCA.result$p.val,", respectively, Figure S1)."))

## [1] "PCA plots before and after sample removal and after normalization do not support batch effects (guided PCA p-values 0.994, 0.267, and 0.124, respectively), or subject-specific effects (guided PCA p-values 0.288, 0.478, and 0.294, respectively, Figure S1)."

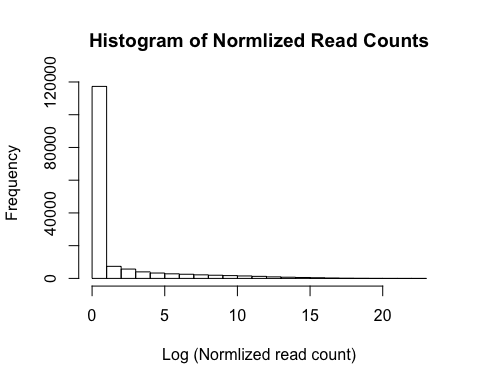
## Store counts data as "features"  
features<-exprs(expSet\_log.normalized)  
head(features)[,1:3]

## EM12\_W5D2 EM12\_W5D3 EM12\_W5D4  
## hsa-let-7a-2 14.554033 13.956142 13.936057  
## hsa-let-7a-3 14.750121 14.559435 15.451481  
## hsa-let-7b 16.993347 19.089923 18.246961  
## hsa-let-7c 15.085236 14.779020 15.343028  
## hsa-let-7d 10.933057 10.523169 12.401490  
## hsa-let-7e 8.730594 9.655014 9.324838

## Replace NA values with half the minimum normalized count values so that modelling works with algorithms  
(min\_normalized\_count<-2^min(features,na.rm = T))

## [1] 1

features[is.na(features)]<-log(min\_normalized\_count/2,base = 2)  
  
## Remove low count miRNAs  
hist(features,main="Histogram of Normlized Read Counts",xlab="Log (Normlized read count)")  
abline(v = -2,col='red')



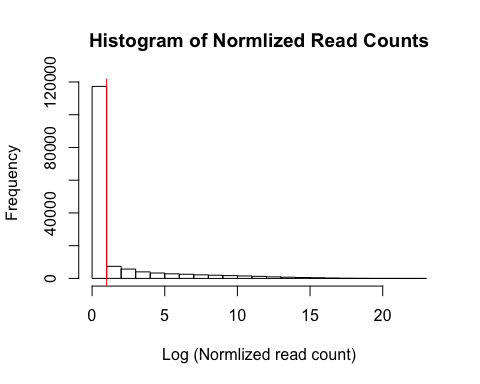
nrow(features)

## [1] 1546

keep\_features<-rowSums(features>log(min\_normalized\_count/2,base = 2))>=(.5\*ncol(features))  
sum(keep\_features)

## [1] 1546

features<-features[keep\_features,] ## At least half of normalized miRNA counts should be >1.   
hist(features,main="Histogram of Normlized Read Counts",xlab="Log (Normlized read count)")  
abline(v = 1,col='red')



## Write PCA plots to file  
cairo\_ps(file=paste0(thesis\_figures\_directory,FIGURE\_QC\_PCA.NORMAL.BYBATCH),width=8,height=5.5)  
plot\_QC.normalized.batch$pca.p  
dev.off()

## quartz\_off\_screen   
## 2

cairo\_ps(file=paste0(thesis\_figures\_directory,FIGURE\_QC\_PCA.NORMAL.BYSUBJ),width=8,height=5.5)  
plot\_QC.normalized.sid$pca.p  
dev.off()

## quartz\_off\_screen   
## 2

## Prepare & Analyze the Nugent and Amsel input tables (predictors) to RF model

## Join the existing metdadata and predicted Amsel diagnosis  
joined\_meta\_amselPred<-join(pData(counts\_meta.qc),data.frame(Pre\_QC\_ID=names(proxy\_amsel\_predictions),AMSEL\_prediction=proxy\_amsel\_predictions),by="Pre\_QC\_ID")  
  
row.names(joined\_meta\_amselPred)<-joined\_meta\_amselPred$Pre\_QC\_ID  
  
##Store the joined data as features\_metadata  
features\_metadata<-ExpressionSet(features,phenoData = AnnotatedDataFrame(joined\_meta\_amselPred))  
  
## Summarize postQC counts data/metadata  
summary\_sid.group<-ddply(pData(counts\_meta.qc),c("SID","BVGroup"),summarise,N=length(SID))  
(summary\_group.sid<-ddply(pData(counts\_meta.qc),c("BVGroup","SID"),summarise,N=length(SID)))

## BVGroup SID N  
## 1 NBV EM12 5  
## 2 NBV UAB007 5  
## 3 NBV UAB093 3  
## 4 NBV UAB102 1  
## 5 NBV UAB117 4  
## 6 PBV UAB008 3  
## 7 PBV UAB021 3  
## 8 PBV UAB022 4  
## 9 PBV UAB055 5  
## 10 PBV UAB116 4  
## 11 TBV UAB003 16  
## 12 TBV UAB005 5  
## 13 TBV UAB006 12  
## 14 TBV UAB015 14  
## 15 TBV UAB115 9  
## 16 TBV UAB121 7

(summary\_group<-ddply(pData(counts\_meta.qc),c("BVGroup"),summarise,N=length(SID)))

## BVGroup N  
## 1 NBV 18  
## 2 PBV 19  
## 3 TBV 63

(summary\_group.uniqSID<-ddply(unique(dplyr::select(pData(counts\_meta.qc),c(SID,BVGroup))),c("BVGroup"),summarise,N=length(SID)))

## BVGroup N  
## 1 NBV 5  
## 2 PBV 5  
## 3 TBV 6

(summary\_group.race<-ddply(unique(dplyr::select(pData(counts\_meta.qc),c(BVGroup,Race,SID))),c("BVGroup","Race"),summarise,N=length(SID)))

## BVGroup Race N  
## 1 NBV 1  
## 2 NBV B 3  
## 3 NBV W 1  
## 4 PBV B 5  
## 5 TBV B 5  
## 6 TBV W 1

print(paste0("A total of ",sum(summary\_sid.group$N)," samples, representing ",sum(summary\_group.uniqSID$N)," unique subjects from one of 3 longitudinal groups were used in the final analysis"))

## [1] "A total of 100 samples, representing 16 unique subjects from one of 3 longitudinal groups were used in the final analysis"

## Subset metadata for easier handling downstream  
selected\_metadata<-dplyr::select(pData(features\_metadata),c(SID,VAG\_INT,FING\_PEN,contains("SEX"),symptoms,symptoms\_nonBV,BirthControl,NUGENT\_SCORE,AMSEL\_prediction,Pre\_QC\_ID,MENSTRUATION\_NORMALIZED\_PHASED,CST)) ## CSTs are \*NOT\* used in model input, but are instead passed along to top mir expression plot/table downstream. Remove CST from running in model.   
  
##Store only counts expressed 'above 0' to reduce noise in RF models  
min(exprs(features\_metadata))

## [1] 0

expressed\_above\_0<-exprs(features\_metadata)[rowSums(exprs(features\_metadata)!=0)>=(1\*ncol(exprs(features\_metadata))),]   
  
## Combine non zero log transformed normalized counts with selected metadata as input into RF models (predictors + responses)  
model\_input<-data.frame(t(expressed\_above\_0),selected\_metadata)  
row.names(model\_input)<-model\_input$Pre\_QC\_ID  
model\_input<-dplyr::select(model\_input,-Pre\_QC\_ID)  
model\_input$AMSEL\_prediction<-as.character(model\_input$AMSEL\_prediction)  
names(model\_input)<-gsub(gsub(gsub(names(model\_input),pattern = "mir.",replacement = "miR-"),pattern = "hsa.",replacement = ""),replacement ="-",pattern = "\\.")  
  
##/// Subset model\_input into the inputs for Amsel and Nugent RF  
model\_input\_Amsel<-dplyr::select(model\_input[!is.na(model\_input$AMSEL\_prediction),],-c(NUGENT\_SCORE,CST))## Keep CSTs in model\_input, just not Amsel or Nugent input  
model\_input\_Nugent<-dplyr::select(model\_input[!is.na(model\_input$NUGENT\_SCORE),],-c(AMSEL\_prediction,CST)) ## Keep CSTs in model\_input, just not Amsel or Nugent input  
  
model\_input\_Amsel[is.na(model\_input\_Amsel)]<-0 ## RF can not have missing values. Replacing with 0's isn't technically valid, but should have a minimal effect on outcome  
model\_input\_Nugent[is.na(model\_input\_Nugent)]<-0 ## RF can not have missing values. Replacing with 0's isn't technically valid, but should have a minimal effect on outcome  
  
##ADD QC STAGE TO SRL\_meta\_table, then write to disk  
SRL\_meta\_table<-left\_join(SRL\_meta\_table,removed\_samples,by="Pre\_QC\_ID")  
print(paste0("There were 5 samples removed due to insufficient library material or failure to sequence, and ",sum(!is.na(SRL\_meta\_table$QC\_removal\_stage) & SRL\_meta\_table$QC\_removal\_stage %in% c("Low\_Coverage","Visual Outlier"))," samples were removed due to low total miRNA reads or outliers."))

## [1] "There were 5 samples removed due to insufficient library material or failure to sequence, and 13 samples were removed due to low total miRNA reads or outliers."

write.csv(SRL\_meta\_table,file=paste0(thesis\_tables\_directory,TABLE\_SRL\_METADATA),row.names=F,quote=F)  
SRL\_seq\_summary<-ddply(SRL\_meta\_table,c("BVGroup","SID"),summarise,PreQC=length(QC\_removal\_stage),PostQC=length(QC\_removal\_stage)-sum(as.numeric(QC\_removal\_stage),na.rm = T))  
SRL\_seq\_summary<-SRL\_seq\_summary[order(SRL\_seq\_summary$BVGroup),]  
write.csv(SRL\_seq\_summary,file=paste0(thesis\_tables\_directory,TABLE\_SEQSUMMARY),row.names=F,quote=F)

## Run Random Forest Models

Use RF to discover miRNAs associated w/ Lactobacillus spp. dominated communities vs CST-IV (BV-associated communities)  
\* Amsel- Random Forest (using predicted Amsel diagnoses as response variable)  
\* Nugent- Random Forest (using Nugent score as response variable)

## //////////  
### Amsel-RF  
## //////////  
## See run\_randomForest function above for paramaters. Note that you can load a previously-built model to save time. Note that if loaded from previous, most of the input is ignored.   
  
Amsel\_RF<-run\_randomForest(predictors\_response\_table = model\_input\_Amsel,  
 response\_variable\_name = "AMSEL\_prediction",  
 subj\_spec =T,  
 nfold = nfolds,  
 nreps = npermutes,  
 permute = T,  
 save\_model = F,  
 file\_n = "Amsel\_RF",  
 load\_prev\_model = T,  
 verbose = F,  
 training\_prop = training\_prop,  
 R\_script\_output\_directory=R\_script\_output\_directory)   
  
## Store top Amsel-RF features  
(Amsel\_RF$top\_features$top\_features.all<-row.names(Amsel\_RF$rfp$importance\_w\_pval[Amsel\_RF$rfp$importance\_w\_pval[,"MeanDecreaseGini.pval"]<=pval\_threshold | Amsel\_RF$rfp$importance\_w\_pval[,"MeanDecreaseAccuracy.pval"]<=pval\_threshold,]))

## [1] "let-7e" "miR-203b" "miR-184" "miR-193b" "let-7a-2" "miR-7-3"   
## [7] "miR-182" "miR-183" "miR-378a" "miR-3607" "miR-100" "miR-320a"  
## [13] "miR-10b" "miR-362" "miR-324" "miR-146a" "miR-16-1" "miR-4510"  
## [19] "miR-500a" "miR-342"

## Calculate accuracy  
NBV.accuracy.AMSEL<-100\*mean(sapply(Amsel\_RF$accuracy\_table, function(x) x["NBV","NBV"])/sapply(Amsel\_RF$accuracy\_table, function(x) sum(x["NBV",])))  
PBV.accuracy.AMSEL<-100\*mean(sapply(Amsel\_RF$accuracy\_table, function(x) x["PBV","PBV"])/sapply(Amsel\_RF$accuracy\_table, function(x) sum(x["PBV",])))  
  
print(paste0("The accuracy of the proxy-Amsel-RF model classification was ",round(NBV.accuracy.AMSEL,digits = 1),"% for NBV and ",round(PBV.accuracy.AMSEL,digits = 1),"% for PBV. There were ",length(Amsel\_RF$top\_features$top\_features.all)," significant miRNAs using proxy-Amsel RF."))

## [1] "The accuracy of the proxy-Amsel-RF model classification was 82.3% for NBV and 80.3% for PBV. There were 20 significant miRNAs using proxy-Amsel RF."

## //////////  
### Nugent-RF  
## //////////  
  
Nugent\_RF<-run\_randomForest(predictors\_response\_table = model\_input\_Nugent,  
 response\_variable\_name = "NUGENT\_SCORE",  
 subj\_spec = T,  
 nfold = nfolds,  
 nreps = npermutes,  
 permute = T,  
 save\_model = F,  
 file\_n = "Nugent\_RF",  
 load\_prev\_model = T,  
 verbose = F,  
 training\_prop = training\_prop,  
 R\_script\_output\_directory=R\_script\_output\_directory)   
  
##Store op Nugent RF features  
(Nugent\_RF$top\_features$top\_features.all<-row.names(Nugent\_RF$rfp$importance\_w\_pval[Nugent\_RF$rfp$importance\_w\_pval[,"%IncMSE.pval"]<=pval\_threshold | Nugent\_RF$rfp$importance\_w\_pval[,"IncNodePurity.pval"]<=pval\_threshold,]))

## [1] "miR-193b" "miR-203b" "miR-324" "miR-130a" "miR-224"   
## [6] "miR-182" "miR-149" "miR-3607" "miR-375" "miR-15a"   
## [11] "miR-3653" "miR-21" "miR-223" "miR-15b" "miR-378a"   
## [16] "miR-203a" "miR-200b" "miR-146a" "miR-205" "miR-16-2"   
## [21] "miR-152" "miR-199b" "miR-20a" "miR-95" "miR-197"   
## [26] "miR-365a" "miR-101-2" "miR-191" "miR-140" "miR-500a"

##Calculate accuracy/mean abs error  
accuracy\_table.Nugent<-Nugent\_RF$accuracy\_table  
(mean\_absolute\_error.Nugent<-sum(abs(accuracy\_table.Nugent$predicted- accuracy\_table.Nugent$actual))/nrow(accuracy\_table.Nugent))

## [1] 1.932824

## //////////  
### Intersect Model Results  
## //////////  
  
intersect(Nugent\_RF$top\_features$top\_features.all,Amsel\_RF$top\_features$top\_features.all)

## [1] "miR-193b" "miR-203b" "miR-324" "miR-182" "miR-3607" "miR-378a"  
## [7] "miR-146a" "miR-500a"

print(paste0("The Nugent-RF model correctly predicted the Nugent score within ",round(mean\_absolute\_error.Nugent,digits = 0)," values on average. There were ",length(Nugent\_RF$top\_features$top\_features.all)," significant miRNAs using Nugent-RF."))

## [1] "The Nugent-RF model correctly predicted the Nugent score within 2 values on average. There were 30 significant miRNAs using Nugent-RF."

## //////////  
### Output Model Results  
## //////////  
  
### Tables  
## Write model input table to file, keeping only variables that were used in either Nugent-RF or Amsel-RF  
model\_input.out<-model\_input[,names(model\_input) %in% unique(c(names(model\_input\_Amsel),names(model\_input\_Nugent)))]  
  
predictor\_names<-names(dplyr::select(model\_input.out,-c(AMSEL\_prediction,NUGENT\_SCORE,SID)))  
predictor\_names.miRs<-sum(grepl(predictor\_names,pattern = "miR|let"))  
print(paste0("The Nugent-RF and proxy-Amsel-RF models used ",length(predictor\_names)," predictors including ",predictor\_names.miRs," non-zero log2 transformed miRNA read counts and ",length(predictor\_names)-predictor\_names.miRs," metadata variables as inputs to rank feature importance (",TABLE\_MODEL\_INPUT,")."))

## [1] "The Nugent-RF and proxy-Amsel-RF models used 178 predictors including 169 non-zero log2 transformed miRNA read counts and 9 metadata variables as inputs to rank feature importance (TABLE\_A5.csv)."

## Record which variables were used in training and testing  
model\_input.out[row.names(model\_input.out) %in% Nugent\_RF$training\_ids,"Nugent\_RF\_trainingTestingSet"]<-"training"  
model\_input.out[row.names(model\_input.out) %in% Nugent\_RF$testing\_ids,"Nugent\_RF\_trainingTestingSet"]<-"testing"  
  
model\_input.out[row.names(model\_input.out) %in% Amsel\_RF$training\_ids,"Amsel\_RF\_trainingTestingSet"]<-"training"  
model\_input.out[row.names(model\_input.out) %in% Amsel\_RF$testing\_ids,"Amsel\_RF\_trainingTestingSet"]<-"testing"  
  
write.csv(model\_input.out,paste0(thesis\_tables\_directory,TABLE\_MODEL\_INPUT),row.names=T,quote=F)  
  
## Combine both RF models' importance information as a single table  
TABLE\_RF\_SUMMARY\_output<-data.frame(features=names(model\_input %>% dplyr::select(-c(AMSEL\_prediction,NUGENT\_SCORE,SID))))  
TABLE\_RF\_SUMMARY\_output.tmp1<-data.frame(features=row.names(Amsel\_RF$rfp$importance\_w\_pval),AMSEL=Amsel\_RF$rfp$importance\_w\_pval)  
TABLE\_RF\_SUMMARY\_output.tmp2<-data.frame(features=row.names(Nugent\_RF$rfp$importance\_w\_pval),NUGENT=Nugent\_RF$rfp$importance\_w\_pval)  
TABLE\_RF\_SUMMARY\_output<-merge(merge(TABLE\_RF\_SUMMARY\_output.tmp1,TABLE\_RF\_SUMMARY\_output.tmp2,all = T),TABLE\_RF\_SUMMARY\_output,all = T)  
TABLE\_RF\_SUMMARY\_output<-dplyr::select(TABLE\_RF\_SUMMARY\_output,-contains("BV"))  
write.csv(TABLE\_RF\_SUMMARY\_output,file=paste0(thesis\_tables\_directory,TABLE\_RF\_SUMMARY),row.names=F,quote=F)  
write.csv(dplyr::select(Clinical\_Visit\_RF$importance,-contains("BV")),file=paste0(thesis\_tables\_directory,TABLE\_RF\_SUMMARY.CV),row.names=T,quote=F)  
  
##Store the top miRNAs for later use  
top\_mirs\_table<-data.frame(miRNA=union(Nugent\_RF$top\_features$top\_features.all,Amsel\_RF$top\_features$top\_features.all),RF\_Group=0)  
top\_mirs\_table[top\_mirs\_table$miRNA %in% Nugent\_RF$top\_features$top\_features.all,"RF\_Group"]<-"Nugent-RF"  
top\_mirs\_table[top\_mirs\_table$miRNA %in% Amsel\_RF$top\_features$top\_features.all,"RF\_Group"]<-"Amsel-RF"  
top\_mirs\_table[top\_mirs\_table$miRNA %in% Amsel\_RF$top\_features$top\_features.all & top\_mirs\_table$miRNA %in% Nugent\_RF$top\_features$top\_features.all,"RF\_Group"]<-"Both"  
  
top\_mirs\_table[order(top\_mirs\_table$RF\_Group),]

## miRNA RF\_Group  
## 31 let-7e Amsel-RF  
## 32 miR-184 Amsel-RF  
## 33 let-7a-2 Amsel-RF  
## 34 miR-7-3 Amsel-RF  
## 35 miR-183 Amsel-RF  
## 36 miR-100 Amsel-RF  
## 37 miR-320a Amsel-RF  
## 38 miR-10b Amsel-RF  
## 39 miR-362 Amsel-RF  
## 40 miR-16-1 Amsel-RF  
## 41 miR-4510 Amsel-RF  
## 42 miR-342 Amsel-RF  
## 1 miR-193b Both  
## 2 miR-203b Both  
## 3 miR-324 Both  
## 6 miR-182 Both  
## 8 miR-3607 Both  
## 15 miR-378a Both  
## 18 miR-146a Both  
## 30 miR-500a Both  
## 4 miR-130a Nugent-RF  
## 5 miR-224 Nugent-RF  
## 7 miR-149 Nugent-RF  
## 9 miR-375 Nugent-RF  
## 10 miR-15a Nugent-RF  
## 11 miR-3653 Nugent-RF  
## 12 miR-21 Nugent-RF  
## 13 miR-223 Nugent-RF  
## 14 miR-15b Nugent-RF  
## 16 miR-203a Nugent-RF  
## 17 miR-200b Nugent-RF  
## 19 miR-205 Nugent-RF  
## 20 miR-16-2 Nugent-RF  
## 21 miR-152 Nugent-RF  
## 22 miR-199b Nugent-RF  
## 23 miR-20a Nugent-RF  
## 24 miR-95 Nugent-RF  
## 25 miR-197 Nugent-RF  
## 26 miR-365a Nugent-RF  
## 27 miR-101-2 Nugent-RF  
## 28 miR-191 Nugent-RF  
## 29 miR-140 Nugent-RF

## Text  
print(paste0("A total of ",length(top\_mirs\_table[top\_mirs\_table$RF\_Group=="Both","miRNA"])," miRNAs were common to both proxy-Amsel-BV and Nugent-BV Random Forest models: ", str\_c(top\_mirs\_table[top\_mirs\_table$RF\_Group=="Both","miRNA"],collapse=", "),"."))

## [1] "A total of 8 miRNAs were common to both proxy-Amsel-BV and Nugent-BV Random Forest models: miR-193b, miR-203b, miR-324, miR-182, miR-3607, miR-378a, miR-146a, miR-500a."

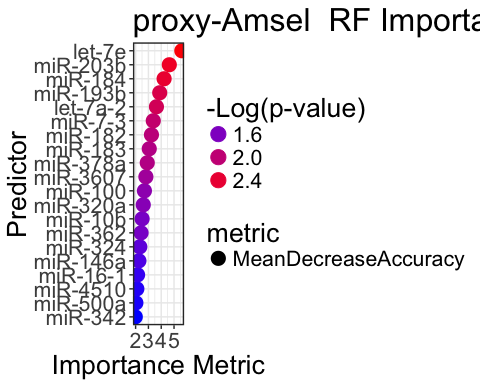
print(paste0("There were ",nrow(top\_mirs\_table[top\_mirs\_table$RF\_Group=="Amsel-RF",])," and ",nrow(top\_mirs\_table[top\_mirs\_table$RF\_Group=="Nugent-RF",])," statistically significant miRNAs unique to each ofproxy-Amsel classification (proxy-Amsel-RF) and Nugent score regression (Nugent-RF), respectively"))

## [1] "There were 12 and 22 statistically significant miRNAs unique to each ofproxy-Amsel classification (proxy-Amsel-RF) and Nugent score regression (Nugent-RF), respectively"

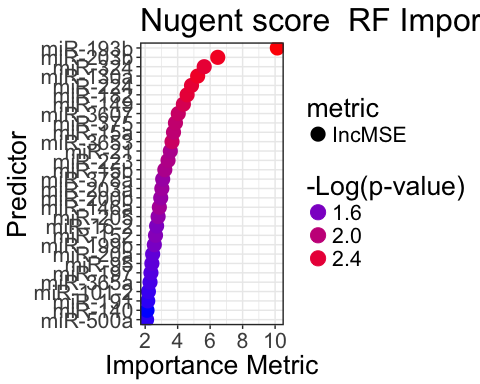
### Figures  
dplyr::select(data.frame(Amsel\_RF$rfp$importance\_w\_pval[row.names(Amsel\_RF$rfp$importance\_w\_pval) %in% top\_mirs\_table[top\_mirs\_table$RF\_Group %in% c("Amsel-RF","Both"),"miRNA"],]),contains("Mean"))

## MeanDecreaseAccuracy MeanDecreaseAccuracy.pval MeanDecreaseGini  
## let-7e 5.607872 0.002195609 0.9470099  
## miR-203b 4.630648 0.003193613 0.7717181  
## miR-184 4.224487 0.003792415 0.5855550  
## miR-193b 3.882583 0.005588822 0.4847252  
## let-7a-2 3.626207 0.006786427 0.3909026  
## miR-7-3 3.378600 0.009181637 0.4143229  
## miR-182 3.235782 0.010778443 0.4662138  
## miR-183 3.065067 0.012375250 0.4495199  
## miR-378a 2.909091 0.012375250 0.3701688  
## miR-3607 2.801747 0.016367265 0.3678346  
## miR-100 2.687550 0.020958084 0.3507171  
## miR-320a 2.597080 0.022754491 0.2273535  
## miR-10b 2.506643 0.026746507 0.3665795  
## miR-362 2.426444 0.026746507 0.2311134  
## miR-324 2.327208 0.034131737 0.2919702  
## miR-146a 2.257027 0.038123752 0.3095903  
## miR-16-1 2.154749 0.043313373 0.2736763  
## miR-4510 2.096317 0.046706587 0.2717785  
## miR-500a 2.020440 0.048103792 0.2481224  
## miR-342 1.973390 0.049900200 0.1733185  
## MeanDecreaseGini.pval  
## let-7e 0.002195609  
## miR-203b 0.002994012  
## miR-184 0.006986028  
## miR-193b 0.020558882  
## let-7a-2 0.057285429  
## miR-7-3 0.041916168  
## miR-182 0.022355289  
## miR-183 0.069261477  
## miR-378a 0.047704591  
## miR-3607 0.102395210  
## miR-100 0.135728543  
## miR-320a 0.257285429  
## miR-10b 0.103393214  
## miR-362 0.250299401  
## miR-324 0.258283433  
## miR-146a 0.185828343  
## miR-16-1 0.150698603  
## miR-4510 0.167664671  
## miR-500a 0.202994012  
## miR-342 0.415369261

Amsel\_RF.plot<-plot\_importance(dplyr::select(data.frame(Amsel\_RF$rfp$importance\_w\_pval[row.names(Amsel\_RF$rfp$importance\_w\_pval) %in% top\_mirs\_table[top\_mirs\_table$RF\_Group %in% c("Amsel-RF","Both"),"miRNA"],]),contains("Accuracy")),ntopfeats = 20,rankBy = "MeanDecreaseAccuracy",nfeats = 20,model\_name = "proxy-Amsel",size\_font = 20)  
  
Nugent\_RF.plot<-plot\_importance(dplyr::select(data.frame(Nugent\_RF$rfp$importance\_w\_pval[row.names(Nugent\_RF$rfp$importance\_w\_pval) %in% top\_mirs\_table[top\_mirs\_table$RF\_Group %in% c("Nugent-RF","Both"),"miRNA"],]),contains("MSE")),ntopfeats = 30,rankBy = "IncMSE",nfeats = 30,model\_name = "Nugent score",size\_font = 20)  
  
plot(Amsel\_RF.plot)



plot(Nugent\_RF.plot)



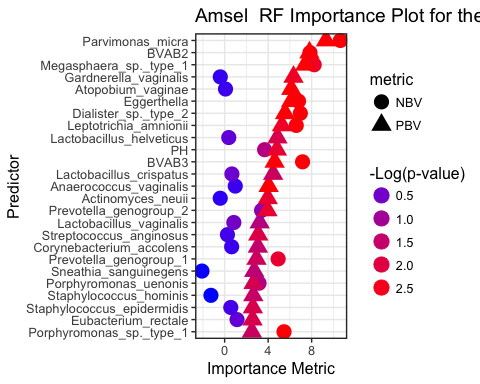
cairo\_ps(file=paste0(thesis\_figures\_directory,FIGURE\_RF\_IMPORTANCE\_AMSEL), width=11, height=8.5)  
plot(Amsel\_RF.plot)  
dev.off()

## quartz\_off\_screen   
## 2

cairo\_ps(file=paste0(thesis\_figures\_directory,FIGURE\_RF\_IMPORTANCE\_NUGENT), width=11, height=8.5)  
plot(Nugent\_RF.plot)  
dev.off()

## quartz\_off\_screen   
## 2

plot\_importance(dplyr::select(Clinical\_Visit\_RF$importance,contains("BV")),ntopfeats = 20,rankBy = "PBV",model\_name = "Amsel")



## Plot Expression of each RF mir as a function of Nugent score & proxy-Amsel prediction  
  
topmir\_plot\_data<-model\_input[!is.na(model\_input$NUGENT\_SCORE),names(model\_input) %in% c("NUGENT\_SCORE","AMSEL\_prediction","CST",as.character(top\_mirs\_table$miRNA[top\_mirs\_table$RF\_Group %in% "Both"]))]  
topmir\_plot\_data$CST<-as.character(topmir\_plot\_data$CST)  
  
topmir\_plot\_data<-melt(topmir\_plot\_data,id.vars = c("NUGENT\_SCORE","AMSEL\_prediction","CST"))  
  
topmir\_plot\_data$variable<-gsub(topmir\_plot\_data$variable,pattern = "hsa.mir.",replacement = "miR-")  
  
## Calcualte linear model fit for miRNA expression data vs Nugent score to rank miR expression plots in figure  
  
fitness<-apply(model\_input[,names(model\_input) %in% unique(topmir\_plot\_data$variable)],MARGIN = 2,function(s) summary(lm(model\_input$NUGENT\_SCORE~s)))  
  
fitness.r2<-data.frame(adjr2=unlist(lapply(fitness,function(x) round(x$adj.r.squared,digits = 3))))  
fitness.r2$variable<-row.names(fitness.r2)  
  
print(fitness.r2[order(abs(fitness.r2$adjr2),decreasing = T),])

## adjr2 variable  
## miR-193b 0.379 miR-193b  
## miR-182 0.214 miR-182  
## miR-203b 0.182 miR-203b  
## miR-378a 0.140 miR-378a  
## miR-3607 0.134 miR-3607  
## miR-324 0.124 miR-324  
## miR-500a 0.075 miR-500a  
## miR-146a 0.001 miR-146a

topmir\_plot\_data<-join(topmir\_plot\_data,fitness.r2)

## Joining by: variable

topmir\_plot\_data$variable<-factor(x=topmir\_plot\_data$variable,levels = unique(topmir\_plot\_data[order(abs(topmir\_plot\_data$adjr2),decreasing = T),c("variable")]),ordered = T)  
  
## Liner correlation coeff for miRNA expression data:  
paste(apply(unique(dplyr::select(topmir\_plot\_data,c(variable,adjr2))),MARGIN = 1,FUN = function(x) paste(sep =":" ,x[1],x[2])),sep=",")

## [1] "miR-146a:0.001" "miR-182:0.214" "miR-193b:0.379" "miR-203b:0.182"  
## [5] "miR-324:0.124" "miR-3607:0.134" "miR-378a:0.140" "miR-500a:0.075"

## Collapse CSTs to a single CST super type  
topmir\_plot\_data[topmir\_plot\_data$CST %in% c("III-A","III-B"),"CST"]<-"III"  
topmir\_plot\_data[topmir\_plot\_data$CST %in% c("I-A","I-B"),"CST"]<-"I"  
  
## Plot the miRNA expression figure  
PLOT\_EXPRESSION<-  
 ggplot(topmir\_plot\_data,aes(x=NUGENT\_SCORE,y=value))+  
 stat\_smooth(method = "lm",col='#FF5733',se=F)+ #  
 geom\_point(aes(col=as.factor(CST),pch=AMSEL\_prediction),size=3,show.legend = T)+  
 facet\_wrap(~variable,scales = "free\_y",nrow=2)+  
 ylab("Normalized Expression")+xlab("Nugent Score")+  
 mBio+  
 theme(text = element\_text(size=20),axis.text.x = element\_text(size=15))+  
 scale\_x\_continuous(breaks=0:10)+  
 scale\_color\_manual(values = cst.colors,guide = guide\_legend(title = "CST"))+  
 guides(pch=guide\_legend(title="proxy Amsel Prediction"))  
  
cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_TOPMIRS),width = 16,height = 8) #  
plot(PLOT\_EXPRESSION)  
dev.off()

## quartz\_off\_screen   
## 2

library(Boruta)

## Loading required package: ranger

##   
## Attaching package: 'ranger'

## The following object is masked from 'package:randomForest':  
##   
## importance

postscript(paste0(R\_script\_output\_directory,"Boruta\_optimization.ps"))  
Nugent\_RF.Boruta<-Boruta(x=dplyr::select(model\_input\_Nugent,-c(SID,NUGENT\_SCORE)),y=model\_input\_Nugent$NUGENT\_SCORE)  
plot(Nugent\_RF.Boruta,main="Nugent-RF Boruta unimportant feature removal")  
Nugent\_RF.Boruta$finalDecision[Nugent\_RF.Boruta$finalDecision=="Confirmed"]

## miR-101-1 miR-101-2 miR-128-1 miR-128-2 miR-130a miR-142   
## Confirmed Confirmed Confirmed Confirmed Confirmed Confirmed   
## miR-143 miR-149 miR-15a miR-182 miR-193b miR-194-2   
## Confirmed Confirmed Confirmed Confirmed Confirmed Confirmed   
## miR-199a-1 miR-199b miR-203a miR-203b miR-205 miR-223   
## Confirmed Confirmed Confirmed Confirmed Confirmed Confirmed   
## miR-320b-1 miR-324 miR-3607 miR-365a miR-378a miR-486   
## Confirmed Confirmed Confirmed Confirmed Confirmed Confirmed   
## Levels: Tentative Confirmed Rejected

Amsel\_RF.Boruta<-Boruta(x=dplyr::select(model\_input\_Amsel,-c(SID,AMSEL\_prediction)),y=as.factor(model\_input\_Amsel$AMSEL\_prediction))  
plot(Amsel\_RF.Boruta,main="proxy-Amsel-RF Boruta unimportant feature removal")  
Amsel\_RF.Boruta$finalDecision[Amsel\_RF.Boruta$finalDecision=="Confirmed"]

## let-7a-2 miR-100 miR-146a miR-182 miR-183 miR-184 miR-193b   
## Confirmed Confirmed Confirmed Confirmed Confirmed Confirmed Confirmed   
## miR-203b miR-223 miR-320a miR-3607 miR-486   
## Confirmed Confirmed Confirmed Confirmed Confirmed   
## Levels: Tentative Confirmed Rejected

intersect(names(Nugent\_RF.Boruta$finalDecision[Nugent\_RF.Boruta$finalDecision=="Confirmed"]),names(Amsel\_RF.Boruta$finalDecision[Amsel\_RF.Boruta$finalDecision=="Confirmed"]))

## [1] "miR-182" "miR-193b" "miR-203b" "miR-223" "miR-3607" "miR-486"

dev.off()

## quartz\_off\_screen   
## 2

## Map miRNAs to Gene Targets and GO Process (miR-GO-Target) (Table 2)

Using experimentally validated miRNA targets, map to direct GO processes to discover common & unique functions of top miRNAs. 1. Output gene target list for each top miRNA.  
2. Count the frequency of each target for each GO process & miRNA. Plot.

### miRNA <-> Taget (gene) <-> Gene Ontology Process (direct)  
### miRNA <- - - - - - - -- > Gene Ontology Process (direct)  
  
## Read in miRTarBase Experimentally Validated Target List  
mir\_targets<-read.table(paste0(R\_script\_input\_directory,"miRTarBase\_SE\_WR\_homosapiens.txt"),header = T,sep="\t",stringsAsFactors = F) %>% filter(Species..miRNA.=="Homo sapiens")  
nrow(mir\_targets)

## [1] 7310

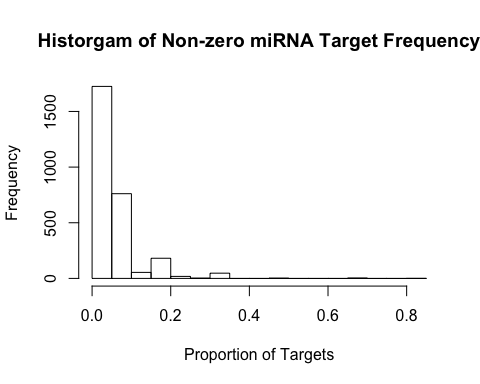
## Format targets list,pattern = "hsa.",replacement = "")  
mir\_targets$miRNA.format<-gsub(gsub(gsub(mir\_targets$miRNA,pattern = "mir.",replacement = "miR-"),replacement ="-",pattern = "\\."),pattern = "-.p",replacement = "-")  
  
## Map miRNA names to target List  
for(mir in as.character(top\_mirs\_table$miRNA)){  
   
 target\_list<-mir\_targets[grep(mir\_targets$miRNA.format,pattern = paste0("hsa-",mir,"-"),ignore.case = T,perl = T),c("miRNA","Target.Gene","References..PMID.")]  
 #if(!nrow(target\_list)==0){target\_list$source<-"canonical"}  
 if(!nrow(target\_list)==0 ){  
 top\_mirs\_table[top\_mirs\_table$miRNA==mir,"targets"]<-str\_c(sort(unique(target\_list$Target.Gene)),collapse = ",")  
 top\_mirs\_table[top\_mirs\_table$miRNA==mir,"sanityCheck"]<-str\_c(sort(unique(target\_list$miRNA)),collapse = ",")  
 top\_mirs\_table[top\_mirs\_table$miRNA==mir,"PMID"]<-str\_c(sort(unique(target\_list$References..PMID.)),collapse = ",")  
 }  
}  
  
write.table(top\_mirs\_table,file=paste0(thesis\_tables\_directory,TABLE\_MIR\_TARGETS),sep="\t",quote = F,row.names = F) ## Needs to be tab delimited due to "," separating gene targets.   
  
### Read in miR-Go-Gene mapping table mapping targets/miRNAs to GO processes. Gene targets mapped to miRNAs using a Python script.   
mir\_go<-read.csv(paste0(R\_script\_input\_directory,"miR\_GODirect\_Gene\_Map.csv"),stringsAsFactors = F)   
  
  
## Filter just the miR-GO map to 'top' miRNAs discovered above belonging to both RF model results   
mir\_go<-mir\_go[mir\_go$mir %in% as.character(filter(top\_mirs\_table,RF\_Group=="Both")$miRNA),]  
ddply(unique(dplyr::select(mir\_go,c(mir,gene))),c("mir"),summarise,n=length(gene))

## mir n  
## 1 miR-146a 48  
## 2 miR-182 46  
## 3 miR-193b 12  
## 4 miR-324 6  
## 5 miR-378a 16  
## 6 miR-500a 3

## Read in miR-target counts, created in outside script.   
mir\_targetCnt<-read.csv(paste0(R\_script\_input\_directory,"miR\_Target\_Counts.csv"),stringsAsFactors = F)  
mir\_targetCnt[mir\_targetCnt$mir %in% top\_mirs\_table[top\_mirs\_table$RF\_Group=="Both","miRNA"],]

## mir number\_targets  
## 9 miR-378a 16  
## 13 miR-193b 13  
## 19 miR-203b 1  
## 21 miR-3607 1  
## 22 miR-324 6  
## 27 miR-182 48  
## 33 miR-146a 51  
## 41 miR-500a 3

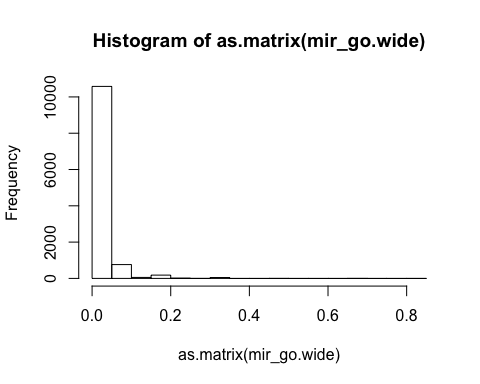
## Determine the number of genes associated with each miRNA- need to determine how many GO processes to plot in figure  
mir\_go.counts<-data.frame(table(dplyr::select(mir\_go,c(mir,GO))))  
mir\_go.miR193b<-mir\_go[mir\_go$mir=="miR-193b",]  
  
mir\_go.counts<-merge(mir\_go.counts,mir\_targetCnt)  
mir\_go.counts$Freq\_Targets<-mir\_go.counts$Freq/mir\_go.counts$number\_targets  
  
hist(mir\_go.counts$Freq\_Targets[!mir\_go.counts$Freq\_Targets==0],main="Historgam of Non-zero miRNA Target Frequency",xlab="Proportion of Targets")



## Somewhere above 0.2 might capture the interesting/common GO processes across miRNAS.   
  
## Make miR-GO counts into wide format (i.e., each GO process is a column, miRNAs are rows, values are proportion of targets in that GO/miRNA combination)  
mir\_go.wide<-tidyr::spread(dplyr::select(mir\_go.counts,-c(Freq,number\_targets)),key = GO,value = Freq\_Targets)  
  
row.names(mir\_go.wide)<-mir\_go.wide$mir  
mir\_go.wide<-dplyr::select(mir\_go.wide,-mir)  
  
## Get a better idea of how to plot common GO processes by determining cutoffs for target proportions  
## Balanace between proportion of targets across all miRNAs and total # across targets  
  
## create dummy dataframe- i and j hold the tuning parameters  
df1<-data.frame(i=rep(seq(0,max(mir\_go.wide),max(mir\_go.wide)/10),each=10),j=rep(seq(0,1,.1),times=10))  
  
## Calcualte number of columns (nc) for i and j cutoffs  
for(a in 1:nrow(df1)){  
 i<-df1[a,"i"]  
 j<-df1[a,"j"]  
 ## number of columns w/ propotion >i and of those, number greater than j of total miRNAs  
 nc<-ncol(mir\_go.wide[,colSums(mir\_go.wide>i)>=j\*nrow(mir\_go.wide)])  
 if(is.null(nc)){nc<-0}  
 df1[a,"ncol"]<-nc  
}  
(optimal\_cutoffs<-df1[df1$ncol==max(df1[df1$ncol<=25 & df1$ncol>=5,"ncol"]),]) ## What are the optimal cutoffs for somewhere between 5 and 25 GO processes on the plot?

## i j ncol  
## 8 0 0.7 19  
## 9 0 0.8 19

## Take the max i/j pair as the cutoffs:  
i<-0  
j<-0.8  
## Note that 19 in the "nc" column is the number of GO processes that will be plotted  
  
## Therefore, the proportion of genes should be >0 across at least 0.8\*6 =4.8 miRNAs.   
hist(as.matrix(mir\_go.wide))



## Apply optimal cutoffs to miR-GO (wide) data  
mir\_go.wide.filtered<-mir\_go.wide[,colSums(mir\_go.wide>i)>=j\*nrow(mir\_go.wide)]  
  
mir\_go.wide.mir193b<-mir\_go.wide[row.names(mir\_go.wide)=="miR-193b",]  
  
## The names of the GO processes to be used in plot:  
names(mir\_go.wide.filtered[,order(colSums(mir\_go.wide.filtered>0),decreasing = T)])

## [1] "GO:0000122~negative regulation of transcription from RNA polymerase II promoter"   
## [2] "GO:0001666~response to hypoxia"   
## [3] "GO:0001701~in utero embryonic development"   
## [4] "GO:0001934~positive regulation of protein phosphorylation"   
## [5] "GO:0006351~transcription<COMMA> DNA-templated"   
## [6] "GO:0006355~regulation of transcription<COMMA> DNA-templated"   
## [7] "GO:0006366~transcription from RNA polymerase II promoter"   
## [8] "GO:0006955~immune response"   
## [9] "GO:0007049~cell cycle"   
## [10] "GO:0007165~signal transduction"   
## [11] "GO:0007275~multicellular organism development"   
## [12] "GO:0008284~positive regulation of cell proliferation"   
## [13] "GO:0010628~positive regulation of gene expression"   
## [14] "GO:0030335~positive regulation of cell migration"   
## [15] "GO:0043066~negative regulation of apoptotic process"   
## [16] "GO:0043547~positive regulation of GTPase activity"   
## [17] "GO:0045893~positive regulation of transcription<COMMA> DNA-templated"   
## [18] "GO:0045944~positive regulation of transcription from RNA polymerase II promoter"   
## [19] "GO:0051091~positive regulation of sequence-specific DNA binding transcription factor activity"

mir\_go.wide.filtered$mir<-row.names(mir\_go.wide.filtered)  
mir\_go.wide.filtered.m<-melt(mir\_go.wide.filtered,id.vars = c("mir"))  
mir\_go.wide.filtered.m$GO\_only<-gsub(mir\_go.wide.filtered.m$variable,pattern = "~.\*",replacement = "") ## Store the GO number only in case needed  
  
## For this plot, summarize the mean gene target proprotion and # of miRNAs for each GO process. Use this to rank GO processes in plot  
(mir\_go.wide.filtered.summary<-ddply(mir\_go.wide.filtered.m,c("variable"),summarise,s=sum(value),n=sum(value>0)))

## variable  
## 1 GO:0000122~negative regulation of transcription from RNA polymerase II promoter  
## 2 GO:0001666~response to hypoxia  
## 3 GO:0001701~in utero embryonic development  
## 4 GO:0001934~positive regulation of protein phosphorylation  
## 5 GO:0006351~transcription<COMMA> DNA-templated  
## 6 GO:0006355~regulation of transcription<COMMA> DNA-templated  
## 7 GO:0006366~transcription from RNA polymerase II promoter  
## 8 GO:0006955~immune response  
## 9 GO:0007049~cell cycle  
## 10 GO:0007165~signal transduction  
## 11 GO:0007275~multicellular organism development  
## 12 GO:0008284~positive regulation of cell proliferation  
## 13 GO:0010628~positive regulation of gene expression  
## 14 GO:0030335~positive regulation of cell migration  
## 15 GO:0043066~negative regulation of apoptotic process  
## 16 GO:0043547~positive regulation of GTPase activity  
## 17 GO:0045893~positive regulation of transcription<COMMA> DNA-templated  
## 18 GO:0045944~positive regulation of transcription from RNA polymerase II promoter  
## 19 GO:0051091~positive regulation of sequence-specific DNA binding transcription factor activity  
## s n  
## 1 1.5555241 6  
## 2 0.7194570 5  
## 3 0.6161388 5  
## 4 0.5043363 5  
## 5 0.8864065 5  
## 6 0.7283183 5  
## 7 0.7479261 5  
## 8 0.7077677 5  
## 9 0.6761878 5  
## 10 0.9829374 5  
## 11 0.5131976 5  
## 12 0.9694570 5  
## 13 0.6918363 5  
## 14 0.6281109 5  
## 15 1.0101810 5  
## 16 0.4457956 5  
## 17 1.3226810 5  
## 18 1.4440045 5  
## 19 0.3881976 5

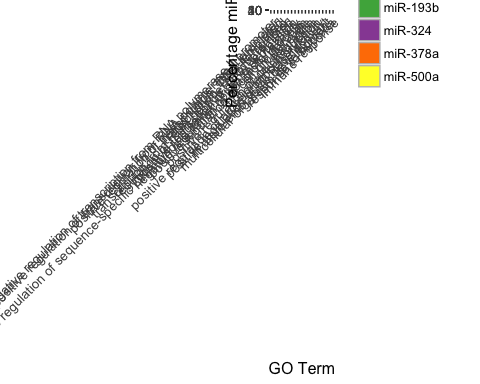
## TO make plot simplier to read, group GO terms in logical super groupings. This is done using a map between GO group and GO term, created outside of script  
group\_go\_map<-read.csv(file=paste0(R\_script\_input\_directory,"GO\_Process\_Grouping\_Map.csv"))  
  
## How many GO terms in each super group  
(group\_freq<-data.frame(table(group\_go\_map$Group)))

## Var1 Freq  
## 1 Cell cycle 4  
## 2 Development 2  
## 3 Hypoxia 1  
## 4 Immune 1  
## 5 Signaling 3  
## 6 Transcription 8

names(group\_freq)<-c("Group","n\_group")  
# MErge the super groups and data found above  
group\_go\_map<-merge(group\_go\_map,group\_freq)  
  
## Order by grouping, then by mean target proportion within each group  
group\_go\_map<-group\_go\_map[order(group\_go\_map$n\_group, group\_go\_map$s,decreasing = T),]  
  
#FOrmat the GO term for plot  
mir\_go.wide.filtered.m$GO\_format<-factor(mir\_go.wide.filtered.m$variable,levels =group\_go\_map$variable ,ordered = T,labels = group\_go\_map$Go\_format)  
  
  
PLOT\_MIR\_TARGET\_GO\_PROP<-ggplot(mir\_go.wide.filtered.m,aes(x=GO\_format,y=100\*value,fill=mir))+geom\_bar(width=.7,position = "dodge",col="grey",stat="identity")+  
 mBio+  
 theme(axis.text.x = element\_text(angle=45,hjust = 1,size=10),text=element\_text(size=12),plot.margin = unit(c(10,25,10,170),units = "pt"))+  
 scale\_fill\_brewer(type = "qual",palette = "Set1")+  
 xlab("GO Term")+  
 ylab("Percentage miRNA Targets")  
  
cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_MIR\_TARGETS\_GO),width = 11,height = 8.5)  
plot(PLOT\_MIR\_TARGET\_GO\_PROP)  
dev.off()

## quartz\_off\_screen   
## 2

plot(PLOT\_MIR\_TARGET\_GO\_PROP)



## NOTE: the # of miR targets table is added later. See above for table.   
  
## Output miR-Target-GO table for miR-193b only  
mir\_go.wide.filtered.m.193b<-filter(mir\_go.wide.filtered.m,mir=="miR-193b")  
  
write.table(dplyr::select(mir\_go.wide.filtered.m.193b,c(mir,GO\_format,value)),file = paste0(thesis\_tables\_directory,TABLE\_TOPMIRS),sep="\t",quote=F,row.names = F)  
  
## Make a GO process bar chart for miR-193b only (for thesis presentation, not used in thesis document)  
mir193b\_GOcount\_groups<-data.frame(table(dplyr::select(unique(dplyr::select(merge(mir\_go.miR193b,group\_go\_map,by.x="GO",by.y="variable"),c(Group,gene))),Group)))  
mir193b\_GOcount\_groups$Var1<-factor(mir193b\_GOcount\_groups$Var1,levels=mir193b\_GOcount\_groups[order(mir193b\_GOcount\_groups$Freq,decreasing = T),"Var1"],ordered = T)  
  
mir193b\_GOcount\_groups.plot<-ggplot(mir193b\_GOcount\_groups,aes(x=Var1,y=Freq))+geom\_bar(width=.7,position = "dodge",col="grey",stat="identity")+  
 mBio+  
 theme(axis.text.x = element\_text(angle=45,hjust = 1),text=element\_text(size=25),plot.margin = unit(c(10,25,10,170),units = "pt"))+  
 scale\_fill\_brewer(type = "qual",palette = "Set1")+  
 scale\_y\_continuous(breaks=c(2,4,6,8))+  
 xlab("GO process")+  
 ylab("Number miR-193b targets")  
  
cairo\_ps(paste0(R\_script\_output\_directory,"miR193b\_GOProcess\_presentation.eps"),width = 11,height = 8.5)  
plot(mir193b\_GOcount\_groups.plot)  
dev.off()

## quartz\_off\_screen   
## 2

# Function of miR-193b and in vitro Experimentation, Implications

Perform further experimentation on miR-193b, one of the top miRNAs found in the discovery phase above.

## Validate SmallRNASeq Results using qPCR

::select 5 swabs each from NBV and PBV subjects, then measure relative hsa-mir-193b expression

## Loads qPCR Data from 5 NBV and PBV subjects to Validate miRNA-Seq  
load(paste0(R\_script\_input\_directory,"miRNA\_qPCR\_Validation.RData"))  
miR\_qPCR\_results.deltaCt<-miR\_qPCR\_results.deltaCt[!is.nan(miR\_qPCR\_results.deltaCt$deltaCt),] ## remove NA from data  
  
## Create a mixed-effects linear model as a function of BVGroup of dCt values, using subject ID as the random effect  
seq\_validation.lme <- lme(deltaCt ~ BVGroup,  
 random = ~ 1|SID,  
 data = miR\_qPCR\_results.deltaCt)  
  
## Coefficient of PBV relative to NBV in model + p value  
effect\_PBV<-seq\_validation.lme$coefficients$fixed["BVGroupPBV"]  
dct.summary<-ddply(miR\_qPCR\_results.deltaCt,c("BVGroup"),summarise,mean=mean(deltaCt))  
2^(dct.summary$mean[2]-dct.summary$mean[1])

## [1] 3.019121

summary(seq\_validation.lme)

## Linear mixed-effects model fit by REML  
## Data: miR\_qPCR\_results.deltaCt   
## AIC BIC logLik  
## 19.19847 18.36551 -5.599237  
##   
## Random effects:  
## Formula: ~1 | SID  
## (Intercept) Residual  
## StdDev: 0.6267906 0.107014  
##   
## Fixed effects: deltaCt ~ BVGroup   
## Value Std.Error DF t-value p-value  
## (Intercept) -1.682609 0.3179302 5 -5.292385 0.0032  
## BVGroupPBV 1.432459 0.4849846 5 2.953617 0.0318  
## Correlation:   
## (Intr)  
## BVGroupPBV -0.656  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -0.62114365 -0.18105809 0.03265107 0.08253511 0.79729434   
##   
## Number of Observations: 8  
## Number of Groups: 7

## miR-193b qPCR Time-Course in VK2 monolayer after BCS Exposure

qPCR is measured with QIAGEN miScript II kit, BCSs are:  
\* L. crispatus  
\* L. jensenii  
\* L. iners  
\* G. vaginalis  
\* Media control

20% BCSs exposed to VK2 monolayer epithelial cells for:  
\* 0.5 hours  
\* 1 hour  
\* 4 hours  
\* 11 hours  
\* 13 hours  
\* 22 hours

Primers:  
\* hsa-miR-193b-3p  
\* hsa-miR-RNU6 (normalization control)

# Calculate mean & s.d. Ct values for each BCS/time point.   
load(file=paste0(R\_script\_input\_directory,"qPCR\_time\_course.Rdata")) ## Ct values  
load(file=paste0(R\_script\_input\_directory,"qpcr\_sigtest.Rdata")) ## Container for holding sig testing comparisons/results  
  
qPCR\_time\_course.summary<-ddply(qPCR\_time\_course,c("Study","ExposureTime","BCS"),summarise,m\_193b=mean(miR193b\_Ct,na.rm = T),sd\_193b=sd(miR193b\_Ct,na.rm = T),  
 m\_R6=mean(RNU6\_Ct,na.rm = T),sd\_R6=sd(RNU6\_Ct,na.rm = T))  
  
## Calcualte delta Ct using RNU6 as endogenous control. Calcualte standard deviation using propagation of error  
qPCR\_time\_course.summary$deltaCt.RNU6<-qPCR\_time\_course.summary$m\_193b-qPCR\_time\_course.summary$m\_R6  
qPCR\_time\_course.summary$deltaCt.RNU6.sd<-sqrt(qPCR\_time\_course.summary$sd\_193b^2+qPCR\_time\_course.summary$sd\_R6^2)  
#   
  
## calcualte signifigance using delta Ct mean + s.d., which uses a modified t.test2 function.   
## Loop through the comparisons and test pair-wise  
for(sigtest in 1:nrow(qpcr\_sigtest)){  
   
 timei<-as.character(qpcr\_sigtest[sigtest,"ExposureTime"]) # TIme point  
 x1<-as.character(qpcr\_sigtest[sigtest,"comparison"]) ## Comparison sample  
 y1<-as.character(qpcr\_sigtest[sigtest,"reference"]) ## Reference sample  
   
 ## When comparing samples, make sure to use the correct study associated with BCS  
 study.x<-as.character(qpcr\_sigtest[sigtest,"Study.comp"])  
 study.y<-as.character(qpcr\_sigtest[sigtest,"Study.ref"])  
   
 # Standard deviations, as computed above   
 sx <- filter(qPCR\_time\_course.summary,BCS==x1 & ExposureTime==timei & Study==study.x)$deltaCt.RNU6.sd  
 sy<-filter(qPCR\_time\_course.summary,BCS==y1 & ExposureTime==timei & Study==study.y)$deltaCt.RNU6.sd  
   
 # deltaCt mean  
 mx<-filter(qPCR\_time\_course.summary,BCS==x1 & ExposureTime==timei & Study==study.x)$deltaCt.RNU6  
 my<-filter(qPCR\_time\_course.summary,BCS==y1 & ExposureTime==timei & Study==study.y)$deltaCt.RNU6  
   
 ## Compute t statistic/p value using mean, s.d. and sample size for two groups  
 tes<-t.test2(m1 = mx,  
 m2= my,  
 s1=sx,  
 s2=sy,  
 n1=3,  
 n2=3)   
   
 qpcr\_sigtest[sigtest,"pval"]<-tes["p-value"]  
 qpcr\_sigtest[sigtest,"mean\_diff"]<-tes["Difference of means"]  
   
 qpcr\_sigtest[sigtest,"se"]<-sqrt(sy^2+sx^2)  
   
 ## ddCt >0 implies x transcript is less abundant than y and ddCt<0 implies x is more abudnact than y. The relative magnatitude on a plot of 2^(-x) != 2(x) although the interpretability is the same (transcript is either 2^-x less abudnant than y or 2^x more abudnant than y). Therefore, take absolute ddCt before raising it to 2.   
 ##Note ddct is actually 2^-ddct  
   
 if(-qpcr\_sigtest[sigtest,"mean\_diff"]<0){  
   
 qpcr\_sigtest[sigtest,"ddct"]<-(-2^(tes["Difference of means"]))  
 qpcr\_sigtest[sigtest,"ddct\_se"]<-(-2^(tes["Difference of means"]+  
 qpcr\_sigtest[sigtest,"se"]))  
 qpcr\_sigtest[sigtest,"ddct\_se\_m"]<-(-2^(tes["Difference of means"]-  
 qpcr\_sigtest[sigtest,"se"]))  
 }else{  
   
 qpcr\_sigtest[sigtest,"ddct"]<-2^(-tes["Difference of means"])   
   
 qpcr\_sigtest[sigtest,"ddct\_se"]<-2^((-tes["Difference of means"])+  
 qpcr\_sigtest[sigtest,"se"])  
 qpcr\_sigtest[sigtest,"ddct\_se\_m"]<-2^((-tes["Difference of means"])-  
 qpcr\_sigtest[sigtest,"se"])  
   
 }  
}  
  
## Annotate sig tests  
qpcr\_sigtest[qpcr\_sigtest$pval<=pval\_threshold & !is.na(qpcr\_sigtest$pval),"sig"]<-"\*"  
  
##convert exposure time to numeric for plot  
qpcr\_sigtest$ExposureTime.n<-as.numeric(gsub(qpcr\_sigtest$ExposureTime,pattern = "hr",replacement = ""))  
  
qpcr\_sigtest$ref<-as.character(qpcr\_sigtest$reference)  
qpcr\_sigtest$comp<-as.character(qpcr\_sigtest$comparison)  
  
## Assign line types  
qpcr\_sigtest[qpcr\_sigtest$ref=="G. vaginalis","lt"]<-1#"dashed"  
qpcr\_sigtest[qpcr\_sigtest$ref=="Cell Culture Medium","lt"]<-3#"solid"  
cairo\_ps(paste0(R\_script\_output\_directory,"miR\_long\_plot\_presentation.eps"),width = 11.5,height = 6)  
## Plot qPCR Timecourse (put lactic acid comparison in sep. figure)  
PLOT\_QPCR\_TIMECOURSE<-  
 ggplot(filter(qpcr\_sigtest,reference=="G. vaginalis" & !grepl(qpcr\_sigtest$comparison,pattern = "\_10|\_766|0\_")), #& qpcr\_sigtest$reference== "G. vaginalis"),  
 aes(x=ExposureTime.n,y=-mean\_diff,col=comp))+  
 theme\_bw()+  
 geom\_hline(yintercept = 0,col="#9e9ac8",size=2.5)+  
 theme(plot.margin=unit(c(0,0,0,0),units = "pt"), text=element\_text(size=16))+  
 geom\_line(size=1.5,show.legend = T,aes(linetype=ref))+   
 geom\_errorbar(aes(ymax=-mean\_diff+se,ymin=-mean\_diff-se),size=1,show.legend = F)+  
 geom\_point(size=6,show.legend = F)+   
 ylab(expression(paste(-Delta,Delta,"Ct")))+  
 xlab("Exposure Time (hours)")+  
 scale\_color\_manual(name=expression(paste("Lactobacillus ",Delta,"Ct")),values = color\_scheme\_BCS)+  
 scale\_x\_continuous(breaks=c(0.5,1,4,11,13,22),minor\_breaks = NULL)+  
 scale\_linetype\_manual(name=expression(paste("Reference ",Delta,"Ct")),values=as.numeric(qpcr\_sigtest$lt))+  
 geom\_text(aes(label=sig),col='#252525',size=8,nudge\_y = -.05)  
  
  
cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_QPCR\_TIMECOURSE),width = 9,height = 6)  
PLOT\_QPCR\_TIMECOURSE  
dev.off()

## quartz\_off\_screen   
## 2

qpcr\_sigtest$comparison<-factor(qpcr\_sigtest$comparison,levels = c("L. crispatus", "L. jensenii","L. iners","G. vaginalis", "PH\_766","D\_10","L\_10","0\_06\_D","0\_06\_L"),ordered = T,labels = c("L. crispatus", "L. jensenii","L. iners","G. vaginalis", "1% lactic acid, pH 7.66","0.1% D-lactic acid","0.1% L-lactic acid","0.06% D-lactic acid","0.06% L-lactic acid"))  
  
qpcr\_sigtest$reference<-factor(qpcr\_sigtest$reference,levels = c("L. jensenii","L. iners","G. vaginalis", "Cell Culture Medium", "L\_10","0\_06\_D","0\_06\_L"),ordered = T,labels = c( "L. jensenii","L. iners","G. vaginalis", "Cell Culture Medium","0.1% L-lactic acid","0.06% D-lactic acid","0.06% L-lactic acid"))  
  
## Show D and L lactic acid qPCR results compared to other references  
ddct\_LacticAcid\_plot<-  
 ggplot(filter(qpcr\_sigtest,ExposureTime=="4hr" & !qpcr\_sigtest$reference=="0.06% D-lactic acid"),aes(x=reference,y=-mean\_diff,col=comparison))+  
 theme\_bw()+  
 geom\_hline(yintercept = 0,col="#9e9ac8",size=2.5)+  
 theme(plot.margin=unit(c(0,0,0,0),units = "pt"), text=element\_text(size=14))+  
 geom\_errorbar(aes(ymax=-mean\_diff+se,ymin=-mean\_diff-se),width=.2,show.legend = F,position = position\_dodge(width = .2))+  
 geom\_point(size=6,show.legend = T,position = position\_dodge(width = .2))+   
 ylab(expression(paste(-Delta,Delta,"Ct")))+  
 xlab(expression(paste("Reference ",Delta,"Ct")))+  
 scale\_color\_manual(name=expression(paste("Exposure ",Delta,"Ct")),values = c(color\_scheme\_BCS,"0.1% D-lactic acid"="#4dac26","0.1% L-lactic acid"="#d01c8b","1% lactic acid, pH 7.66"="#386cb0","0.06% L-lactic acid"="#f1b6da","0.06% D-lactic acid"="#b8e186"))+  
 geom\_text(aes(label=sig),col='#252525',size=8,nudge\_y = -.05)  
  
cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_DL\_LACTICACID\_QPCR),width = 9,height = 6)  
ddct\_LacticAcid\_plot  
dev.off()

## quartz\_off\_screen   
## 2

paste0("Additionally, the ∆∆Ct of miR-193b expression after 4 hours of exposure to 0.06% D-lactic acid relative to 0.06% L-lactic acid was found to be non-significant ( p=",qpcr\_sigtest[qpcr\_sigtest$comparison=="L-lactic acid" & qpcr\_sigtest$reference=="D-lactic acid","pval"],").")

## [1] "Additionally, the ∆∆Ct of miR-193b expression after 4 hours of exposure to 0.06% D-lactic acid relative to 0.06% L-lactic acid was found to be non-significant ( p=)."

## Write plot as a table  
qpcr\_sigtests<-dplyr::select(qpcr\_sigtest[with(qpcr\_sigtest,order(ExposureTime.n,comparison,reference)),],c(comparison,reference,ExposureTime,mean\_diff,pval,sig))  
qpcr\_sigtests$Figure<-gsub(FIGURE\_QPCR\_TIMECOURSE,pattern=".eps",replacement = "")  
  
write.csv(qpcr\_sigtests,file=paste0(thesis\_tables\_directory,TABLE\_QPCR\_TIMECOURSE),row.names = F)

## Quantify VK2 proliferation (function of miR-193b) exposed to BCS

* Expose VK2 cells to BCS
* Measure EdU detection and filled scratch area

#in\_vitro\_experiments<-read.csv(paste0("~/Dropbox (IGS)/Jacques\_Steve\_Shared/Manuscript/miRNA/Working/R\_script\_input/","In\_vitro\_Experiments.csv"))  
#save(in\_vitro\_experiments,file=paste0(R\_script\_input\_directory,"In\_Vitro\_Experiments.Rdata"))  
load(file=paste0(R\_script\_input\_directory,"In\_Vitro\_Experiments.Rdata"))  
  
## //////////////  
### Scratch Assay  
## //////////////  
  
## Melt scratch assay data for easier handling  
scratch.m<-filter(in\_vitro\_experiments,Experiment=="Scratch" & Observation == "Proliferation" & !grepl(x = as.character(in\_vitro\_experiments$Treatment),pattern = "CONTROL\_")) %>% dplyr::select(-c(Experiment,Coverslip)) #melt(proliferation)  
  
  
## Re-name the melted data  
names(scratch.m)<-c("Observation","BCS","percent\_cells","Field")  
  
#Factor BCS so that order is enforced  
scratch.m$BCS<-factor(scratch.m$BCS,ordered = T,levels = c("L. crispatus","L. jensenii","L. iners","G. vaginalis","Cell Culture Medium"))  
  
## Create a significance testing data frame to hold results  
setup\_sigtest.data<-setup\_sigtest(pval\_threshold = pval\_threshold,raw\_data = scratch.m,test\_function = "t.test",Experiment = "Scratch")

## [1] 1  
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6  
## [1] 7  
## [1] 8  
## [1] 9  
## [1] 10

scratch\_sigtest<-setup\_sigtest.data$sigtest  
scratch.summary<-setup\_sigtest.data$summary\_stats  
  
  
## Create Barplots for scratch assay  
  
plot\_scratch.prolif<-  
 ggplot(scratch.summary)+geom\_bar(aes(x=as.factor(BCS),y=mean,fill=BCS,col=BCS),stat="identity",show.legend = F)+  
 geom\_errorbar(aes(x=BCS,ymin=mean-sd,ymax=mean+sd),width=.3,show.legend = F)+  
 mBio+  
 ylab("Filled Scratch Area (%)")+  
 xlab("BCS")+  
 theme(plot.margin=unit(c(0,0,0,0),units = "pt"), text=element\_text(size=16),axis.text.x=element\_text(face="italic",angle = 45,vjust = 1,hjust = 1))+  
 scale\_fill\_manual(values=color\_scheme\_BCS)+   
 scale\_y\_continuous(limits=c(0,102))+  
 scale\_colour\_manual(values=color\_scheme\_BCS)+  
 annotate("text",x=scratch\_sigtest$midpoints,y=scratch\_sigtest$y.sig,label=scratch\_sigtest$sig,size=4)+  
 geom\_segment(data=setup\_sigtest.data$statbars,aes(x=x,xend=xend,y=y,yend=yend),size=1)+  
 geom\_segment(data=setup\_sigtest.data$statbars2,aes(x=x,xend=xend,y=y,yend=yend),size=1)  
  
##Write plot  
  
cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_SCRATCH\_QUANT),width = 8,height = 5.5)  
plot(plot\_scratch.prolif)  
dev.off()

## quartz\_off\_screen   
## 2

## //////////////  
### EdU Assay  
## //////////////  
  
## Melt EdU assay data for easier handling, clean up input  
EdU.m<-filter(in\_vitro\_experiments,Experiment=="Scratch" & Observation == "EdU") %>% dplyr::select(-c(Experiment)) #melt(scratch)  
EdU.m<-rename(EdU.m,BCS=Treatment)  
  
#Factor BCS so that order is enforced  
EdU.m$BCS<-factor(EdU.m$BCS,ordered = T,levels = c("L. crispatus","L. jensenii","L. iners","G. vaginalis","Cell Culture Medium"))  
  
## Create a significance testing data frame to hold results  
setup\_sigtest.data<-setup\_sigtest(pval\_threshold = pval\_threshold,raw\_data = EdU.m,test\_function = "t.test",Experiment = "Scratch")

## [1] 1  
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6  
## [1] 7  
## [1] 8  
## [1] 9  
## [1] 10

statbars<-setup\_sigtest.data$statbars  
statbars2<-setup\_sigtest.data$statbars2  
EdU\_sigtest<-setup\_sigtest.data$sigtest  
EdU.summary<-setup\_sigtest.data$summary\_stats  
  
## Create Barplots for EdU assay  
  
EdUPlot<-  
 ggplot(EdU.summary)+geom\_bar(aes(x=factor(BCS),y=mean,fill=BCS,col=BCS),stat="identity", position="dodge",show.legend = F)+  
 geom\_errorbar(aes(x=factor(BCS),ymin=mean-sd,ymax=mean+sd),position="dodge",show.legend = F,width=.3)+  
 mBio+  
 ylab("Epithelial Cells Positive for EdU (%)")+  
 xlab("BCS")+  
 theme(plot.margin=unit(c(0,0,0,0),units = "pt"), text=element\_text(size=16),axis.text.x=element\_text(face="italic",angle = 45,vjust = 1,hjust = 1))+  
 scale\_fill\_manual(values=color\_scheme\_BCS)+   
 scale\_y\_continuous(limits=c(0,102))+  
 scale\_colour\_manual(values=color\_scheme\_BCS)+  
 annotate("text",x=EdU\_sigtest$midpoints,y=EdU\_sigtest$y.sig,label=EdU\_sigtest$sig,size=4)+  
 geom\_segment(data=statbars,aes(x=x,xend=xend,y=y,yend=yend),size=1)+  
 geom\_segment(data=statbars2,aes(x=x,xend=xend,y=y,yend=yend),size=1)  
  
##Write plot  
  
cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_EDU\_QUANT),width = 8,height = 5.5)  
plot(EdUPlot)  
dev.off()

## quartz\_off\_screen   
## 2

## Scratch assay data for TSB and NYC media exposed cells   
control\_meds<-filter(in\_vitro\_experiments,grepl(x = as.character(in\_vitro\_experiments$Treatment),pattern = "CONTROL\_"))  
  
control\_meds.nyc.mean<-mean(control\_meds[control\_meds$Treatment=="CONTROL\_NYCmed\_20pct","percent\_cells"])  
control\_meds.tsb.mean<-mean(control\_meds[control\_meds$Treatment=="CONTROL\_TSBmed\_20pct","percent\_cells"])  
  
control\_meds.nyc.sd<-sd(control\_meds[control\_meds$Treatment=="CONTROL\_NYCmed\_20pct","percent\_cells"])  
control\_meds.tsb.sd<-sd(control\_meds[control\_meds$Treatment=="CONTROL\_TSBmed\_20pct","percent\_cells"])  
  
control\_meds.sig<-t.test(x=filter(control\_meds,Treatment=="CONTROL\_NYCmed\_20pct")[,"percent\_cells"],y =filter(control\_meds,Treatment=="CONTROL\_TSBmed\_20pct")[,"percent\_cells"],paired = F)  
  
  
paste0("Percent cell proliferation between 20% NYC-III and TSB bacterial culture media as evaluated by scratch assay was not significant (mean +/- standard deviation NYC-III cell proliferation=",control\_meds.nyc.mean,"% +/- ",control\_meds.nyc.sd,"%, mean +/- standard deviation TSB cell proliferation=,",control\_meds.tsb.mean,"% +/- ",control\_meds.tsb.sd,"%, p=",control\_meds.sig$p.value,")")

## [1] "Percent cell proliferation between 20% NYC-III and TSB bacterial culture media as evaluated by scratch assay was not significant (mean +/- standard deviation NYC-III cell proliferation=45.94945353125% +/- 9.11375283414706%, mean +/- standard deviation TSB cell proliferation=,50.704530715% +/- 14.5706994714208%, p=0.449364642797794)"

## Combine proliferation sig tests and write to file  
proliferation\_sigtests<-rbind(data.frame(Figure=gsub(FIGURE\_SCRATCH\_QUANT,replacement = "",pattern = ".eps"),scratch\_sigtest),data.frame(Figure=gsub(FIGURE\_EDU\_QUANT,replacement = "",pattern = ".eps"),EdU\_sigtest))  
proliferation\_sigtests<-proliferation\_sigtests[with(proliferation\_sigtests, order(Figure, x,xend)),]  
proliferation\_sigtests<-dplyr::select(proliferation\_sigtests,Figure,xref,reference,pval,mean\_diff,sig)  
  
proliferation\_sigtests<-rbind(proliferation\_sigtests,data.frame(Figure="FIGURE\_S5",xref="TSB",reference="NYC",pval=control\_meds.sig$p.value,mean\_diff=unname(control\_meds.sig$estimate[2]-control\_meds.sig$estimate[1]),sig=ifelse(control\_meds.sig$p.value<pval\_threshold,"\*","N.S.")))  
  
  
PLOT\_nyc\_v\_tsb<-ggplot(data.frame(media=c("NYC (Lactobacillus spp.)","TSB (G. vaginalis)"),scratch.mean=c(control\_meds.nyc.mean,control\_meds.tsb.mean),scratch.sd=c(control\_meds.nyc.sd,control\_meds.tsb.sd)))+  
 geom\_bar(aes(x=media,y=scratch.mean,fill=media),stat="identity",show.legend =F)+  
 geom\_errorbar(aes(x=media,ymax=scratch.mean+scratch.sd,ymin=scratch.mean-scratch.sd),width=.2)+  
 mBio+ylab("Percent Scratch Area Filled")+  
 xlab("VK2 Cell Culture Medium + 20% Bacterial Culture Medium")+  
 scale\_y\_continuous(limits=c(0,100))+  
 annotate("text",x=1.5,y=90,label="N.S.",size=4)+  
 geom\_segment(data=data.frame(x=1,xend=2,y=87,yend=87),aes(x=x,xend=xend,y=y,yend=yend),size=1)+  
 geom\_segment(data=data.frame(x=c(1,2),xend=c(1,2),y=c(85,85),yend=c(87,87)),aes(x=x,xend=xend,y=y,yend=yend),size=1)  
  
cairo\_ps(file=paste0(thesis\_figures\_directory,FIGURE\_nyc\_v\_tsb),width = 8,height = 5.5)  
PLOT\_nyc\_v\_tsb  
dev.off()

## quartz\_off\_screen   
## 2

write.csv(proliferation\_sigtests,file=paste0(thesis\_tables\_directory,TABLE\_EDU\_SCRATCH\_QUANT),row.names = F)  
  
## PErcent proliferation in 1% DL lactic acid, ph buffered 7.66 expoed cells  
DL\_PH766\_percent\_proliferation<-data.frame(DL\_1pct\_766=c(  
7.706708323,  
4.631917974,  
8.766697901,  
7.771365876)  
)  
mean(DL\_PH766\_percent\_proliferation$DL\_1pct\_766)

## [1] 7.219173

sd(DL\_PH766\_percent\_proliferation$DL\_1pct\_766)

## [1] 1.791771

t.test(x = scratch.m[scratch.m$BCS=="Cell Culture Medium","percent\_cells"],y = DL\_PH766\_percent\_proliferation$DL\_1pct\_766)

##   
## Welch Two Sample t-test  
##   
## data: scratch.m[scratch.m$BCS == "Cell Culture Medium", "percent\_cells"] and DL\_PH766\_percent\_proliferation$DL\_1pct\_766  
## t = 6.0479, df = 2.0264, p-value = 0.02546  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 19.92453 114.08363  
## sample estimates:  
## mean of x mean of y   
## 74.223251 7.219173

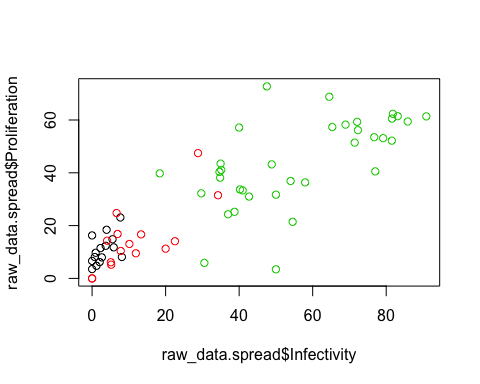
t.test(x = scratch.m[scratch.m$BCS=="G. vaginalis","percent\_cells"],y = DL\_PH766\_percent\_proliferation$DL\_1pct\_766)

##   
## Welch Two Sample t-test  
##   
## data: scratch.m[scratch.m$BCS == "G. vaginalis", "percent\_cells"] and DL\_PH766\_percent\_proliferation$DL\_1pct\_766  
## t = 6.0415, df = 2.2459, p-value = 0.01982  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 8.070304 37.070274  
## sample estimates:  
## mean of x mean of y   
## 29.789462 7.219173

## Inhbit A2EN cell proliferation & Observe CT Infectivity

* Expose A2EN cells to proliferation inhibitors
* Infect w/ CT
* Measure EdU detection and infection

load(file=paste0(R\_script\_input\_directory,"In\_Vitro\_Experiments.Rdata"))  
  
## Melt Ct infection assay data for easier handling  
pi.m<-filter(in\_vitro\_experiments,Experiment=="Infection") %>% dplyr::select(-Experiment)   
  
## Factor Treatment to enforce order and make labels  
pi.m$Treatment<-factor(pi.m$Treatment,levels = c("Cdk4 400nM","Fascaplysin 350nM","Cell Culture Medium"),ordered = T,labels = c("CAS 546102-60-7","Fascaplysin","Cell Culture Medium"))  
  
## Create a significance testing data frame to hold results  
setup\_sigtest.data<-setup\_sigtest(pval\_threshold = pval\_threshold,raw\_data = pi.m,test\_function = "t.test2",Experiment = "Infection")



## [1] 1  
## [1] "Proliferation"  
## Difference of means Std Error t   
## -6.6236985 3.8807666 -1.7068016   
## p-value   
## 0.1488428   
## [1] 2  
## [1] "Proliferation"  
## Difference of means Std Error t   
## -3.515431e+01 4.999389e+00 -7.031721e+00   
## p-value   
## 6.107794e-05   
## [1] 3  
## [1] "Proliferation"  
## Difference of means Std Error t   
## -2.853061e+01 5.386636e+00 -5.296555e+00   
## p-value   
## 3.539033e-04   
## [1] 4  
## [1] "Infectivity"  
## Difference of means Std Error t   
## -8.79632006 2.06013167 -4.26978536   
## p-value   
## 0.01525339   
## [1] 5  
## [1] "Infectivity"  
## Difference of means Std Error t   
## -5.338663e+01 7.813379e+00 -6.832720e+00   
## p-value   
## 3.346066e-04   
## [1] 6  
## [1] "Infectivity"  
## Difference of means Std Error t   
## -44.590306829 7.736338543 -5.763748132   
## p-value   
## 0.001000266

pi.summary<-setup\_sigtest.data$summary\_stats  
ct\_sigtest<-setup\_sigtest.data$sigtest  
statbars<-setup\_sigtest.data$statbars  
statbars2<-setup\_sigtest.data$statbars2  
  
## Split data into infection and proliferation (EdU) data  
ct\_sigtest\_inf<-filter(ct\_sigtest,Observation=="Infectivity")  
ct\_sigtest\_pro<-filter(ct\_sigtest,Observation=="Proliferation")  
  
## Plot Infectivity  
Fig3\_inf<-ggplot(filter(pi.summary,Observation=="Infectivity"),aes(x=Treatment,y=grand\_mean))+geom\_bar(aes(fill=Treatment),stat="identity",show.legend = F)+geom\_errorbar(aes(ymin=grand\_mean-grand\_sd,ymax=grand\_mean+grand\_sd,width=.3))+  
 scale\_fill\_manual(values=c("Cell Culture Medium"='blue',"Fascaplysin"="#fdb863","CAS 546102-60-7"="#b2df8a"))+  
 mBio+  
 theme(plot.margin=unit(c(0,0,0,0),units = "pt"), text=element\_text(size=16),axis.text.x=element\_text(angle = 45,vjust = 1,hjust = 1,size=20))+scale\_y\_continuous(limits=c(0,102))+  
 annotate("text",x=ct\_sigtest\_inf$midpoints,y=ct\_sigtest\_inf$y.sig,label=ct\_sigtest\_inf$sig,size=2.5)+  
 geom\_segment(data=statbars,aes(x=x,xend=xend,y=y,yend=yend))+  
 geom\_segment(data=statbars2,aes(x=x,xend=xend,y=y,yend=yend),col="Black")+ylab("Epithelial Cells with C. trachomatis Inclusion (%)")  
  
## Write plot  
  
  
cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_CT\_INFECT\_QUANT),width=6,height = 5.5)#,width = 8,height = 5.5)  
plot(Fig3\_inf)  
dev.off()

## quartz\_off\_screen   
## 2

##Plot Ct Assay EdU  
  
Fig3\_prolif<-ggplot(filter(pi.summary,Observation=="Proliferation"),aes(x=Treatment,y=grand\_mean))+geom\_bar(stat="identity",show.legend = F,aes(fill=Treatment))+geom\_errorbar(aes(ymin=grand\_mean-grand\_sd,ymax=grand\_mean+grand\_sd,width=.3))+mBio+scale\_fill\_manual(values=c("Cell Culture Medium"='blue',SIS3="#c2a5cf","Fascaplysin"="#fdb863","CAS 546102-60-7"="#b2df8a"))+ylab("Epithelial Cells Positive for EdU (%)")+  
 theme(plot.margin=unit(c(0,0,0,0),units = "pt"), text=element\_text(size=20),axis.text.x=element\_text(angle = 45,vjust = 1,hjust = 1))+scale\_y\_continuous(limits=c(0,102))+  
 annotate("text",x=ct\_sigtest\_pro$midpoints,y=ct\_sigtest\_pro$y.sig,label=ct\_sigtest\_pro$sig,size=2.5)+  
 geom\_segment(data=statbars,aes(x=x,xend=xend,y=y,yend=yend))+  
 geom\_segment(data=statbars2,aes(x=x,xend=xend,y=y,yend=yend),col="Black")  
  
##Write plot  
cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_CT\_EDU\_QUANT),width=6,height = 5.5)#width = 8,height = 5.5)  
plot(Fig3\_prolif)  
dev.off()

## quartz\_off\_screen   
## 2

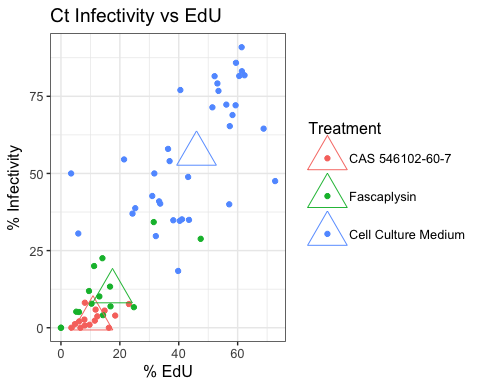
## Print text describing results  
  
## Create variables to hold values for printing  
ct\_sigtest<-unique(ct\_sigtest)  
pro\_C<-ct\_sigtest[ct\_sigtest$Observation=="Proliferation" & ct\_sigtest$reference=="Cell Culture Medium" & ct\_sigtest$xref=="CAS 546102-60-7" ,c("mean\_diff","pval")]  
pro\_F<-ct\_sigtest[ct\_sigtest$Observation=="Proliferation" & ct\_sigtest$reference=="Cell Culture Medium" & ct\_sigtest$xref=="Fascaplysin" ,c("mean\_diff","pval")]  
  
ct\_C<-ct\_sigtest[ct\_sigtest$Observation=="Infectivity" & ct\_sigtest$reference=="Cell Culture Medium" & ct\_sigtest$xref=="CAS 546102-60-7" ,c("mean\_diff","pval")]  
ct\_F<-ct\_sigtest[ct\_sigtest$Observation=="Infectivity" & ct\_sigtest$reference=="Cell Culture Medium" & ct\_sigtest$xref=="Fascaplysin" ,c("mean\_diff","pval")]  
  
  
paste0("Epithelial cell proliferation was decreased by ",-pro\_C$mean\_diff,"% (p=",pro\_C$pval,") in CAS 546102-60-7 and ",-pro\_F$mean\_diff,"% (p=",pro\_F$pval,") in Fascaplysin treated cells relative to Cell Culture Medium treated cells, respectively")

## [1] "Epithelial cell proliferation was decreased by 35.1543095335083% (p=6.10779413826187e-05) in CAS 546102-60-7 and 28.530611023625% (p=0.000353903348334145) in Fascaplysin treated cells relative to Cell Culture Medium treated cells, respectively"

paste0("C. trachomatis infection was decreased by ",-ct\_C$mean\_diff,"% (p=",ct\_C$pval,") in CAS 546102-60-7 and ",-ct\_F$mean\_diff,"% (p=",ct\_F$pval,") in Fascaplysin treated cells relative to Cell Culture Medium treated cells, respectively")

## [1] "C. trachomatis infection was decreased by 53.3866268869905% (p=0.000334606564696431) in CAS 546102-60-7 and 44.5903068287238% (p=0.00100026580119664) in Fascaplysin treated cells relative to Cell Culture Medium treated cells, respectively"

## Calcualte correlation between mean infectivity and mean proliferation  
corr\_mx<-spread(pi.m,Observation, value=percent\_cells)  
ggplot(corr\_mx,aes(y=Infectivity,x=Proliferation,col=Treatment))+geom\_point()+ggtitle("Ct Infectivity vs EdU")+mBio+ylab("% Infectivity")+xlab("% EdU")+geom\_point(data=spread(dplyr::select(pi.summary,-c(grand\_sd,n)),Observation,grand\_mean),aes(y=Infectivity,x=Proliferation,col=Treatment),size=10,pch=2)



ct\_pro\_lm<-lm(formula = Infectivity~Proliferation,data = data.frame(corr\_mx[!rowSums(is.na(corr\_mx))>0,]))  
  
ct\_pro\_summary\_lm<-lm(Infectivity~Proliferation,data=spread(dplyr::select(pi.summary,-c(grand\_sd,n)),Observation,grand\_mean))  
  
summary(ct\_pro\_lm)

##   
## Call:  
## lm(formula = Infectivity ~ Proliferation, data = data.frame(corr\_mx[!rowSums(is.na(corr\_mx)) >   
## 0, ]))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -37.707 -9.275 -1.017 9.703 47.669   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.79531 3.29799 -0.544 0.588   
## Proliferation 1.19670 0.09104 13.145 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 15.21 on 63 degrees of freedom  
## Multiple R-squared: 0.7328, Adjusted R-squared: 0.7286   
## F-statistic: 172.8 on 1 and 63 DF, p-value: < 2.2e-16

summary(ct\_pro\_summary\_lm)

##   
## Call:  
## lm(formula = Infectivity ~ Proliferation, data = spread(dplyr::select(pi.summary,   
## -c(grand\_sd, n)), Observation, grand\_mean))  
##   
## Residuals:  
## 1 2 3   
## 0.6049 -0.7453 0.1404   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -14.27998 1.06922 -13.36 0.0476 \*  
## Proliferation 1.53185 0.03672 41.72 0.0153 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9701 on 1 degrees of freedom  
## Multiple R-squared: 0.9994, Adjusted R-squared: 0.9989   
## F-statistic: 1740 on 1 and 1 DF, p-value: 0.01526

## Combine proliferation sig tests and write to file  
ct\_sigtests<-data.frame(Figure=paste0("FIGURE\_4",ct\_sigtest$Observation),ct\_sigtest)  
ct\_sigtests<-dplyr::select(ct\_sigtests[with(ct\_sigtests,order(Observation,x,xend)),],c(Figure,xref,reference,pval,mean\_diff,sig))  
  
write.csv(ct\_sigtests,file=paste0(thesis\_tables\_directory,TABLE\_CT\_QUANT),row.names=F,quote=F)

# Subject Longitudinal Plots (Figure 1)

Create longitudinal plots for subjects used in study

### Load previously prepared 16S metataxonomic data. metatdata  
load(file=paste0(R\_script\_input\_directory,"subject\_plot\_data.Rdata"))  
OTU\_METADATA<-subject\_plot\_data$OTU\_METADATA ## metadata  
rRNA\_16S<-subject\_plot\_data$relativeAbundance ## taxa assignments/relative abundances  
sampleInfoColNames<-subject\_plot\_data$sampleInfoColNames ## holds which column names are associated with metadata  
  
miRNA\_extractions<-SRL\_meta\_table[is.na(SRL\_meta\_table$QC\_removal\_stage),] ## Samples associated with miRNA libraries used in final study- i.e., don't include samples removed due to poor QC.   
subject\_plot\_list<- unique(miRNA\_extractions$SID) ## Subject IDs to plot  
  
global\_species\_list<-NULL ## Initialize a container to store all species plotted in fig 1 as a legend  
  
for(s in subject\_plot\_list){ ## iterate through the subject's  
 print(s)  
 #s<-"UAB008"  
 ## ////////////////////////////////// ##  
 ## /// Subeset Data by Subject /// ##  
 ## ////////////////////////////////// ##  
   
 relabundance<-filter(rRNA\_16S,SID==s) ## taxa relative abundance for subject  
 otu\_count<-OTU\_METADATA[OTU\_METADATA$SID==s,] ## Metadata for subject  
   
 ## Drop 16S samples that have less than the threshold for high confidence taxa assignments  
 lowCountThreshold<-1000  
 low\_count\_samples<-otu\_count[otu\_count$X16S\_total\_counts<lowCountThreshold,"SERIAL"]  
   
 ##Nugent score data for subject  
 nugent<-filter(OTU\_METADATA,SID==s) %>% dplyr::select(SERIAL,NUGENT\_SCORE)  
 nugent$color<-"grey"  
 ## Colors/handling for missing Nugent scores  
 nugent[is.na(nugent$NUGENT\_SCORE),"color"]<-"red"  
 nugent[!is.na(nugent$NUGENT\_SCORE),"color"]<-"black"  
 nugent[is.na(nugent$NUGENT\_SCORE),"NUGENT\_SCORE"]<-(-1)  
   
 ## pH data for subject  
 ph<-filter(OTU\_METADATA,SID==s) %>% dplyr::select(SERIAL,PH)  
 ## Colors/handling for missing pH values  
 ph$color<-"grey"  
 ph[is.na(ph$PH),"color"]<-"red"  
 ph[!is.na(ph$PH),"color"]<-"black"  
 ph[is.na(ph$PH),"PH"]<-2  
   
 ##MEtadata for subject  
 dailyDiaryMetadata<-filter(OTU\_METADATA,SID==s)  
   
 ## ////////////////////////////////// ##  
 ## Determine global plot time scale ##  
 ## ////////////////////////////////// ##  
   
 SERIAL\_global<-sort(unique(dailyDiaryMetadata$SERIAL,relabundance$SERIAL,miRNA\_extractions$SERIAL)) ## All time points  
 day<-SERIAL\_global%%7 ## Day relative to all SERIALized time points  
 day[SERIAL\_global%%7==0]<-7 ## the mod calcualtion causes all day 7 to be 0. Repalce w/ 7.   
 week<-((SERIAL\_global-day)/7)+1 ## Week back calculated from SERIALized time points.   
 time\_points<-data.frame(SERIAL=SERIAL\_global, day=day, week=week) ## container  
 time\_points$plot\_label=""  
 time\_points[time\_points$day==7,"plot\_label"]<-time\_points[time\_points$day==7,"week"] ## plot labels every week  
 global\_min\_time<-max(c(min(miRNA\_extractions[miRNA\_extractions$SID==s,"SERIAL"]-5),min(SERIAL\_global,na.rm = T) ))  
   
 ## min time point - the max of the min of either miR extractions or all data  
 global\_max\_time<-min(c(max(miRNA\_extractions[miRNA\_extractions$SID==s,"SERIAL"]+5),max(SERIAL\_global,na.rm = T))) ## max time point across all data  
 removed\_samples  
 ## cleaner way to define as variable for X axis:  
 time\_breaks<-time\_points$SERIAL   
 time\_label<-time\_points$plot\_label  
 time\_limits<-c(global\_min\_time-1,global\_max\_time+1)  
   
 ## Determine the min & max of time points  
 rect\_min<-time\_breaks[!time\_breaks%in% miRNA\_extractions[miRNA\_extractions$SID==s,"SERIAL"]]-.5  
 rect\_max<-time\_breaks[!time\_breaks%in% miRNA\_extractions[miRNA\_extractions$SID==s,"SERIAL"]]+.5  
   
 ## ////////////////////////////////// ##  
 ## / Determine most adbundant species ##  
 ## ////////////////////////////////// ##  
   
 # ////////////////////////////////////////////////////////////////////////  
 # Most abundant species defined per subject & based on cutoff. ##  
 # All other taxa binned into "other" #  
 # ////////////////////////////////////////////////////////////////////////  
   
 ### Grab just relative abundances, no sample info columns:  
 relabundance[is.na(relabundance)]<-0  
 relabundance<-relabundance[relabundance$SERIAL>=time\_limits[1] & relabundance$SERIAL<=time\_limits[2],]  
 relabundance\_for\_max\_calc<-relabundance[,!(names(relabundance) %in% sampleInfoColNames)]  
   
 ### Calucalte max for each taxa  
 max\_relabundances<-apply(relabundance\_for\_max\_calc,2,max)  
 ### number of species whose rel abundance is above a certain threshold  
 numHighAbundSpecies<-sum(max\_relabundances>raThreshold,na.rm = T)   
   
 ##### Plot either the top X most abundant species,   
 ##### or the most abundant species above max plottable species, whichever is lesser.  
 ##### This helps prevent "taxa overload" on the plot.   
   
 most\_abundant\_species<-"" ## Will hold names of most abundant species.   
 if(numHighAbundSpecies>nSpecies){   
 most\_abundant\_species<-names(sort(max\_relabundances,decreasing = T))[1:nSpecies]  
 }else{  
 most\_abundant\_species<- names(max\_relabundances[!is.na(max\_relabundances) & max\_relabundances>raThreshold])  
 }  
   
 ##### Pull the most abundant species (defined above), bin the remainder into "other".  
 ##### Also recombine the 'sample info' onto most abundant/other species table.   
   
 most\_abundant\_relabundance<-relabundance[most\_abundant\_species]  
 other\_relabundance<-rowSums(relabundance[!(names(relabundance) %in% most\_abundant\_species |   
 names(relabundance) %in% sampleInfoColNames)],na.rm=T)  
 otu\_count\_relabundance<-cbind(relabundance[names(relabundance) %in% c("SERIAL",sampleInfoColNames)],  
 most\_abundant\_relabundance,Other=other\_relabundance)  
   
 ##Update global species list with any new species  
 global\_species\_list<-unique(c(most\_abundant\_species,global\_species\_list,"Other"))  
   
   
 reshape\_names<-c("Pre\_QC\_ID","SID","UID","SERIAL")  
 otu\_count\_reshape<-melt(data = otu\_count\_relabundance,id.vars=reshape\_names)  
 names(otu\_count\_reshape)<-c(reshape\_names,"species","count")   
   
 ### ///////////////////////////////////////////// ###  
 ### ///////// OTU Plot ///////// ###  
 ### ///////////////////////////////////////////// ###  
   
 otuPlot<- ggplot(otu\_count\_reshape)+  
 geom\_area(aes(x=SERIAL,y=count,fill=species),  
 stat="identity",show.legend=F,position="fill")+#,width=1)+  
 mBio+  
 theme(axis.ticks = element\_blank(),  
 axis.title.x=element\_blank(),  
 axis.text.x = element\_blank(),  
 axis.text.y = element\_text(size = rel(1.5)),  
 legend.key.size=unit(8, "points"),  
 legend.title = element\_blank(),  
 legend.text = element\_text(size = rel(.5),face="italic"),  
 axis.title = element\_text(size = rel(sizes)),  
 plot.margin=unit(c(2.5,40,2.5,10),units="points"),#margins  
 panel.grid.major.y=element\_line(colour = "grey73"),  
 panel.grid.minor.x = element\_blank())+  
 ylab("Phylotype Relative\nAbundance (%)")+  
 geom\_vline(xintercept = otu\_count\_reshape$SERIAL,size=rel(.2),col="grey")+  
 scale\_fill\_manual(values=subject\_long\_taxa\_colors)+  
 ggtitle(paste0(s))+  
 annotate("rect", xmin=rect\_min, xmax=rect\_max, ymin=0, ymax=1, alpha=alpha\_rect, fill=rect\_fill)+  
 scale\_x\_continuous(breaks=time\_breaks,label=time\_label,limits=time\_limits)  
   
 ## Determine any dropped sample (post QC) time points and place an \* above   
 dropped\_samples.serial<-otu\_count[otu\_count$Pre\_QC\_ID %in% removed\_samples$Pre\_QC\_ID,"SERIAL"]  
 if(length(dropped\_samples.serial)!=0){otuPlot<-otuPlot+annotate("text", x =dropped\_samples.serial , y = 1.01, label = "\*",size=8)}  
   
   
 ### ///////////////////////////////////////////// ###  
 ### ///////// Nugent Plot ///////// ###  
 ### ///////////////////////////////////////////// ###  
 nugentPlot<-ggplot(data=nugent)+  
 geom\_bar(aes(x=as.numeric(SERIAL),y=NUGENT\_SCORE,width=.9,fill=color),  
 stat="identity",position = position\_dodge(width=0.5))+  
 mBio+  
 geom\_hline(yintercept = c(3,7),size=rel(1),col="pink")+ ## Defines Nugent score 3 & 7 (BV   
 theme(axis.ticks = element\_blank(),  
 axis.title.x=element\_blank(),  
 axis.text.x = element\_blank(),  
 axis.text.y = element\_text(size = rel(1.5\*sizes)),  
 legend.position="none",  
 plot.margin=margins,  
 panel.grid.major.y=element\_line(colour = "grey73"),  
 panel.grid.minor.x = element\_blank())+  
 ylab("Nugent\nScore")+  
 scale\_fill\_manual(values=c('black','red'))+  
 scale\_x\_continuous(breaks=time\_breaks,label=time\_label,limits=time\_limits)+  
 annotate("rect", xmin=rect\_min, xmax=rect\_max, ymin=0, ymax=10, alpha=alpha\_rect, fill=rect\_fill)+  
 scale\_y\_continuous(breaks=c(0,3,7,10),limits=c(-1.25,10.5))  
   
   
 ### ///////////////////////////////////////////// ###  
 ### ///////// Metadata Plot ///////// ###  
 ### ///////////////////////////////////////////// ###  
 metaPlot<-ggplot(dailyDiaryMetadata, aes(x=as.numeric(SERIAL)))+  
 geom\_point(aes(y=1\*(as.numeric(VAG\_DIS)==1),size=2),  
 pch=16,col='blue',position = position\_dodge(width=0.5))+  
 geom\_point(aes(y=2\*(as.numeric(VAG\_ODOR)==1),size=2),  
 pch=16,col='blue',position = position\_dodge(width=0.5))+  
 geom\_point(aes(y=3\*(as.numeric(MENSTRUATION)>0),size=as.numeric(MENSTRUATION)),  
 col='red',pch=16, position = position\_dodge(width=0.5))+  
 scale\_size(range = c(2+2,4+2))+ ## for menstru point sizes. Bounded by 3 points on a scale.   
 mBio+  
 theme(legend.position="none",  
 plot.title = element\_text(size = rel(sizes)),  
 axis.text = element\_text(size = rel(2\*sizes)),  
 axis.title = element\_text(size = rel(sizes)),  
 axis.title.y=element\_text(vjust=.2),  
 axis.title.x=element\_text(vjust=-.2),  
 axis.text.y=element\_text(size=rel(0.75)), ## change back to 0.75  
 plot.margin=unit(c(-2.5,40,5,5),units="points"),  
 panel.grid.major.y=element\_line(colour = "grey73"),  
 panel.grid.minor.x = element\_blank())+  
 xlab("Week")+  
 ylab("")+  
 annotate("rect", xmin=rect\_min, xmax=rect\_max, ymin=1, ymax=3, alpha=alpha\_rect, fill=rect\_fill)+  
 scale\_x\_continuous(breaks=time\_breaks,label=time\_label,limits=time\_limits)+  
 scale\_y\_continuous(breaks=1:3,labels = c("Discharge",  
 "Odor",  
 "Menstruation"),limits=c(0.5,3.5))  
   
 ###### Determine if subject has low qualiyt daily dairy flag and annotate plot:  
   
 if(!sum(is.na(dailyDiaryMetadata$Diary\_QUALITY\_FLAG))){  
 min<-min(timeTable$SERIAL,na.rm = T)  
 max<-max(timeTable$SERIAL,na.rm = T)  
 middle.x<-(max-min)/2  
 middle.y<<-(16-1)/2  
 metaPlot<-metaPlot + annotate("text", x = middle.x,   
 y = middle.y, label = "?",  
 size=rel(40),  
 col="grey")  
 }  
   
 ### ///////////////////////////////////////////// ###  
 ### ///////// pH Plot ///////// ###  
 ### ///////////////////////////////////////////// ###  
 ph\_normalization\_factor<-3.5  
 ## Notice the pH value scale is "normalized" by subtracting "ph\_normalization\_factor" from the actual pH value, then re-labeling the y axis. This is very dangerous, but ggplot will not permit bar plots that start from > 0 .   
 ph[ph$PH==2,"PH"]<-ph\_normalization\_factor-0.5  
   
 phPlot<-ggplot(ph,aes(x=as.numeric(SERIAL),  
 y=as.numeric(PH)-ph\_normalization\_factor,width=.9,fill=color))+   
 geom\_bar(stat="identity",position = position\_dodge(width=0.5))+  
 scale\_fill\_manual(values=c('black','red'))+  
 mBio+  
 geom\_hline(yintercept = c(4.5-ph\_normalization\_factor),  
 size=rel(1),col="pink")+ ## Vaginal pH>4.5 one criteria for BV.   
 theme(axis.ticks = element\_blank(),  
 axis.title.x=element\_blank(),  
 axis.text.x = element\_blank(),  
 axis.text = element\_text(size = rel(1.5\*sizes)),  
 legend.position="none",  
 plot.margin=margins,  
 panel.grid.major.y=element\_line(colour = "grey73"),  
 panel.grid.minor.x = element\_blank())+  
 ylab("pH")+  
 annotate("rect", xmin=rect\_min, xmax=rect\_max, ymin=min(0,min(ph$PH,na.rm = T)-ph\_normalization\_factor-0.25), ymax= max(ph$PH,na.rm = T)+0.25-ph\_normalization\_factor, alpha=alpha\_rect, fill=rect\_fill)+  
 scale\_x\_continuous(breaks=time\_breaks,label=time\_label,limits=time\_limits)+  
 scale\_y\_continuous(breaks=c(4,4.5,5,5.5)-ph\_normalization\_factor, limits=c(-0.75,5.8+0.25-ph\_normalization\_factor),labels=c("4","4.5","5","5.5"))  
   
   
 ### ///////////////////////////////////////////// ###  
 ### ///////// Tie Plots Together ///////// ###  
 ### ///////////////////////////////////////////// ###  
   
 ## ////////////////////////// ##  
 ## //Define plots as Grobs/// ##  
 ## ////////////////////////// ##  
   
 grob.otuPlot <- ggplotGrob(otuPlot)  
 grob.nugentPlot <- ggplotGrob(nugentPlot)  
 grob.phPlot<-ggplotGrob(phPlot)  
 grob.metaPlot <- ggplotGrob(metaPlot)  
   
 ## ////////////////////////// ##  
 ## /// Find max width /// ##  
 ## ////////////////////////// ##  
 maxWidth = grid::unit.pmax(grob.otuPlot$widths[1:6],  
 grob.nugentPlot$widths[1:5],  
 grob.phPlot$widths[1:5],  
 grob.metaPlot$widths[1:5])  
   
 ## ////////////////////////// ##  
 ## /Redefine common max width ##  
 ## ////////////////////////// ##  
 grob.nugentPlot$widths[1:6] <- as.list(maxWidth)  
 grob.otuPlot$widths[1:6] <- as.list(maxWidth)  
 grob.metaPlot$widths[1:6] <- as.list(maxWidth)  
 grob.phPlot$widths[1:6]<-as.list(maxWidth)  
   
 ### ///////////////////////////////////////////// ###  
 ### ///////// Write/Draw Plot ///////// ###  
 ### ///////////////////////////////////////////// ###  
   
 cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_SUBJECT\_PLOTS,s,".eps"),width = 11,height = 8.5)  
   
 grid.arrange(grob.otuPlot,  
 grob.nugentPlot,  
 grob.phPlot,  
 grob.metaPlot,   
 ncol=1,nrow=4,  
 heights=c(2.5,1,1,1))   
 dev.off()  
}

## [1] "EM12"

## [1] "UAB003"

## [1] "UAB005"

## [1] "UAB006"

## [1] "UAB007"

## [1] "UAB008"

## [1] "UAB015"

## [1] "UAB021"

## [1] "UAB022"

## [1] "UAB055"

## [1] "UAB093"

## [1] "UAB102"

## [1] "UAB115"

## [1] "UAB116"

## [1] "UAB117"

## [1] "UAB121"

## //////  
## The following plots the figure legend containing colors for all taxa plotted in for loos  
  
dummy\_globalSpeciesList<-data.frame(SERIAL=1,species=global\_species\_list,count=1)  
dummy\_globalSpeciesList\_plot<-ggplot(dummy\_globalSpeciesList)+  
 geom\_area(aes(x=SERIAL,y=count,fill=species,order=order(as.numeric(as.factor(dummy\_globalSpeciesList$species)),decreasing = F)),  
 stat="identity",show.legend=T,position="fill")+#,width=1)+  
 mBio+  
 theme(axis.ticks = element\_blank(),  
 axis.title.x=element\_blank(),  
 axis.text.x = element\_blank(),  
 legend.key.size=unit(8, "points"),  
 legend.title = element\_blank(),  
 legend.text = element\_text(size = rel(.5),face="italic"),  
 axis.title = element\_text(size = rel(sizes)),  
 plot.margin=unit(c(2.5,40,2.5,10),units="points"),#margins  
 panel.grid.major.y=element\_line(colour = "grey73"))+  
 scale\_fill\_manual(values=subject\_long\_taxa\_colors)+  
 ggtitle("Global Species List Color Codes")  
  
##Write legend to file  
cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_SUBJECT\_PLOTS,"GlobalSpeciesList.eps"),width = 11,height = 8.5)  
dummy\_globalSpeciesList\_plot  
dev.off()

## quartz\_off\_screen   
## 2

# Ribo-reduced RNA-seq Analysis

## RIN Quality Distribition   
TRL\_RNA\_Sample\_QuantQual<-read.csv(paste0(R\_script\_input\_directory,"TRL\_RNA\_Sample\_Quality.csv"),header = F)  
postscript(paste0(thesis\_figures\_directory,FIGURE\_TRL\_RIN\_HIST))  
p<-ggplot(TRL\_RNA\_Sample\_QuantQual,aes(x=V1))+geom\_histogram()+xlab("RINe")+ylab("Number of Samples")+geom\_vline(xintercept = median(TRL\_RNA\_Sample\_QuantQual$V1),col='red')+mBio  
plot(p)

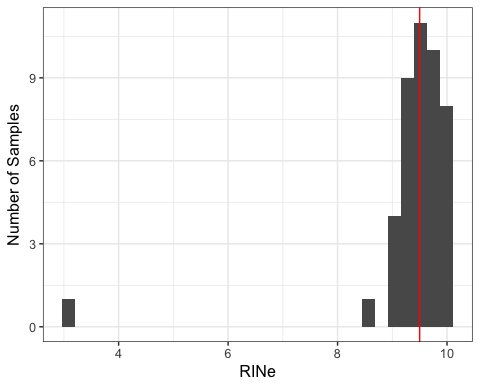
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

dev.off()

## quartz\_off\_screen   
## 2

plot(p)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



summary(TRL\_RNA\_Sample\_QuantQual$V1)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.100 9.200 9.500 9.366 9.800 10.000

summary(10\*TRL\_RNA\_Sample\_QuantQual[TRL\_RNA\_Sample\_QuantQual$V1!=3.1,"V2"])

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 226.0 438.5 634.0 616.5 767.5 1120.0

## Read counts data in from server

### ---------------------------------  
## Set up unix connection using sshfs  
### ---------------------------------  
Sys.which('bash')  
Sys.which('sh')  
#echo hello world  
#   
system("sshfs stsmith@medusa.igs.umaryland.edu:/local/projects-t2/HRBV/ ~/IGS/sshfs\_medusa/", intern = FALSE,  
 ignore.stdout = FALSE, ignore.stderr = FALSE,  
 wait = TRUE, input = NULL, show.output.on.console = TRUE,  
 minimized = FALSE, invisible = F)  
  
manifest<-read.table("~/IGS/sshfs\_medusa/Config/W100083533.manifest",sep = "\t",header = T)  
TRL\_counts\_table<-data.frame(Feature="None")  
  
for(i in 1:length(manifest$Pre\_QC\_ID)){  
 #i<-3  
 sample\_name<-as.character(manifest[i,"Pre\_QC\_ID"])  
 sample\_path<-paste0("~/IGS/sshfs\_medusa/TRL/alignment/TRL\_",sample\_name,"\_aln\_hg19/TRL\_",sample\_name,".hg\_counts")  
 new\_table<-read.table(sample\_path,header=T,sep="\t")  
 names(new\_table)<-c("Feature",sample\_name)  
 #sub\_feature\_set<-!new\_table$Feature %in% c(filtered\_out\_features,"mirna\_info")  
 ## Don't forget about ambigous/none features!  
 #new\_table<-new\_table[sub\_feature\_set,]  
 TRL\_counts\_table<-merge(TRL\_counts\_table,new\_table,by.x = "Feature",by.y="Feature",all=T)  
}  
  
row.names(TRL\_counts\_table)<-TRL\_counts\_table$Feature  
TRL\_counts\_table<-TRL\_counts\_table[!TRL\_counts\_table$Feature=="None",]  
TRL\_counts\_table<-dplyr::select(TRL\_counts\_table,-c(Feature,POSITIVE\_CONTROL))  
head(TRL\_counts\_table)  
write.csv(TRL\_counts\_table,file=paste0(R\_script\_output\_directory,TABLE\_TRL\_COUNTS\_RAW),row.names = T,quote=F)

## Create and Prepare Tables

TRL\_counts\_table<-read.csv(file=paste0(R\_script\_output\_directory,TABLE\_TRL\_COUNTS\_RAW),row.names = 1)  
write.csv(TRL\_counts\_table,file=paste0(thesis\_tables\_directory,TABLE\_TRL\_COUNTS\_RAW),row.names = T,quote=F)  
  
## Filter out ambiguous, no feature reads. Make note of how many there are  
TRL\_counts\_table[row.names(TRL\_counts\_table) %in% c("alignment\_not\_unique","no\_feature","ambiguous"),]

## VK2\_GVAGINALIS\_BCS\_13HR\_rep1  
## alignment\_not\_unique 1659810  
## ambiguous 72201  
## no\_feature 3075288  
## VK2\_LCRISPATUS\_BCS\_13HR\_rep2  
## alignment\_not\_unique 1412089  
## ambiguous 36836  
## no\_feature 3279466  
## VK2\_LJENSENII\_BCS\_13HR\_rep2 VK2\_LINERS\_BCS\_13HR\_rep2  
## alignment\_not\_unique 1568882 657465  
## ambiguous 43491 24472  
## no\_feature 3468681 2136668  
## VK2\_GVAGINALIS\_BCS\_13HR\_rep2 VK2\_MEDIA\_BCS\_13HR\_rep1  
## alignment\_not\_unique 1019153 813706  
## ambiguous 47530 32173  
## no\_feature 3914945 3257161  
## VK2\_LCRISPATUS\_BCS\_22HR\_rep1  
## alignment\_not\_unique 888176  
## ambiguous 20079  
## no\_feature 3186044  
## VK2\_LJENSENII\_BCS\_22HR\_rep1 VK2\_LINERS\_BCS\_22HR\_rep1  
## alignment\_not\_unique 1547911 479715  
## ambiguous 47238 22808  
## no\_feature 2931450 2268805  
## VK2\_LJENSENII\_BCS\_4HR\_rep2 VK2\_LINERS\_BCS\_4HR\_rep2  
## alignment\_not\_unique 1056023 1259736  
## ambiguous 48041 43226  
## no\_feature 3037420 3488226  
## VK2\_GVAGINALIS\_BCS\_4HR\_rep2 VK2\_MEDIA\_BCS\_4HR\_rep2  
## alignment\_not\_unique 712114 1230561  
## ambiguous 26952 44876  
## no\_feature 2369104 3902295  
## VK2\_LCRISPATUS\_BCS\_4HR\_rep1  
## alignment\_not\_unique 3511899  
## ambiguous 79107  
## no\_feature 5717691  
## VK2\_LJENSENII\_BCS\_4HR\_rep1 VK2\_LINERS\_BCS\_4HR\_rep1  
## alignment\_not\_unique 673805 968908  
## ambiguous 19770 33938  
## no\_feature 4664422 4209178  
## VK2\_GVAGINALIS\_BCS\_4HR\_rep1 VK2\_MEDIA\_BCS\_4HR\_rep1  
## alignment\_not\_unique 1596106 1336499  
## ambiguous 51607 52997  
## no\_feature 3658460 4357425  
## VK2\_LCRISPATUS\_BCS\_13HR\_rep1  
## alignment\_not\_unique 1467776  
## ambiguous 41804  
## no\_feature 2886596  
## VK2\_LJENSENII\_BCS\_13HR\_rep1 VK2\_LINERS\_BCS\_13HR\_rep1  
## alignment\_not\_unique 1009315 378146  
## ambiguous 25940 11378  
## no\_feature 3342529 3070524  
## VK2\_MEDIA\_BCS\_22HR\_rep1 VK2\_LCRISPATUS\_BCS\_4HR\_rep2  
## alignment\_not\_unique 1531300 774318  
## ambiguous 66155 20755  
## no\_feature 5086258 3881756  
## VK2\_MEDIA\_BCS\_13HR\_rep2 VK2\_LCRISPATUS\_BCS\_22HR\_rep2  
## alignment\_not\_unique 3011458 3026810  
## ambiguous 92696 60297  
## no\_feature 3169404 4151315  
## VK2\_LJENSENII\_BCS\_22HR\_rep2 VK2\_LINERS\_BCS\_22HR\_rep2  
## alignment\_not\_unique 1190320 462056  
## ambiguous 30935 14413  
## no\_feature 4117821 3010909  
## VK2\_GVAGINALIS\_BCS\_22HR\_rep2  
## alignment\_not\_unique 453069  
## ambiguous 16728  
## no\_feature 3168877  
## VK2\_GVAGINALIS\_BCS\_4HR\_rep3 VK2\_MEDIA\_BCS\_4HR\_rep3  
## alignment\_not\_unique 2380178 1313853  
## ambiguous 93189 60327  
## no\_feature 4742894 4355267  
## VK2\_LCRISPATUS\_BCS\_13HR\_rep3  
## alignment\_not\_unique 188513  
## ambiguous 2133  
## no\_feature 2477532  
## VK2\_LJENSENII\_BCS\_13HR\_rep3 VK2\_LINERS\_BCS\_13HR\_rep3  
## alignment\_not\_unique 1389726 3545491  
## ambiguous 54050 108724  
## no\_feature 4957907 4458395  
## VK2\_GVAGINALIS\_BCS\_13HR\_rep3 VK2\_MEDIA\_BCS\_13HR\_rep3  
## alignment\_not\_unique 5904365 2844329  
## ambiguous 230423 73038  
## no\_feature 5816169 5443374  
## VK2\_LCRISPATUS\_BCS\_22HR\_rep3  
## alignment\_not\_unique 7539963  
## ambiguous 171506  
## no\_feature 5283834  
## VK2\_LJENSENII\_BCS\_22HR\_rep3 VK2\_LINERS\_BCS\_22HR\_rep3  
## alignment\_not\_unique 6003165 2546317  
## ambiguous 195482 101360  
## no\_feature 5238683 4494104  
## VK2\_GVAGINALIS\_BCS\_22HR\_rep3 VK2\_MEDIA\_BCS\_22HR\_rep3  
## alignment\_not\_unique 2979388 2308331  
## ambiguous 87512 104417  
## no\_feature 4729799 4141951

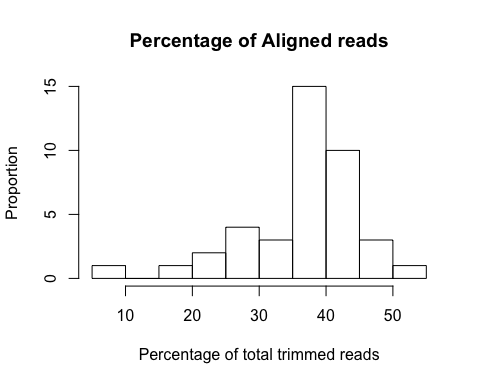
## Percentage of ambiguous/no feature/non unique reads & alignment stats  
(ambig\_nofeat<-colSums(TRL\_counts\_table[row.names(TRL\_counts\_table) %in% c("alignment\_not\_unique","no\_feature","ambiguous"),]))

## VK2\_GVAGINALIS\_BCS\_13HR\_rep1 VK2\_LCRISPATUS\_BCS\_13HR\_rep2   
## 4807299 4728391   
## VK2\_LJENSENII\_BCS\_13HR\_rep2 VK2\_LINERS\_BCS\_13HR\_rep2   
## 5081054 2818605   
## VK2\_GVAGINALIS\_BCS\_13HR\_rep2 VK2\_MEDIA\_BCS\_13HR\_rep1   
## 4981628 4103040   
## VK2\_LCRISPATUS\_BCS\_22HR\_rep1 VK2\_LJENSENII\_BCS\_22HR\_rep1   
## 4094299 4526599   
## VK2\_LINERS\_BCS\_22HR\_rep1 VK2\_LJENSENII\_BCS\_4HR\_rep2   
## 2771328 4141484   
## VK2\_LINERS\_BCS\_4HR\_rep2 VK2\_GVAGINALIS\_BCS\_4HR\_rep2   
## 4791188 3108170   
## VK2\_MEDIA\_BCS\_4HR\_rep2 VK2\_LCRISPATUS\_BCS\_4HR\_rep1   
## 5177732 9308697   
## VK2\_LJENSENII\_BCS\_4HR\_rep1 VK2\_LINERS\_BCS\_4HR\_rep1   
## 5357997 5212024   
## VK2\_GVAGINALIS\_BCS\_4HR\_rep1 VK2\_MEDIA\_BCS\_4HR\_rep1   
## 5306173 5746921   
## VK2\_LCRISPATUS\_BCS\_13HR\_rep1 VK2\_LJENSENII\_BCS\_13HR\_rep1   
## 4396176 4377784   
## VK2\_LINERS\_BCS\_13HR\_rep1 VK2\_MEDIA\_BCS\_22HR\_rep1   
## 3460048 6683713   
## VK2\_LCRISPATUS\_BCS\_4HR\_rep2 VK2\_MEDIA\_BCS\_13HR\_rep2   
## 4676829 6273558   
## VK2\_LCRISPATUS\_BCS\_22HR\_rep2 VK2\_LJENSENII\_BCS\_22HR\_rep2   
## 7238422 5339076   
## VK2\_LINERS\_BCS\_22HR\_rep2 VK2\_GVAGINALIS\_BCS\_22HR\_rep2   
## 3487378 3638674   
## VK2\_GVAGINALIS\_BCS\_4HR\_rep3 VK2\_MEDIA\_BCS\_4HR\_rep3   
## 7216261 5729447   
## VK2\_LCRISPATUS\_BCS\_13HR\_rep3 VK2\_LJENSENII\_BCS\_13HR\_rep3   
## 2668178 6401683   
## VK2\_LINERS\_BCS\_13HR\_rep3 VK2\_GVAGINALIS\_BCS\_13HR\_rep3   
## 8112610 11950957   
## VK2\_MEDIA\_BCS\_13HR\_rep3 VK2\_LCRISPATUS\_BCS\_22HR\_rep3   
## 8360741 12995303   
## VK2\_LJENSENII\_BCS\_22HR\_rep3 VK2\_LINERS\_BCS\_22HR\_rep3   
## 11437330 7141781   
## VK2\_GVAGINALIS\_BCS\_22HR\_rep3 VK2\_MEDIA\_BCS\_22HR\_rep3   
## 7796699 6554699

(aligned<-colSums(TRL\_counts\_table[!row.names(TRL\_counts\_table) %in% c("alignment\_not\_unique","no\_feature","ambiguous"),]))

## VK2\_GVAGINALIS\_BCS\_13HR\_rep1 VK2\_LCRISPATUS\_BCS\_13HR\_rep2   
## 3483930 3024389   
## VK2\_LJENSENII\_BCS\_13HR\_rep2 VK2\_LINERS\_BCS\_13HR\_rep2   
## 2974822 1729117   
## VK2\_GVAGINALIS\_BCS\_13HR\_rep2 VK2\_MEDIA\_BCS\_13HR\_rep1   
## 3155228 2030629   
## VK2\_LCRISPATUS\_BCS\_22HR\_rep1 VK2\_LJENSENII\_BCS\_22HR\_rep1   
## 1416161 2950478   
## VK2\_LINERS\_BCS\_22HR\_rep1 VK2\_LJENSENII\_BCS\_4HR\_rep2   
## 1532055 3206459   
## VK2\_LINERS\_BCS\_4HR\_rep2 VK2\_GVAGINALIS\_BCS\_4HR\_rep2   
## 3115872 1880874   
## VK2\_MEDIA\_BCS\_4HR\_rep2 VK2\_LCRISPATUS\_BCS\_4HR\_rep1   
## 3207936 5558855   
## VK2\_LJENSENII\_BCS\_4HR\_rep1 VK2\_LINERS\_BCS\_4HR\_rep1   
## 1533479 2645213   
## VK2\_GVAGINALIS\_BCS\_4HR\_rep1 VK2\_MEDIA\_BCS\_4HR\_rep1   
## 3555610 3745190   
## VK2\_LCRISPATUS\_BCS\_13HR\_rep1 VK2\_LJENSENII\_BCS\_13HR\_rep1   
## 3386120 1921098   
## VK2\_LINERS\_BCS\_13HR\_rep1 VK2\_MEDIA\_BCS\_22HR\_rep1   
## 863055 4670961   
## VK2\_LCRISPATUS\_BCS\_4HR\_rep2 VK2\_MEDIA\_BCS\_13HR\_rep2   
## 1648935 4892080   
## VK2\_LCRISPATUS\_BCS\_22HR\_rep2 VK2\_LJENSENII\_BCS\_22HR\_rep2   
## 4392899 2049302   
## VK2\_LINERS\_BCS\_22HR\_rep2 VK2\_GVAGINALIS\_BCS\_22HR\_rep2   
## 1021441 1354926   
## VK2\_GVAGINALIS\_BCS\_4HR\_rep3 VK2\_MEDIA\_BCS\_4HR\_rep3   
## 5603541 5132964   
## VK2\_LCRISPATUS\_BCS\_13HR\_rep3 VK2\_LJENSENII\_BCS\_13HR\_rep3   
## 161415 3802258   
## VK2\_LINERS\_BCS\_13HR\_rep3 VK2\_GVAGINALIS\_BCS\_13HR\_rep3   
## 6838588 10538780   
## VK2\_MEDIA\_BCS\_13HR\_rep3 VK2\_LCRISPATUS\_BCS\_22HR\_rep3   
## 4873897 8448815   
## VK2\_LJENSENII\_BCS\_22HR\_rep3 VK2\_LINERS\_BCS\_22HR\_rep3   
## 8633553 5445716   
## VK2\_GVAGINALIS\_BCS\_22HR\_rep3 VK2\_MEDIA\_BCS\_22HR\_rep3   
## 5545096 7577938

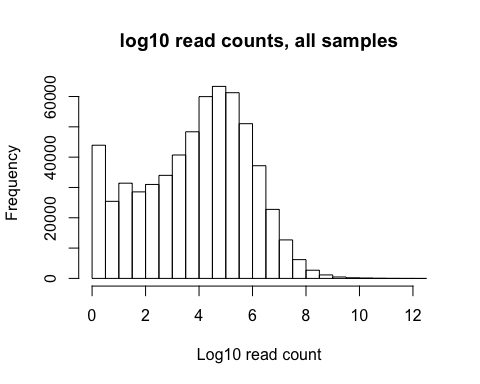
alignment\_stats<-data.frame(aligne=aligned,non\_aligned=ambig\_nofeat,aligned.percent=100\*aligned/colSums(TRL\_counts\_table))  
write.csv(alignment\_stats,file = paste0(thesis\_tables\_directory,TABLE\_TRL\_ALIGNSTATS))  
  
## Look at proportion of Ambigous/no feature/non-unique  
hist(alignment\_stats$aligned.percent,main="Percentage of Aligned reads",xlab="Percentage of total trimmed reads",ylab="Proportion")



ambig\_nofeat\_readpercent.high<-alignment\_stats[(100-alignment\_stats$aligned.percent)>80,]  
paste0("The following samples have ambigous reads >80%: ", str\_c(row.names(ambig\_nofeat\_readpercent.high),collapse = ", "))

## [1] "The following samples have ambigous reads >80%: VK2\_LINERS\_BCS\_13HR\_rep1, VK2\_LCRISPATUS\_BCS\_13HR\_rep3"

#Filter out ambigous reads from counts  
TRL\_counts\_table<-TRL\_counts\_table[!row.names(TRL\_counts\_table) %in% c("alignment\_not\_unique","no\_feature","ambiguous"),]  
TRL\_counts\_table<-TRL\_counts\_table[,order(names(TRL\_counts\_table))]  
hist(log(as.matrix(TRL\_counts\_table)),main="log10 read counts, all samples",xlab="Log10 read count")



##Create Design Table  
TRL\_design<-data.frame(Sample=names(TRL\_counts\_table))  
row.names(TRL\_design)<-TRL\_design$Sample  
TRL\_design<-separate(TRL\_design,Sample,sep = "\_",into = c("CellLine","BCS","DROP","ExposureTime","rep")) %>% dplyr::select(-c(CellLine,DROP))  
  
## Create replicate #, lactobacillus indicator. Then combine BCS and exposure time for 'group1' for eventually creating constrasts  
TRL\_design$rep<-as.numeric(gsub(TRL\_design$rep,pattern = "rep",replacement = ""))  
TRL\_design$Lactobacillus<-grepl(TRL\_design$BCS,pattern = "^L")  
TRL\_design$group1 <- factor(paste(TRL\_design$BCS,TRL\_design$ExposureTime,sep="."))  
TRL\_design$BCS<-factor(TRL\_design$BCS,levels = c("LCRISPATUS","LJENSENII","LINERS","GVAGINALIS","MEDIA"),ordered = T)  
TRL\_design$ExposureTime<-factor(TRL\_design$ExposureTime,levels = c("4HR","13HR","22HR"),ordered=T)  
sample\_order<-with(TRL\_design,order(ExposureTime,BCS,rep))  
  
##Make an expression set object for coupled handling of coutns and design matrix  
TRL\_design<-TRL\_design[sample\_order,]  
TRL\_counts\_table<-TRL\_counts\_table[,sample\_order]  
TRL\_counts\_meta<-ExpressionSet(assayData = as.matrix(TRL\_counts\_table),phenoData = AnnotatedDataFrame(TRL\_design))

## Plot Replicates

## THis may take a while given there are ~20,000 points to plot for each comparison. Consdier trimming the lower expresed reads by using rmlow = log(10,10) for example  
postscript(paste0(R\_script\_output\_directory,"TRL\_RNASeq\_Replicate\_plots.eps"),width = 10,height = 8)  
hist(colSums(exprs(TRL\_counts\_meta)))  
  
plot\_replicates(TRL\_counts\_meta,BCS.selection = c("LCRISPATUS"),ExposureTime.selection = c("4HR","13HR","22HR"),logt=T)  
plot\_replicates(TRL\_counts\_meta,BCS.selection = c("LJENSENII"),ExposureTime.selection = c("4HR","13HR","22HR"),logt=T)  
plot\_replicates(TRL\_counts\_meta,BCS.selection = c("LINERS"),ExposureTime.selection = c("4HR","13HR","22HR"),logt=T)  
plot\_replicates(TRL\_counts\_meta,BCS.selection = c("GVAGINALIS"),ExposureTime.selection = c("4HR","13HR","22HR"),logt=T)  
plot\_replicates(TRL\_counts\_meta,BCS.selection = c("MEDIA"),ExposureTime.selection = c("4HR","13HR","22HR"),logt=T)  
  
dev.off()

## Drop samples

dropped\_TRL\_samples<-c("VK2\_MEDIA\_BCS\_4HR\_rep3","VK2\_LCRISPATUS\_BCS\_13HR\_rep3") ## determined from replciate plots and # of ambigous (rRNA) reads.  
  
##Drop poor QC samples  
TRL\_counts\_meta.qc<-subset\_ExpressionSet(TRL\_counts\_meta,filterOut =dropped\_TRL\_samples )  
  
## Proportion of genes with at least 1 read across all samples  
sum(rowSums(exprs(TRL\_counts\_meta.qc)>0)>0)/nrow(exprs(TRL\_counts\_meta.qc))

## [1] 0.7899965

#exprs(TRL\_counts\_meta.qc)[rowSums(exprs(TRL\_counts\_meta.qc)>0)>0,]  
  
##Summary of # of samples with at least one read across all samples  
summary(rowSums(exprs(TRL\_counts\_meta.qc)[rowSums(exprs(TRL\_counts\_meta.qc)>0)>0,]>0))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.0 19.0 38.0 28.8 38.0 38.0

##Total post QC samples:  
ncol(exprs(TRL\_counts\_meta.qc))

## [1] 38

## Summary of remaining total read counts  
summary(colSums(exprs(TRL\_counts\_meta.qc)))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 863100 1948000 3207000 3796000 4888000 10540000

(reps\_per\_treatment<-ddply(data.frame(pData(TRL\_counts\_meta.qc)),c("BCS","ExposureTime"),summarise,n=length(BCS)))

## BCS ExposureTime n  
## 1 LCRISPATUS 4HR 2  
## 2 LCRISPATUS 13HR 2  
## 3 LCRISPATUS 22HR 3  
## 4 LJENSENII 4HR 2  
## 5 LJENSENII 13HR 3  
## 6 LJENSENII 22HR 3  
## 7 LINERS 4HR 2  
## 8 LINERS 13HR 3  
## 9 LINERS 22HR 3  
## 10 GVAGINALIS 4HR 3  
## 11 GVAGINALIS 13HR 3  
## 12 GVAGINALIS 22HR 2  
## 13 MEDIA 4HR 2  
## 14 MEDIA 13HR 3  
## 15 MEDIA 22HR 2

## Create model matrix

This is done outside edgeR GLM as it is used by other chunks, but edgeR chunk is not evaluated

## take counts table design created above and make edgeR object  
design <-model.matrix(~0+group1,data = pData(TRL\_counts\_meta.qc))

## edgeR GLM FIT

Not evualted as it takes some time. Change to eval=T to re-compute edgeR results

## Estimate dispersion, calc norm facots, and fit to GLM  
y<-DGEList(exprs(TRL\_counts\_meta.qc))  
y<-estimateDisp(y,design)  
y<-calcNormFactors(y)  
plotBCV(y,main="BCV Plot")  
TRL\_glmFit <- glmFit(y, design)  
  
save(TRL\_glmFit,file = paste0(R\_script\_output\_directory,"TRL\_glmFit.RData"))

## Perfom edgeR LRT

load(paste0(R\_script\_output\_directory,"TRL\_glmFit.RData"))  
## Go through each pait-wise comparison in constarsts and compute differential expression using glmLRT  
contr<-makeContrasts(  
 LCGV.4="group1LCRISPATUS.4HR-group1GVAGINALIS.4HR",  
 LCM.4="group1LCRISPATUS.4HR-group1MEDIA.4HR",  
 LJGV.4="group1LJENSENII.4HR-group1GVAGINALIS.4HR",  
 LJM.4="group1LJENSENII.4HR-group1MEDIA.4HR",  
 LIGV.4="group1LINERS.4HR-group1GVAGINALIS.4HR",  
 LIM.4="group1LINERS.4HR-group1MEDIA.4HR",  
 LCLJ.4="group1LCRISPATUS.4HR-group1LJENSENII.4HR",  
 LCLI.4="group1LCRISPATUS.4HR-group1LINERS.4HR",  
 LJLI.4="group1LJENSENII.4HR-group1LINERS.4HR",  
 GVM.4="group1GVAGINALIS.4HR-group1MEDIA.4HR",  
   
 LCGV.13="group1LCRISPATUS.13HR-group1GVAGINALIS.13HR",  
 LCM.13="group1LCRISPATUS.13HR-group1MEDIA.13HR",  
 LJGV.13="group1LJENSENII.13HR-group1GVAGINALIS.13HR",  
 LJM.13="group1LJENSENII.13HR-group1MEDIA.13HR",  
 LIGV.13="group1LINERS.13HR-group1GVAGINALIS.13HR",  
 LIM.13="group1LINERS.13HR-group1MEDIA.13HR",  
 LCLJ.13="group1LCRISPATUS.13HR-group1LJENSENII.13HR",  
 LCLI.13="group1LCRISPATUS.13HR-group1LINERS.13HR",  
 LJLI.13="group1LJENSENII.13HR-group1LINERS.13HR",  
 GVM.13="group1GVAGINALIS.13HR-group1MEDIA.13HR",  
   
 LCGV.22="group1LCRISPATUS.22HR-group1GVAGINALIS.22HR",  
 LCM.22="group1LCRISPATUS.22HR-group1MEDIA.22HR",  
 LJGV.22="group1LJENSENII.22HR-group1GVAGINALIS.22HR",  
 LJM.22="group1LJENSENII.22HR-group1MEDIA.22HR",  
 LIGV.22="group1LINERS.22HR-group1GVAGINALIS.22HR",  
 LIM.22="group1LINERS.22HR-group1MEDIA.22HR",  
 LCLJ.22="group1LCRISPATUS.22HR-group1LJENSENII.22HR",  
 LCLI.22="group1LCRISPATUS.22HR-group1LINERS.22HR",  
 LJLI.22="group1LJENSENII.22HR-group1LINERS.22HR",  
 GVM.22="group1GVAGINALIS.22HR-group1MEDIA.22HR",  
   
 levels=design  
)  
  
desets<-list()  
comparisons<-names(data.frame(contr))  
  
#postscript(paste0(R\_script\_output\_directory,"TRL\_SmearPlots.eps"),height = 8,width = 10)  
  
for(comparison in comparisons){  
 #comparison<-"LCGV.13"  
 comp<-glmLRT(TRL\_glmFit,contrast = contr[,comparison]) ## DE using constrast  
  
 de.table<-comp$table[abs(comp$table$logFC)>1 & comp$table$logCPM>1 & p.adjust(comp$table$PValue,method = "fdr")<=0.01,]  
 comp$table$PValue.adj<-p.adjust(comp$table$PValue,method = "fdr")  
 desets[[comparison]]<-list(detags=nrow(de.table),fulltable=comp$table)  
 #plotSmear(comp,de.tags = names(de.table),main=paste0(comparison,": ",length(detags)," DE genes"))  
  
}

Upload DE table to IPA, run IPA pathway analysis, then save the pathway comparisons to file Performed March-April 2017. See thesis for version/db builds.

## Read in IPA Results

##Timecourse files are from IPA- contain pathway list and activation z values.  
  
pathway\_zscore\_files<-list.files(path = R\_script\_input\_directory, pattern = "timecourse.txt")  
pathway\_zscores<-data.frame(Canonical.Pathway="DROP")  
functions<-data.frame(Comparison="DROP",Categories="DROP",Diseases.or.Functions.Annotation="DROP",p.Value=1,Predicted.Activation.State="DROP",Activation.z.score=0,Flags="DROP", Molecules="DROP")  
  
for(tab in pathway\_zscore\_files){  
 #tab<-"LCGV\_LCM\_timecourse.txt"  
 newt<- read.table(paste0(R\_script\_input\_directory,tab),header = T,sep = "\t",skip = 1,na.strings = "N/A",stringsAsFactors = F) %>% dplyr::select(-X)  
 pathway\_zscores<-merge(newt,pathway\_zscores,all = T)  
}  
  
## Put pathway z scores into matricies and then split absed on G. vag or medium reference  
pathway\_zscores.matrix<-as.matrix(pathway\_zscores[,2:ncol(pathway\_zscores)])  
row.names(pathway\_zscores.matrix)<-pathway\_zscores$Canonical.Pathway  
pathway\_zscores.matrix<-pathway\_zscores.matrix[rowSums(abs(pathway\_zscores.matrix),na.rm = T)>0,colSums(abs(pathway\_zscores.matrix),na.rm = T)>0]  
  
pathway\_zscores.melt<-melt(pathway\_zscores,id.vars = "Canonical.Pathway")  
row.names(pathway\_zscores)<-pathway\_zscores$Canonical.Pathway  
pathway\_zscores.matrix[is.na(pathway\_zscores.matrix)]<-0  
  
## Create a design table for the pathways  
Pathways\_design<-data.frame(comparison=names(data.frame(pathway\_zscores.matrix)))  
Pathways\_design$comp<-sapply(strsplit(as.character(Pathways\_design$comparison),"\\."),function(x) x[[1]])  
Pathways\_design$ExposureTime<-sapply(strsplit(as.character(Pathways\_design$comparison),"\\."),function(x) x[[2]])  
Pathways\_design$L<-sapply(strsplit(Pathways\_design$comp,split = "\*"),function(x) paste0(x[[1]],x[[2]]))  
Pathways\_design$ref<-sapply(strsplit(Pathways\_design$comp,split = "\*"),function(x) paste0(x[[3]]))  
row.names(Pathways\_design)<-Pathways\_design$comparison  
  
## Map mapthway names to classifications  
path\_type\_map<-read.csv(paste0(R\_script\_input\_directory,"Pathway\_classification.csv"),stringsAsFactors = F)  
  
  
## Subset to look at only cell culture medium references  
medium\_comparisons<-names(data.frame(pathway\_zscores.matrix))[grepl(names(data.frame(pathway\_zscores.matrix)),pattern = "M")]  
pathway\_zscores.matrix.medium<-pathway\_zscores.matrix[,names(data.frame(pathway\_zscores.matrix)) %in% medium\_comparisons]  
  
## Summary table for pathway classification  
summary\_pathways<-cbind(data.frame(num\_cycle.p=colSums(pathway\_zscores.matrix.medium[row.names(data.frame(pathway\_zscores.matrix.medium)) %in% path\_type\_map[path\_type\_map$class=="c","pathway"],]>2)),  
 data.frame(num\_cycle.n=colSums(pathway\_zscores.matrix.medium[row.names(data.frame(pathway\_zscores.matrix.medium)) %in% path\_type\_map[path\_type\_map$class=="c","pathway"],]<(-2))),  
 data.frame(num\_immune.p=colSums(pathway\_zscores.matrix.medium[row.names(data.frame(pathway\_zscores.matrix.medium)) %in% path\_type\_map[path\_type\_map$class=="i","pathway"],]>2)),   
 data.frame(num\_immune.n=colSums(pathway\_zscores.matrix.medium[row.names(data.frame(pathway\_zscores.matrix.medium)) %in% path\_type\_map[path\_type\_map$class=="i","pathway"],]<(-2))),  
  
 data.frame(num\_immune\_pro.n=colSums(pathway\_zscores.matrix.medium[row.names(data.frame(pathway\_zscores.matrix.medium)) %in% path\_type\_map[path\_type\_map$X=="pro","pathway"],]<(-2))),  
  
 data.frame(num\_immune\_pro.p=colSums(pathway\_zscores.matrix.medium[row.names(data.frame(pathway\_zscores.matrix.medium)) %in% path\_type\_map[path\_type\_map$X=="pro","pathway"],]>2)))  
  
summary\_pathways$comparison<-row.names(summary\_pathways)  
summary\_pathways<-merge(summary\_pathways,Pathways\_design)  
  
## Write table summarizing the number of pathways above or below z score, grouped by pathway category  
write.csv(summary\_pathways,paste0(thesis\_tables\_directory,TABLE\_TRL\_SUMMARY\_PATHWAYS),row.names=F,quote=F)  
  
##Sort # of z>2 pathways by negative cycle, positive immune and negative immune  
summary\_pathways[order(summary\_pathways$num\_cycle.n,decreasing = T),]

## comparison num\_cycle.p num\_cycle.n num\_immune.p num\_immune.n  
## 4 LCM.13 1 4 2 0  
## 5 LCM.22 0 3 0 1  
## 6 LCM.4 3 2 11 0  
## 10 LJM.13 0 2 1 1  
## 11 LJM.22 0 2 1 1  
## 3 GVM.4 3 1 18 0  
## 8 LIM.22 0 1 3 0  
## 12 LJM.4 1 1 2 0  
## 1 GVM.13 1 0 8 0  
## 2 GVM.22 0 0 2 0  
## 7 LIM.13 0 0 7 0  
## 9 LIM.4 4 0 16 0  
## num\_immune\_pro.n num\_immune\_pro.p comp ExposureTime L ref  
## 4 0 1 LCM 13 LC M  
## 5 0 0 LCM 22 LC M  
## 6 0 5 LCM 4 LC M  
## 10 0 1 LJM 13 LJ M  
## 11 0 1 LJM 22 LJ M  
## 3 0 8 GVM 4 GV M  
## 8 0 1 LIM 22 LI M  
## 12 0 2 LJM 4 LJ M  
## 1 0 3 GVM 13 GV M  
## 2 0 1 GVM 22 GV M  
## 7 0 4 LIM 13 LI M  
## 9 0 7 LIM 4 LI M

summary\_pathways[order(summary\_pathways$num\_immune.p,decreasing = T),]

## comparison num\_cycle.p num\_cycle.n num\_immune.p num\_immune.n  
## 3 GVM.4 3 1 18 0  
## 9 LIM.4 4 0 16 0  
## 6 LCM.4 3 2 11 0  
## 1 GVM.13 1 0 8 0  
## 7 LIM.13 0 0 7 0  
## 8 LIM.22 0 1 3 0  
## 2 GVM.22 0 0 2 0  
## 4 LCM.13 1 4 2 0  
## 12 LJM.4 1 1 2 0  
## 10 LJM.13 0 2 1 1  
## 11 LJM.22 0 2 1 1  
## 5 LCM.22 0 3 0 1  
## num\_immune\_pro.n num\_immune\_pro.p comp ExposureTime L ref  
## 3 0 8 GVM 4 GV M  
## 9 0 7 LIM 4 LI M  
## 6 0 5 LCM 4 LC M  
## 1 0 3 GVM 13 GV M  
## 7 0 4 LIM 13 LI M  
## 8 0 1 LIM 22 LI M  
## 2 0 1 GVM 22 GV M  
## 4 0 1 LCM 13 LC M  
## 12 0 2 LJM 4 LJ M  
## 10 0 1 LJM 13 LJ M  
## 11 0 1 LJM 22 LJ M  
## 5 0 0 LCM 22 LC M

summary\_pathways[order(summary\_pathways$num\_immune.n,decreasing = T),]

## comparison num\_cycle.p num\_cycle.n num\_immune.p num\_immune.n  
## 5 LCM.22 0 3 0 1  
## 10 LJM.13 0 2 1 1  
## 11 LJM.22 0 2 1 1  
## 1 GVM.13 1 0 8 0  
## 2 GVM.22 0 0 2 0  
## 3 GVM.4 3 1 18 0  
## 4 LCM.13 1 4 2 0  
## 6 LCM.4 3 2 11 0  
## 7 LIM.13 0 0 7 0  
## 8 LIM.22 0 1 3 0  
## 9 LIM.4 4 0 16 0  
## 12 LJM.4 1 1 2 0  
## num\_immune\_pro.n num\_immune\_pro.p comp ExposureTime L ref  
## 5 0 0 LCM 22 LC M  
## 10 0 1 LJM 13 LJ M  
## 11 0 1 LJM 22 LJ M  
## 1 0 3 GVM 13 GV M  
## 2 0 1 GVM 22 GV M  
## 3 0 8 GVM 4 GV M  
## 4 0 1 LCM 13 LC M  
## 6 0 5 LCM 4 LC M  
## 7 0 4 LIM 13 LI M  
## 8 0 1 LIM 22 LI M  
## 9 0 7 LIM 4 LI M  
## 12 0 2 LJM 4 LJ M

## Use pathways that are expressed abs(z-score)>2 in 10% of comparisons   
pathway\_zscores.matrix.medium<-pathway\_zscores.matrix.medium[(rowSums(abs(pathway\_zscores.matrix.medium)>2))>=.1\*ncol(pathway\_zscores.matrix.medium),]  
  
## Clean up some of the pathway names in pathway map  
path\_type\_map[path\_type\_map$pathway=="NF-\_B Signaling","pathway"]<-"NF-κB Signaling"  
path\_type\_map[path\_type\_map$pathway=="PKC\_ Signaling in T Lymphocytes","pathway"]<-"PKCθ Signaling in T Lymphocytes"  
  
path\_type\_map[path\_type\_map$X=="pro","class"]<-"pro"  
  
## Add in pathway classification based on mapping file  
pathway\_zscores.matrix.medium<-data.frame(pathway\_zscores.matrix.medium)  
pathway\_zscores.matrix.medium$pathway<-row.names(pathway\_zscores.matrix.medium)  
pathway\_zscores.matrix.medium<-merge(pathway\_zscores.matrix.medium,path\_type\_map) ## maps classifications to pathway names  
row.names(pathway\_zscores.matrix.medium)<-pathway\_zscores.matrix.medium$pathway  
  
table(path\_type\_map[path\_type\_map$pathway %in% row.names(data.frame(pathway\_zscores.matrix.medium)),"class"])

##   
## c i o pro   
## 11 12 5 7

## Melt the pathway z-scores table  
pathway\_zscores.medium.melt<-melt(pathway\_zscores.matrix.medium,id.vars = c("class","X","pathway"))  
pathway\_zscores.medium.melt<-dplyr::rename(pathway\_zscores.medium.melt,"comparison"=variable)  
pathway\_zscores.medium.melt<-merge(pathway\_zscores.medium.melt,Pathways\_design)  
  
## Ensure the exposure times and BCS are ordered  
summary\_pathways$ExposureTime<-as.numeric(summary\_pathways$ExposureTime)  
summary\_pathways$ExposureTime<-factor(summary\_pathways$ExposureTime,levels = c(4,13,22),ordered = T)  
pathway\_zscores.medium.melt$comp<-factor(pathway\_zscores.medium.melt$comp,levels = c("LCM","LJM","LIM","GVM"),ordered = T)  
pathway\_zscores.medium.melt$ExposureTime<-factor(pathway\_zscores.medium.melt$ExposureTime,levels = c("4","13","22"),ordered = T)  
pathway\_zscores.medium.melt$x<-paste(pathway\_zscores.medium.melt$comp,pathway\_zscores.medium.melt$pathway)  
  
##Clean up the pathway names for better plotting  
pathway\_zscores.medium.melt[pathway\_zscores.medium.melt$pathway=="Role of IL-17F in Allergic Inflammatory Airway Diseases","pathway"]<-"IL-17F in Allgc. Inflam. Arwy Dis."  
pathway\_zscores.medium.melt[pathway\_zscores.medium.melt$pathway=="Role of Pattern Recognition Receptors in Recognition of Bacteria and Viruses","pathway"]<-"PRRs/ Bacteria and Viruses"  
pathway\_zscores.medium.melt[pathway\_zscores.medium.melt$pathway=="Production of Nitric Oxide and Reactive Oxygen Species in Macrophages","pathway"]<-"Production of NO and ROS in Macrophages"  
pathway\_zscores.medium.melt[pathway\_zscores.medium.melt$pathway=="PKCθ Signaling in T Lymphocytes","pathway"]<-"PKCθ Signaling"  
pathway\_zscores.medium.melt[pathway\_zscores.medium.melt$pathway=="PI3K Signaling in B Lymphocytes","pathway"]<-"PI3K Signaling"  
  
##Subset z-scores table by immune (proinflammatory) pathways or cell cycle pathways  
pathway\_zscores.medium.melt.immune<-dplyr::filter(pathway\_zscores.medium.melt,class %in% c("pro", "i"))  
pathway\_zscores.medium.melt.cycle<-dplyr::filter(pathway\_zscores.medium.melt,class %in% c("c"))  
  
##Write figures  
postscript(paste0(thesis\_figures\_directory,FIGURE\_COMBINED\_PATHWAYS\_IMMUNE,".ps"),width=10,height=8)  
  
unique(dplyr::select(pathway\_zscores.medium.melt,c(class,pathway)))

## class pathway  
## 1 pro Acute Phase Response Signaling  
## 2 c Antioxidant Action of Vitamin C  
## 3 c ATM Signaling  
## 4 pro B Cell Receptor Signaling  
## 5 c Cholecystokinin/Gastrin-mediated Signaling  
## 6 c Colorectal Cancer Metastasis Signaling  
## 7 c Cyclins and Cell Cycle Regulation  
## 8 c Death Receptor Signaling  
## 9 i Dendritic Cell Maturation  
## 10 c Estrogen-mediated S-phase Entry  
## 11 pro HMGB1 Signaling  
## 12 pro IL-1 Signaling  
## 13 i IL-6 Signaling  
## 14 i IL-8 Signaling  
## 15 c ILK Signaling  
## 16 i iNOS Signaling  
## 17 i Interferon Signaling  
## 18 o LXR/RXR Activation  
## 19 pro MIF Regulation of Innate Immunity  
## 20 c Mitotic Roles of Polo-Like Kinase  
## 21 i NF-κB Signaling  
## 22 o NRF2-mediated Oxidative Stress Response  
## 23 i p38 MAPK Signaling  
## 24 i PI3K Signaling  
## 25 i PKCθ Signaling  
## 26 o PPAR Signaling  
## 27 i Production of NO and ROS in Macrophages  
## 28 c RANK Signaling in Osteoclasts  
## 29 c Role of BRCA1 in DNA Damage Response  
## 30 i IL-17F in Allgc. Inflam. Arwy Dis.  
## 31 pro PRRs/ Bacteria and Viruses  
## 32 pro Toll-like Receptor Signaling  
## 33 i TREM1 Signaling  
## 34 o Type I Diabetes Mellitus Signaling  
## 35 o UVA-Induced MAPK Signaling

paste0("Number of pathways belonging to each class:")

## [1] "Number of pathways belonging to each class:"

table(unique(dplyr::select(pathway\_zscores.medium.melt,c(class,pathway))) %>% dplyr::select(class))

##   
## c i o pro   
## 11 12 5 7

paste0(c("the remaining 5 pathways did not belong to either cell cycle or immunity:",str\_c(unique(dplyr::select(pathway\_zscores.medium.melt,c(class,pathway))) %>% dplyr::filter(class=="o") %>% dplyr::select(pathway),collapse = ", ")))

## [1] "the remaining 5 pathways did not belong to either cell cycle or immunity:"   
## [2] "c(\"LXR/RXR Activation\", \"NRF2-mediated Oxidative Stress Response\", \"PPAR Signaling\", \"Type I Diabetes Mellitus Signaling\", \"UVA-Induced MAPK Signaling\")"

ggplot(pathway\_zscores.medium.melt.immune)+geom\_tile(aes(x=ExposureTime,y=pathway,fill=value))+  
scale\_fill\_gradient2(high="red",mid="white",low="blue",   
 na.value="yellow", midpoint=0)+facet\_wrap(~comp,nrow=1)+  
 #mBio+  
 theme\_bw() + theme(text = element\_text(colour = "black",size=12))  
 #theme(text = element\_text(size=12),plot.margin = unit(c(0,30,0,30),units = "pt"))  
  
dev.off()

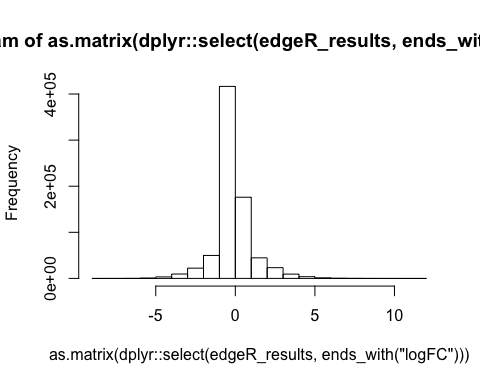
## quartz\_off\_screen   
## 2

cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_COMBINED\_PATHWAYS\_CYCLE,".eps"),width=10,height=8)  
  
ggplot(pathway\_zscores.medium.melt.cycle)+geom\_tile(aes(x=ExposureTime,y=pathway,fill=value))+  
scale\_fill\_gradient2(high="red",mid="white",low="blue",   
 na.value="yellow", midpoint=0)+facet\_wrap(~comp,nrow=1)+  
 mBio+  
 theme(text = element\_text(size=12),plot.margin = unit(c(0,30,0,0),units = "pt"))  
  
dev.off()

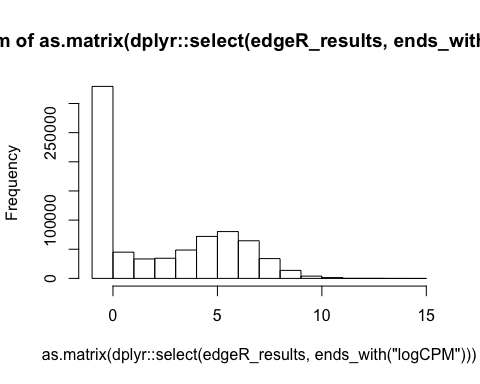
## quartz\_off\_screen   
## 2

## Extract and plot logFC values from LRT table

## Pull out DE tables from each comparison within list, and write this table to file  
edgeR\_results<-data.frame(gene=row.names(exprs(TRL\_counts\_meta.qc)))  
row.names(edgeR\_results)<-edgeR\_results$gene  
  
num\_de\_genes<-data.frame(num\_de\_genes=sapply(desets,function(x) x[[1]]))   
num\_de\_genes$comparison<-row.names(num\_de\_genes)  
write.csv(num\_de\_genes[num\_de\_genes$comparison %in% medium\_comparisons,],file=paste0(m=thesis\_tables\_directory,TABLE\_TRL\_NUMDEGENES))  
  
##Exctract genes from the list  
for(i in names(desets)){  
 #i<-"LCGV.4"  
 tmp.df<-data.frame(desets[[i]]$fulltable)  
 names(tmp.df)<-paste0(i,".",names(tmp.df))  
 tmp.df$gene<-row.names(desets[[i]]$fulltable)  
 edgeR\_results<-join(edgeR\_results,tmp.df,"gene")  
}  
  
hist(as.matrix(dplyr::select(edgeR\_results,ends\_with("logFC"))))



hist(as.matrix(dplyr::select(edgeR\_results,ends\_with("logCPM"))))



quantile(as.matrix(dplyr::select(edgeR\_results,ends\_with("logFC"))),probs = c(0.68,0.95))

## 68% 95%   
## 0.04733638 2.00227280

edgeR\_results[1:5,1:6]

## gene LCGV.4.logFC LCGV.4.logCPM LCGV.4.LR LCGV.4.PValue  
## 1 A1BG 0.1939510 2.4093604 0.1224118 0.7264333  
## 2 A1BG-AS1 -0.3227594 3.4936307 0.9826982 0.3215336  
## 3 A1CF 0.0000000 -0.9245545 0.0000000 1.0000000  
## 4 A2M 0.0000000 -0.9245545 0.0000000 1.0000000  
## 5 A2M-AS1 0.6333559 -0.4253955 0.2134882 0.6440470  
## LCGV.4.PValue.adj  
## 1 1.0000000  
## 2 0.8634145  
## 3 1.0000000  
## 4 1.0000000  
## 5 1.0000000

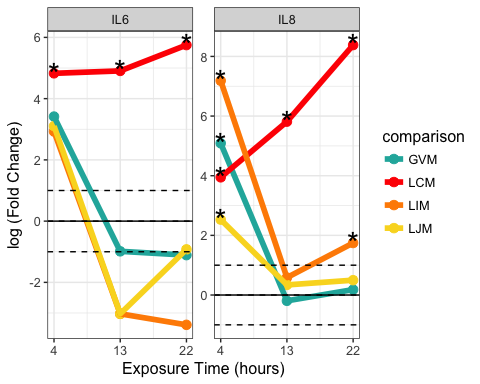
row.names(edgeR\_results)<-edgeR\_results$gene  
dplyr::select(edgeR\_results[edgeR\_results$gene %in% c("EGFR","EP300","HDAC4","CDKN1A"),],contains("M")) %>% dplyr::select(contains("PValue.adj"))

## LCM.4.PValue.adj LJM.4.PValue.adj LIM.4.PValue.adj GVM.4.PValue.adj  
## CDKN1A 7.765397e-10 7.132150e-06 5.904157e-11 0.0004304325  
## EGFR 1.000000e+00 6.757156e-01 5.802838e-01 0.4210450337  
## EP300 9.065477e-01 1.000000e+00 5.297162e-01 1.0000000000  
## HDAC4 1.000000e+00 1.000000e+00 7.308229e-01 1.0000000000  
## LCM.13.PValue.adj LJM.13.PValue.adj LIM.13.PValue.adj  
## CDKN1A 1.098515e-32 2.310890e-15 1  
## EGFR 1.073840e-09 2.314689e-05 1  
## EP300 1.352439e-05 6.697016e-02 1  
## HDAC4 1.539614e-06 9.621443e-02 1  
## GVM.13.PValue.adj LCM.22.PValue.adj LJM.22.PValue.adj  
## CDKN1A 0.7364812 2.587257e-21 1.154165e-02  
## EGFR 1.0000000 8.637272e-35 5.610384e-07  
## EP300 1.0000000 2.969729e-05 8.009564e-01  
## HDAC4 1.0000000 8.871445e-05 1.127586e-01  
## LIM.22.PValue.adj GVM.22.PValue.adj  
## CDKN1A 0.5303285 0.8958338  
## EGFR 0.5905420 1.0000000  
## EP300 0.7554520 1.0000000  
## HDAC4 0.5601210 1.0000000

write.csv(edgeR\_results,paste0(thesis\_tables\_directory,TABLE\_EDGER\_RESULTS),quote = F,row.names = F)  
#write.table(edgeR\_results,paste0(thesis\_tables\_directory,"TABLE\_A11.txt"),quote = F,row.names = F,sep="\t")  
#write.csv(dplyr::select(edgeR\_results,c(gene,ends\_with("logFC"),ends\_with("PValue"))),paste0(root\_directory,"edgeR\_results\_LCPval.csv"),quote = F,row.names = F)  
  
## Clean up the table containg the DE expresison information  
edgeR\_results.melt<-melt(edgeR\_results)

## Using gene as id variables

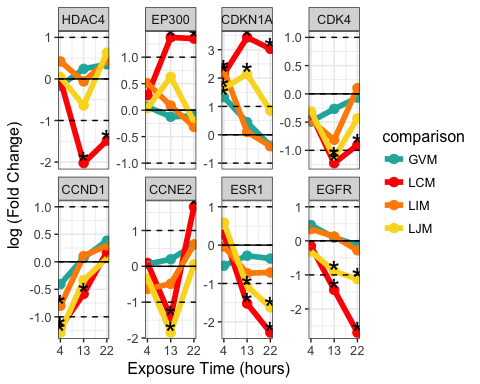
edgeR\_results.melt$variable<-gsub(pattern = "PValue.adj",replacement = "Pvalue\_adj",x = edgeR\_results.melt$variable)  
edgeR\_results.melt<-separate(edgeR\_results.melt,"variable",sep="\\.",into = c("comparison","ExposureTime","value\_type"))  
  
## Assign colors to BCS  
color\_map<-c("LCGV"=unname(subject\_long\_taxa\_colors["Lactobacillus\_crispatus"]),"LCM"=unname(subject\_long\_taxa\_colors["Lactobacillus\_crispatus"]),"LJGV"=unname(subject\_long\_taxa\_colors["Lactobacillus\_jensenii"]),"LJM"=unname(subject\_long\_taxa\_colors["Lactobacillus\_jensenii"]),"LIM"=unname(subject\_long\_taxa\_colors["Lactobacillus\_iners"]),"LIGV"=unname(subject\_long\_taxa\_colors["Lactobacillus\_iners"]),"GVM"=unname(subject\_long\_taxa\_colors["Gardnerella\_vaginalis"]))  
  
## Use different line types for G. vaginalis and medium (if used)  
line\_map<-data.frame(comparison=names(color\_map),line\_type=c("solid","dashed","solid","dashed","dashed","solid","dashed"))  
  
## Merge tables together to map DE results to plot annotations  
edgeR\_results\_colors<-data.frame(comparison=names(color\_map),colr=unname(color\_map))  
edgeR\_results.melt<-merge(edgeR\_results.melt,edgeR\_results\_colors)  
edgeR\_results.melt<-merge(edgeR\_results.melt,line\_map)  
edgeR\_results.melt<-merge(edgeR\_results.melt,dplyr::select(Pathways\_design,c(comp,L,ref)),all.x=T,by.x="comparison",by.y="comp")  
edgeR\_results.melt$ExposureTime<-as.numeric(edgeR\_results.melt$ExposureTime)  
  
  
## Format and plot the selected immune gene expression's logFC over the timecourse  
immune\_genes<-c("IL6","CXCL8")  
immune\_genes\_expression<- dplyr::filter(edgeR\_results.melt,gene %in% immune\_genes & ref=="M" ) ## Only include selected immune related genes vs. the cell culture medium reference  
immune\_genes\_expression<-unique(immune\_genes\_expression)  
immune\_genes\_expression<-spread(immune\_genes\_expression,key = value\_type,value = value) ## This will make the logFC, FDR, and other DE attributes into colums for easier plotting. The plot will use logFC and FDR information  
immune\_genes\_expression[immune\_genes\_expression$Pvalue\_adj<0.01,"DE.pval"]<-"\*" ## Annotate which samples are DE by FDR  
immune\_genes\_expression$gene<-factor(immune\_genes\_expression$gene,levels = c("IL6","CXCL8"),ordered = T,labels = c("IL6","IL8")) ## Make plot consistent  
  
##Make plot  
long\_plot.immune<-ggplot(immune\_genes\_expression)+geom\_point(aes(x=as.numeric(ExposureTime),y=logFC,col=comparison),size=3)+  
 geom\_line(aes(x=as.numeric(ExposureTime),y=logFC,col=comparison),size=2)+  
 facet\_wrap(~gene,scales = "free\_y",ncol =2)+theme\_bw()+scale\_color\_manual(values = color\_map)+  
 geom\_hline(yintercept =0)+geom\_hline(yintercept =c(-1,0,1),lty=2)+xlab("Exposure Time (hours)")+  
 geom\_text(aes(x=as.numeric(ExposureTime),y=logFC,label=DE.pval),size=8)+  
 mBio+  
 ylab("log (Fold Change)")+scale\_x\_continuous(breaks=c(4,13,22))  
  
plot(long\_plot.immune)



cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_LONGITDUINAL\_GENEEXP.immune),width = 8,height = 6)  
plot(long\_plot.immune)  
dev.off()

## quartz\_off\_screen   
## 2

## Format and plot the cell cycle pathway-related gene expression's logFC over the timecourse  
cycle\_genes<-c("HDAC4","EP300","CDKN1A","CDK4","CCND1","CCNE2","ESR1","EGFR") ## Select which cell cycle genes to plot  
cycle\_genes\_expression<- dplyr::filter(edgeR\_results.melt,gene %in% cycle\_genes & ref=="M" ) ## Only include the cell cycle genes and cell culture medium as the reference  
cycle\_genes\_expression<-unique(cycle\_genes\_expression)  
cycle\_genes\_expression<-spread(cycle\_genes\_expression,key = value\_type,value = value) ## This will make the logFC, FDR, and other DE attributes into colums for easier plotting. The plot will use logFC and FDR information  
cycle\_genes\_expression$gene<-factor(cycle\_genes\_expression$gene,levels = cycle\_genes,ordered = T) ## Maintain order of genes- this follows logical order discussed in thesis.   
cycle\_genes\_expression[cycle\_genes\_expression$Pvalue\_adj<0.01,"DE.pval"]<-"\*" ## Annotate which samples are DE by FDR  
  
## Make plot  
long\_plot.cc<-ggplot(cycle\_genes\_expression)+geom\_point(aes(x=as.numeric(ExposureTime),y=logFC,col=comparison),size=3)+  
 geom\_line(aes(x=as.numeric(ExposureTime),y=logFC,col=comparison),size=2)+  
 facet\_wrap(~gene,scales = "free\_y",nrow =2)+theme\_bw()+scale\_color\_manual(values = color\_map)+  
 geom\_hline(yintercept =0)+geom\_hline(yintercept =c(-1,0,1),lty=2)+xlab("Exposure Time (hours)")+  
 geom\_text(aes(x=as.numeric(ExposureTime),y=logFC,label=DE.pval),size=8)+  
 mBio+  
 ylab("log (Fold Change)")+scale\_x\_continuous(breaks=c(4,13,22))  
  
plot(long\_plot.cc)



cairo\_ps(paste0(thesis\_figures\_directory,FIGURE\_LONGITDUINAL\_GENEEXP.cycle),width = 8,height = 6)  
plot(long\_plot.cc)  
dev.off()

## quartz\_off\_screen   
## 2

# End Timestamp

## Log session info  
filewritable\_time<-gsub(gsub(Sys.time(),pattern = " ",replacement = "\_"),pattern = "-|:",replacement = "")  
(sessionInfo\_latex<-toLatex(sessionInfo()))

## \begin{itemize}\raggedright  
## \item R version 3.2.3 (2015-12-10), \verb|x86\_64-apple-darwin13.4.0|  
## \item Locale: \verb|en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8|  
## \item Base packages: base, datasets, graphics, grDevices, grid,  
## methods, parallel, stats, utils  
## \item Other packages: Biobase~2.28.0, BiocGenerics~0.14.0,  
## Boruta~5.2.0, caret~6.0-73, dplyr~0.5.0, edgeR~3.10.5,  
## ggbiplot~0.55, ggplot2~2.2.1, gPCA~1.0, gplots~3.0.1,  
## gridExtra~2.2.1, lattice~0.20-35, limma~3.24.15, nlme~3.1-131,  
## plotly~4.5.6, plyr~1.8.4, psych~1.7.3.21, purrr~0.2.2,  
## randomForest~4.6-12, ranger~0.7.0, RColorBrewer~1.1-2,  
## readr~1.1.0, reshape~0.8.6, rfPermute~2.1.5, scales~0.4.1,  
## squash~1.0.7, stringr~1.2.0, tibble~1.3.0, tidyr~0.6.1,  
## tidyverse~1.1.1  
## \item Loaded via a namespace (and not attached): abind~1.4-5,  
## assertthat~0.1, backports~1.0.5, base64enc~0.1-3,  
## bitops~1.0-6, broom~0.4.2, car~2.1-4, caTools~1.17.1,  
## codetools~0.2-15, colorspace~1.3-2, DBI~0.6-1, deldir~0.1-12,  
## digest~0.6.12, evaluate~0.10, forcats~0.2.0, foreach~1.4.3,  
## foreign~0.8-67, gdata~2.17.0, goftest~1.0-3, gtable~0.2.0,  
## gtools~3.5.0, haven~1.0.0, hms~0.3, htmltools~0.3.5,  
## htmlwidgets~0.8, httr~1.2.1, iterators~1.0.8, jsonlite~1.4,  
## KernSmooth~2.23-15, knitr~1.15.1, labeling~0.3,  
## lazyeval~0.2.0, lme4~1.1-12, lubridate~1.6.0, magrittr~1.5,  
## mapdata~2.2-6, maps~3.1.1, MASS~7.3-45, Matrix~1.2-8,  
## MatrixModels~0.4-1, mgcv~1.8-17, minqa~1.2.4, mnormt~1.5-5,  
## ModelMetrics~1.1.0, modelr~0.1.0, munsell~0.4.3, nloptr~1.0.4,  
## nnet~7.3-12, pbkrtest~0.4-7, polyclip~1.6-1, quantreg~5.29,  
## R6~2.2.0, Rcpp~0.12.10, readxl~0.1.1, reshape2~1.4.2,  
## rmarkdown~1.4, rpart~4.1-10, rprojroot~1.2, rvest~0.3.2,  
## SparseM~1.76, spatstat~1.45-2, splines~3.2.3, stats4~3.2.3,  
## stringi~1.1.5, swfscMisc~1.2, tensor~1.5, tools~3.2.3,  
## viridisLite~0.2.0, xml2~1.1.1, yaml~2.1.14  
## \end{itemize}

write(sessionInfo\_latex,paste0(R\_script\_output\_directory,"R\_sessionInfo\_",filewritable\_time,".log.txt"))  
  
timestamp()

## ##------ Wed May 31 10:15:57 2017 ------##