Looking at RSEM vsg analysis results

setwd("/Users/virginiahowick/Documents/Tryps/VSG")  
library(scater)

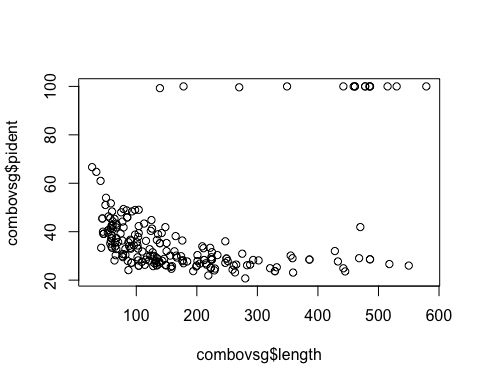
## Warning: package 'BiocGenerics' was built under R version 4.0.5

## Warning: package 'GenomeInfoDb' was built under R version 4.0.5

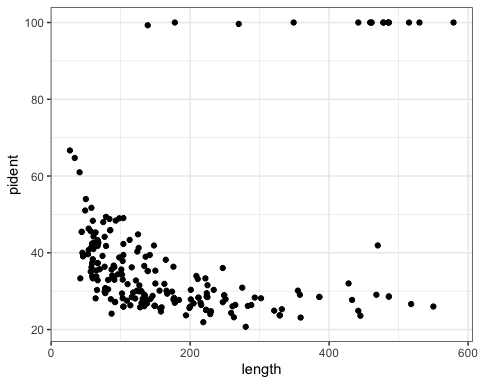
library(pheatmap)  
library(viridis)  
library(RColorBrewer)  
library(tidyverse)  
library(gplots)  
library(devtools)

tca <- readRDS("/Users/virginiahowick/Documents/Tryps/for\_github/sc\_data/howick\_tryps\_sce.rds")

counts <- read.csv("/Users/virginiahowick/Documents/Tryps/VSG/allgenes.alltotalgenes.results.csv", header=TRUE, row.names = 1)  
combovsg <- read.csv("/Users/virginiahowick/Documents/Tryps/VSG/blastpmvsg.outfmt6.csv", header=TRUE)  
plot(combovsg$length, combovsg$pident)



ggplot(combovsg, aes(x=length, y=pident)) + geom\_point() + theme\_bw()



combovsg <- droplevels(subset(combovsg, pident > 90))  
#combovsg <- droplevels(subset(combovsg, length > 300))  
length(unique(combovsg$gene\_id))

## [1] 13

#combovsg$for19 <- paste(combovsg$gene\_id, combovsg$isoform, sep = "\_")  
  
combovsgcounts <- counts[rownames(counts) %in% combovsg$gene\_id, ]  
  
dim(combovsgcounts)

## [1] 13 676

#sample ids were different for RSEM and need to subset to cells of interest  
cd <- as.data.frame(colData(tca))  
  
subcd <- cd[cd$rsemname %in% colnames(combovsgcounts), ]  
  
subcombovsgcounts <- combovsgcounts[, colnames(combovsgcounts) %in% cd$rsemname]  
  
subcd <- subcd[match(colnames(subcombovsgcounts), subcd$rsemname), ]  
rownames(subcd) <- subcd$rsemname  
  
#Sum counts for VSG that are on top of eachother  
subcombovsgcounts <- aggregate(subcombovsgcounts, list(Group=replace(rownames(subcombovsgcounts),rownames(subcombovsgcounts) %in% c("TRINITY\_DN13046\_c0\_g1","TRINITY\_DN13046\_c0\_g2"), "TRINITY\_DN13046\_c0\_g1g2")), sum)  
  
rownames(subcombovsgcounts) <- subcombovsgcounts$Group  
subcombovsgcounts <- subcombovsgcounts[ , 2:389]  
  
subcombovsgcounts <- aggregate(subcombovsgcounts, list(Group=replace(rownames(subcombovsgcounts),rownames(subcombovsgcounts) %in% c("TRINITY\_DN143\_c0\_g1","TRINITY\_DN143\_c0\_g2"), "TRINITY\_DN143\_c0\_g1g2")), sum)  
  
rownames(subcombovsgcounts) <- subcombovsgcounts$Group  
subcombovsgcounts <- subcombovsgcounts[ , 2:389]

#make SCE with vsg counts  
vsg <- SingleCellExperiment(assays =   
 list(  
 counts = as.matrix(subcombovsgcounts),  
 logcounts = log2(as.matrix(subcombovsgcounts) + 1)  
), colData = subcd)  
  
#filter\_genes <- apply(counts(mca[ , colData(mca)$use]), 1, function(x) length(x[x >= 1]) >= 2)  
  
CellQC <- perCellQCMetrics(vsg)  
FeatQC <- perFeatureQCMetrics(vsg)  
  
colData(vsg)$sumcomboVSG <- CellQC$sum  
colData(vsg)$detectedcomboVSG <- CellQC$detected  
  
counts <- as.data.frame(counts(vsg))  
  
#cells must have more than one read to count as expressing that VSG  
counts[counts == 1] <- 0  
  
colData(vsg)$sum\_g1\_VSG <- (as.data.frame(colSums(counts)))$`colSums(counts)`  
  
  
numcellsexp <- apply(counts, 1, function(x) length(x[x > 1]))  
numcellsexp

## TRINITY\_DN1222\_c0\_g1 TRINITY\_DN13046\_c0\_g1g2 TRINITY\_DN13046\_c1\_g2   
## 32 14 16   
## TRINITY\_DN143\_c0\_g1g2 TRINITY\_DN16022\_c0\_g1 TRINITY\_DN16709\_c0\_g1   
## 21 9 1   
## TRINITY\_DN18105\_c1\_g1 TRINITY\_DN21891\_c0\_g1 TRINITY\_DN26789\_c0\_g1   
## 7 12 1   
## TRINITY\_DN28824\_c0\_g1 TRINITY\_DN9452\_c0\_g1   
## 1 9

#add number of VSG detected to coldata  
detected\_g1\_VSG <- apply(as.data.frame(t(counts)), 1, function(x) length(x[x > 1]))  
colData(vsg)$detected\_g1\_VSG <- (as.data.frame(detected\_g1\_VSG))$detected\_g1\_VSG  
  
#add sum of total vsg to coldata  
colData(vsg)$sumlogcountsVSG <- colSums(as.data.frame(logcounts(vsg)))  
colnames(FeatQC) <- c("comboVSGmean", "comboVSGdetected")  
rowData(vsg) <- cbind(rowData(vsg), FeatQC)  
  
  
mean(vsg$sumcomboVSG)

## [1] 2579.198

mean(vsg$detectedcomboVSG)

## [1] 0.5180412

mean(vsg$sum\_g1\_VSG)

## [1] 2579

mean(vsg$detected\_g1\_VSG)

## [1] 0.3170103

vsg <- vsg[, vsg$num\_cells == "SC"]  
median(colData(vsg)$detected\_g1\_VSG)

## [1] 0

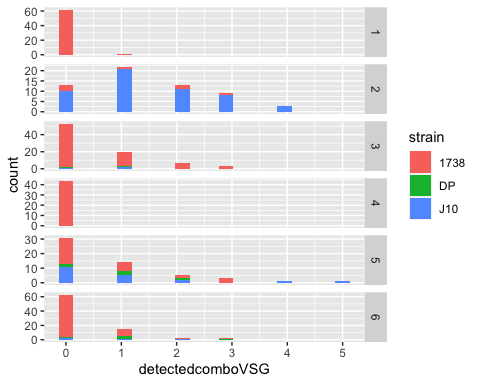
data <- as.data.frame(colData(vsg))  
  
tapply(data$sum\_g1\_VSG, data$spr\_strain, mean)

## 1738 hybrid J10   
## 595.42392 18.77778 9604.92663

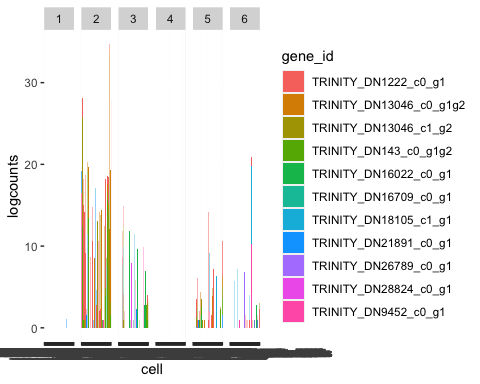
tapply(data$sum\_g1\_VSG, data$sc3\_6\_clusters, mean)

## 1 2 3 4 5 6   
## 1.634921e-02 1.615040e+04 1.325783e+02 0.000000e+00 3.330564e+02 2.777108e+01

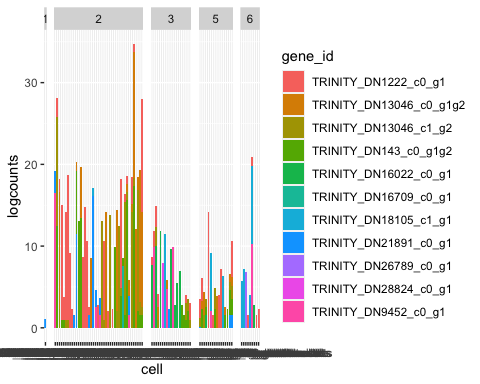
tab <- as.data.frame(colData(vsg))  
ggplot(tab, aes(x=detectedcomboVSG, fill = strain)) + geom\_histogram(bins = 20) + facet\_grid(sc3\_6\_clusters~., scales="free")



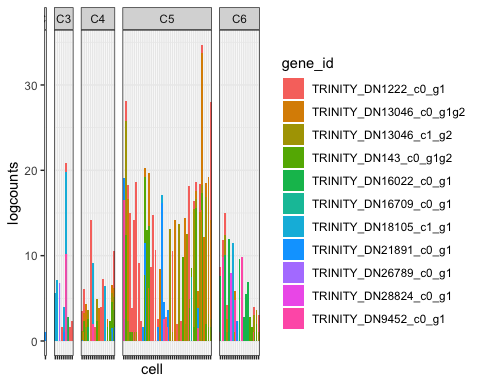
lc <- as.data.frame(logcounts(vsg))  
tlc <- as.data.frame(t(lc))  
tlc$cell <- rownames(tlc)  
  
tlc2 <- tlc %>% pivot\_longer(cols = starts\_with("TRINITY"), names\_to = "gene\_id", values\_to = "logcounts")  
tlc2$cluster <- data[match(tlc2$cell, data[, 47]), 40]  
  
#ggplot(tlc2, aes(x=cell, y=logcounts, fill = gene\_id)) + geom\_bar(position="stack", stat="identity") + facet\_grid(cluster~., scales="free")  
ggplot(tlc2, aes(x=cell, y=logcounts, fill = gene\_id)) + geom\_bar(position="stack", stat="identity") + facet\_grid(cols = vars(cluster), scales="free")



tlc2$detected\_g1\_VSG <- data[match(tlc2$cell, data[, 47]), 54]  
tlc3 <- tlc2[tlc2$detected\_g1\_VSG > 0, ]  
ggplot(tlc3, aes(x=cell, y=logcounts, fill = gene\_id)) + geom\_bar(position="stack", stat="identity") + facet\_grid(cols = vars(cluster), scales="free", space="free")



tlc3$cluster\_name <- rep("C1", length(tlc3$cell))  
tlc3[which(tlc3$cluster==4), ]$cluster\_name <- "C2"  
tlc3[which(tlc3$cluster==6), ]$cluster\_name <- "C3"  
tlc3[which(tlc3$cluster==5), ]$cluster\_name <- "C4"  
tlc3[which(tlc3$cluster==2), ]$cluster\_name <- "C5"  
tlc3[which(tlc3$cluster==3), ]$cluster\_name <- "C6"  
  
ggplot(tlc3, aes(x=cell, y=logcounts, fill = gene\_id)) + geom\_bar(position="stack", stat="identity") + facet\_grid(cols = vars(cluster\_name), scales="free", space="free") + geom\_col(position = "dodge") + theme\_bw() + theme(axis.text.x = element\_blank())



data <- as.data.frame(colData(vsg))  
data$cluster\_name <- rep("C1", length(data$sample\_id))  
data[which(data$sc3\_6\_clusters==4), ]$cluster\_name <- "C2"  
data[which(data$sc3\_6\_clusters==6), ]$cluster\_name <- "C3"  
data[which(data$sc3\_6\_clusters==5), ]$cluster\_name <- "C4"  
data[which(data$sc3\_6\_clusters==2), ]$cluster\_name <- "C5"  
data[which(data$sc3\_6\_clusters==3), ]$cluster\_name <- "C6"  
  
  
pcentFun <- function(x) {  
 res <- x > 0  
 100 \* (sum(res) / length(res))  
}  
  
with(data, tapply(data$detectedcomboVSG, cluster\_name, pcentFun))

## C1 C2 C3 C4 C5 C6   
## 1.587302 0.000000 24.096386 43.636364 78.333333 36.144578

subdata <- data[data$detected\_g1\_VSG > 0, ]  
pcentFun <- function(x) {  
 res <- x >= 1  
 100 \* (sum(res) / length(res))  
}  
with(data, tapply(data$detected\_g1\_VSG, data$cluster\_name, pcentFun))

## C1 C2 C3 C4 C5 C6   
## 1.587302 0.000000 10.843373 29.090909 70.000000 22.891566

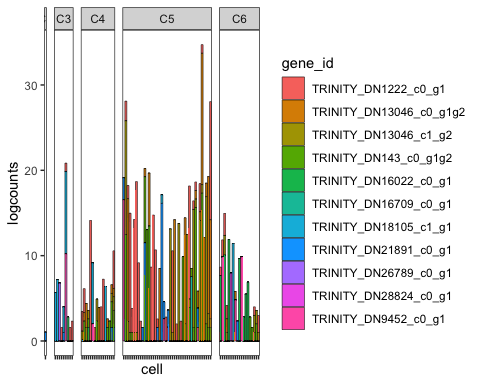
with(subdata, tapply(subdata$detected\_g1\_VSG, subdata$cluster\_name, mean))

## C1 C3 C4 C5 C6   
## 1.000000 1.111111 1.500000 1.523810 1.263158

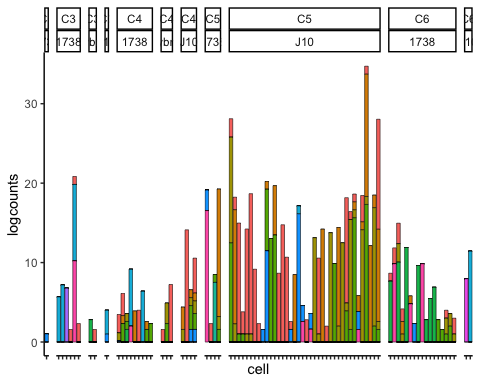
pcentFun <- function(x) {  
 res <- x >= 2  
 100 \* (sum(res) / length(res))  
}  
with(subdata, tapply(subdata$detected\_g1\_VSG, subdata$cluster\_name, pcentFun))

## C1 C3 C4 C5 C6   
## 0.00000 11.11111 37.50000 38.09524 21.05263

g1 <- subdata$rsemname  
  
tlc4 <- tlc3[tlc3$cell %in% g1, ]  
  
ggplot(tlc4, aes(x=cell, y=logcounts, fill = gene\_id)) + geom\_bar(position="stack", stat="identity", colour="black", size=0.2) + facet\_grid(cols = vars(cluster\_name), scales="free", space="free") + geom\_col(position = "dodge") + theme\_bw() + theme(axis.text.x = element\_blank(), panel.grid.major = element\_blank(), panel.grid.minor = element\_blank())



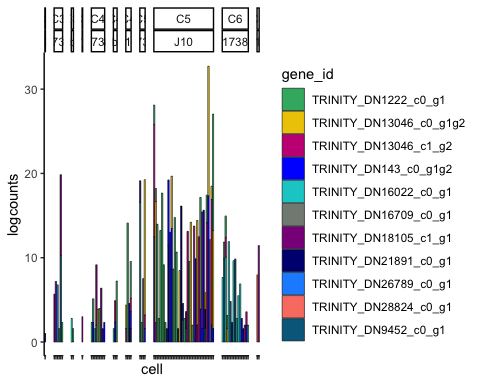
#######  
tlc4$spr\_strain <- data[match(tlc4$cell, data[, 47]), 50]  
  
ggplot(tlc4, aes(x=cell, y=logcounts, fill = gene\_id)) + geom\_bar(position="stack", stat="identity", colour="black", size=0.2) + facet\_grid(cols = vars(cluster\_name, spr\_strain), scales="free", space="free") + theme\_classic() + theme(axis.text.x = element\_blank(), panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(), legend.position = "none")



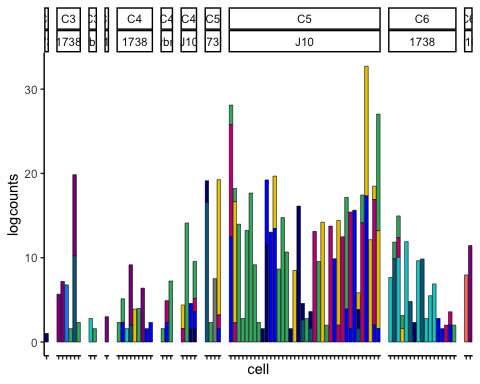
tlc5 <- tlc4[tlc4$logcounts > 1 , ]  
  
clustcol <- c("mediumseagreen", "gold2" , "mediumvioletred", "blue", "cyan3", "honeydew4", "darkmagenta","navy" , "dodgerblue", "salmon", "deepskyblue4", "purple", "thistle3")  
  
  
col2hex(c("mediumseagreen", "gold2" , "mediumvioletred", "blue", "cyan3", "honeydew4", "darkmagenta","navy" , "dodgerblue", "salmon", "deepskyblue4"))

## [1] "#3CB371" "#EEC900" "#C71585" "#0000FF" "#00CDCD" "#838B83" "#8B008B"  
## [8] "#000080" "#1E90FF" "#FA8072" "#00688B"

ggplot(tlc5, aes(x=cell, y=logcounts, fill = gene\_id)) + geom\_bar(position="stack", stat="identity", colour="black", size=0.2) + facet\_grid(cols = vars(cluster\_name, spr\_strain), scales="free", space="free") + theme\_classic() + theme(axis.text.x = element\_blank(), panel.grid.major = element\_blank(), panel.grid.minor = element\_blank()) + scale\_fill\_manual(values=clustcol)



ggplot(tlc5, aes(x=cell, y=logcounts, fill = gene\_id)) + geom\_bar(position="stack", stat="identity", colour="black", size=0.2) + facet\_grid(cols = vars(cluster\_name, spr\_strain), scales="free", space="free") + theme\_classic() + theme(axis.text.x = element\_blank(), panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(), legend.position = "none") + scale\_fill\_manual(values=clustcol)



write.csv(as.data.frame(colData(tca)), file="/Users/virginiahowick/Documents/Tryps/for\_github/vsgcoldata\_forgithub.csv")  
write.csv(subcombovsgcounts, file="/Users/virginiahowick/Documents/Tryps/for\_github/fig2\_subcombovsgcounts\_forgithub.csv")  
  
session\_info()

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value   
## version R version 4.0.3 (2020-10-10)  
## os macOS Big Sur 10.16   
## system x86\_64, darwin17.0   
## ui X11   
## language (EN)   
## collate en\_GB.UTF-8   
## ctype en\_GB.UTF-8   
## tz Europe/London   
## date 2022-01-11   
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date lib source   
## assertthat 0.2.1 2019-03-21 [1] CRAN (R 4.0.2)  
## backports 1.2.1 2020-12-09 [1] CRAN (R 4.0.2)  
## beachmat 2.6.3 2020-12-12 [1] Bioconductor   
## beeswarm 0.2.3 2016-04-25 [1] CRAN (R 4.0.2)  
## Biobase \* 2.50.0 2020-10-27 [1] Bioconductor   
## BiocGenerics \* 0.36.1 2021-04-16 [1] Bioconductor   
## BiocNeighbors 1.8.2 2020-12-07 [1] Bioconductor   
## BiocParallel 1.24.1 2020-11-06 [1] Bioconductor   
## BiocSingular 1.6.0 2020-10-27 [1] Bioconductor   
## bitops 1.0-7 2021-04-24 [1] CRAN (R 4.0.2)  
## broom 0.7.4 2021-01-29 [1] CRAN (R 4.0.2)  
## callr 3.5.1 2020-10-13 [1] CRAN (R 4.0.2)  
## caTools 1.18.0 2020-01-17 [1] CRAN (R 4.0.2)  
## cellranger 1.1.0 2016-07-27 [1] CRAN (R 4.0.2)  
## cli 3.0.1 2021-07-17 [1] CRAN (R 4.0.2)  
## colorspace 2.0-2 2021-06-24 [1] CRAN (R 4.0.2)  
## crayon 1.4.1 2021-02-08 [1] CRAN (R 4.0.2)  
## DBI 1.1.0 2019-12-15 [1] CRAN (R 4.0.2)  
## dbplyr 2.1.0 2021-02-03 [1] CRAN (R 4.0.3)  
## DelayedArray 0.16.3 2021-03-24 [1] Bioconductor   
## DelayedMatrixStats 1.12.1 2020-11-24 [1] Bioconductor   
## desc 1.2.0 2018-05-01 [1] CRAN (R 4.0.2)  
## devtools \* 2.3.2 2020-09-18 [1] CRAN (R 4.0.2)  
## digest 0.6.27 2020-10-24 [1] CRAN (R 4.0.2)  
## dplyr \* 1.0.7 2021-06-18 [1] CRAN (R 4.0.2)  
## ellipsis 0.3.2 2021-04-29 [1] CRAN (R 4.0.2)  
## evaluate 0.14 2019-05-28 [1] CRAN (R 4.0.1)  
## fansi 0.5.0 2021-05-25 [1] CRAN (R 4.0.2)  
## farver 2.1.0 2021-02-28 [1] CRAN (R 4.0.2)  
## forcats \* 0.5.1 2021-01-27 [1] CRAN (R 4.0.2)  
## fs 1.5.0 2020-07-31 [1] CRAN (R 4.0.2)  
## generics 0.1.0 2020-10-31 [1] CRAN (R 4.0.2)  
## GenomeInfoDb \* 1.26.7 2021-04-08 [1] Bioconductor   
## GenomeInfoDbData 1.2.4 2020-12-11 [1] Bioconductor   
## GenomicRanges \* 1.42.0 2020-10-27 [1] Bioconductor   
## ggbeeswarm 0.6.0 2017-08-07 [1] CRAN (R 4.0.2)  
## ggplot2 \* 3.3.5 2021-06-25 [1] CRAN (R 4.0.2)  
## glue 1.4.2 2020-08-27 [1] CRAN (R 4.0.2)  
## gplots \* 3.1.1 2020-11-28 [1] CRAN (R 4.0.2)  
## gridExtra 2.3 2017-09-09 [1] CRAN (R 4.0.2)  
## gtable 0.3.0 2019-03-25 [1] CRAN (R 4.0.2)  
## gtools 3.8.2 2020-03-31 [1] CRAN (R 4.0.2)  
## haven 2.3.1 2020-06-01 [1] CRAN (R 4.0.2)  
## hms 1.0.0 2021-01-13 [1] CRAN (R 4.0.2)  
## htmltools 0.5.0 2020-06-16 [1] CRAN (R 4.0.2)  
## httr 1.4.2 2020-07-20 [1] CRAN (R 4.0.2)  
## IRanges \* 2.24.1 2020-12-12 [1] Bioconductor   
## irlba 2.3.3 2019-02-05 [1] CRAN (R 4.0.2)  
## jsonlite 1.7.2 2020-12-09 [1] CRAN (R 4.0.3)  
## KernSmooth 2.23-18 2020-10-29 [1] CRAN (R 4.0.2)  
## knitr 1.30 2020-09-22 [1] CRAN (R 4.0.2)  
## labeling 0.4.2 2020-10-20 [1] CRAN (R 4.0.2)  
## lattice 0.20-41 2020-04-02 [1] CRAN (R 4.0.3)  
## lifecycle 1.0.0 2021-02-15 [1] CRAN (R 4.0.2)  
## lubridate 1.7.9.2 2020-11-13 [1] CRAN (R 4.0.2)  
## magrittr 2.0.1 2020-11-17 [1] CRAN (R 4.0.2)  
## Matrix 1.2-18 2019-11-27 [1] CRAN (R 4.0.3)  
## MatrixGenerics \* 1.2.1 2021-01-30 [1] Bioconductor   
## matrixStats \* 0.60.0 2021-07-26 [1] CRAN (R 4.0.2)  
## memoise 1.1.0 2017-04-21 [1] CRAN (R 4.0.2)  
## modelr 0.1.8 2020-05-19 [1] CRAN (R 4.0.2)  
## munsell 0.5.0 2018-06-12 [1] CRAN (R 4.0.2)  
## pheatmap \* 1.0.12 2019-01-04 [1] CRAN (R 4.0.2)  
## pillar 1.6.2 2021-07-29 [1] CRAN (R 4.0.2)  
## pkgbuild 1.1.0 2020-07-13 [1] CRAN (R 4.0.2)  
## pkgconfig 2.0.3 2019-09-22 [1] CRAN (R 4.0.2)  
## pkgload 1.1.0 2020-05-29 [1] CRAN (R 4.0.2)  
## prettyunits 1.1.1 2020-01-24 [1] CRAN (R 4.0.2)  
## processx 3.4.5 2020-11-30 [1] CRAN (R 4.0.2)  
## ps 1.5.0 2020-12-05 [1] CRAN (R 4.0.2)  
## purrr \* 0.3.4 2020-04-17 [1] CRAN (R 4.0.2)  
## R6 2.5.0 2020-10-28 [1] CRAN (R 4.0.2)  
## RColorBrewer \* 1.1-2 2014-12-07 [1] CRAN (R 4.0.2)  
## Rcpp 1.0.7 2021-07-07 [1] CRAN (R 4.0.2)  
## RCurl 1.98-1.3 2021-03-16 [1] CRAN (R 4.0.2)  
## readr \* 1.4.0 2020-10-05 [1] CRAN (R 4.0.2)  
## readxl 1.3.1 2019-03-13 [1] CRAN (R 4.0.2)  
## remotes 2.2.0 2020-07-21 [1] CRAN (R 4.0.2)  
## reprex 1.0.0 2021-01-27 [1] CRAN (R 4.0.2)  
## rlang 0.4.11 2021-04-30 [1] CRAN (R 4.0.2)  
## rmarkdown 2.5 2020-10-21 [1] CRAN (R 4.0.3)  
## rprojroot 2.0.2 2020-11-15 [1] CRAN (R 4.0.2)  
## rstudioapi 0.13 2020-11-12 [1] CRAN (R 4.0.2)  
## rsvd 1.0.3 2020-02-17 [1] CRAN (R 4.0.2)  
## rvest 0.3.6 2020-07-25 [1] CRAN (R 4.0.2)  
## S4Vectors \* 0.28.1 2020-12-09 [1] Bioconductor   
## scales 1.1.1 2020-05-11 [1] CRAN (R 4.0.2)  
## scater \* 1.18.3 2020-11-08 [1] Bioconductor   
## scuttle 1.0.3 2020-11-23 [1] Bioconductor   
## sessioninfo 1.1.1 2018-11-05 [1] CRAN (R 4.0.2)  
## SingleCellExperiment \* 1.12.0 2020-10-27 [1] Bioconductor   
## sparseMatrixStats 1.2.0 2020-10-27 [1] Bioconductor   
## stringi 1.7.3 2021-07-16 [1] CRAN (R 4.0.2)  
## stringr \* 1.4.0 2019-02-10 [1] CRAN (R 4.0.2)  
## SummarizedExperiment \* 1.20.0 2020-10-27 [1] Bioconductor   
## testthat 3.0.0 2020-10-31 [1] CRAN (R 4.0.2)  
## tibble \* 3.1.3 2021-07-23 [1] CRAN (R 4.0.2)  
## tidyr \* 1.1.2 2020-08-27 [1] CRAN (R 4.0.2)  
## tidyselect 1.1.1 2021-04-30 [1] CRAN (R 4.0.2)  
## tidyverse \* 1.3.0 2019-11-21 [1] CRAN (R 4.0.2)  
## usethis \* 2.0.0 2020-12-10 [1] CRAN (R 4.0.2)  
## utf8 1.2.2 2021-07-24 [1] CRAN (R 4.0.2)  
## vctrs 0.3.8 2021-04-29 [1] CRAN (R 4.0.2)  
## vipor 0.4.5 2017-03-22 [1] CRAN (R 4.0.2)  
## viridis \* 0.5.1 2018-03-29 [1] CRAN (R 4.0.2)  
## viridisLite \* 0.4.0 2021-04-13 [1] CRAN (R 4.0.2)  
## withr 2.4.2 2021-04-18 [1] CRAN (R 4.0.2)  
## xfun 0.19 2020-10-30 [1] CRAN (R 4.0.2)  
## xml2 1.3.2 2020-04-23 [1] CRAN (R 4.0.2)  
## XVector 0.30.0 2020-10-28 [1] Bioconductor   
## yaml 2.2.1 2020-02-01 [1] CRAN (R 4.0.2)  
## zlibbioc 1.36.0 2020-10-28 [1] Bioconductor   
##   
## [1] /Library/Frameworks/R.framework/Versions/4.0/Resources/library