

Reanalysis

Pooja Bhat

February 27, 2018

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(DESeq2)
```

```
## Loading required package: S4Vectors
## Loading required package: stats4
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##   clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##   clusterExport, clusterMap, parApply, parCapply, parLapply,
##   parLapplyLB, parRapply, parSapply, parSapplyLB
## The following objects are masked from 'package:stats':
##
##   IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##   anyDuplicated, append, as.data.frame, cbind, colMeans,
##   colnames, colSums, do.call, duplicated, eval, evalq, Filter,
##   Find, get, grep, grepl, intersect, is.unsorted, lapply,
##   lengths, Map, mapply, match, mget, order, paste, pmax,
##   pmax.int, pmin, pmin.int, Position, rank, rbind, Reduce,
##   rowMeans, rownames, rowSums, sapply, setdiff, sort, table,
##   tapply, union, unique, unsplit, which, which.max, which.min
##
## Attaching package: 'S4Vectors'
## The following object is masked from 'package:base':
##
##   expand.grid
## Loading required package: IRanges
## Loading required package: GenomicRanges
## Warning: package 'GenomicRanges' was built under R version 3.4.1
```

```

## Loading required package: GenomeInfoDb
## Warning: package 'GenomeInfoDb' was built under R version 3.4.1
## Loading required package: SummarizedExperiment
## Warning: package 'SummarizedExperiment' was built under R version 3.4.1
## Loading required package: Biobase
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname)".
## Loading required package: DelayedArray
## Loading required package: matrixStats
##
## Attaching package: 'matrixStats'
## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians
##
## Attaching package: 'DelayedArray'
## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
## The following object is masked from 'package:base':
##
##     apply
library(plyr)

##
## Attaching package: 'plyr'
## The following object is masked from 'package:matrixStats':
##
##     count
## The following object is masked from 'package:IRanges':
##
##     desc
## The following object is masked from 'package:S4Vectors':
##
##     rename
#### reading in the count data from the replicate experiment

countData_timeCourse = list.files(path = "/Volumes/groups/amerres/Pooja/Projects/zebrafishAnnotation/sequencingData/countData_timeCourse",
countData_timeCourse = countData_timeCourse[grep("combinedFile",countData_timeCourse)]
countData_timeCourse = as.data.frame(countData_timeCourse)
countData_timeCourse$path = paste0("/Volumes/groups/amerres/Pooja/Projects/zebrafishAnnotation/sequencingData/countData_timeCourse",
countData_timeCourse_datasets = lapply(countData_timeCourse$path, function(x) read.table(x,stringsAsFactors=TRUE))

```

```

splitNames = unlist(lapply(strsplit(as.character(countData_timeCourse$countData_timeCourse), "_", T), function(x) {
countData_timeCourse$barcodes = unlist(lapply(strsplit(splitNames, ".", T), function(x) { x[1] } ))
names(countData_timeCourse_datasets) = countData_timeCourse$barcodes

### reading in the sample information file

sampleInfo = read.table("/Volumes/groups/amerres/Pooja/Projects/zebrafishAnnotation/sequencingRun_december")
countData_timeCourse_datasets_names = as.data.frame(names(countData_timeCourse_datasets))
colnames(countData_timeCourse_datasets_names) = "V2"
countData_timeCourse_datasets_names = join(countData_timeCourse_datasets_names, sampleInfo)

## Joining by: V2
names(countData_timeCourse_datasets) = as.character(countData_timeCourse_datasets_names$V3)

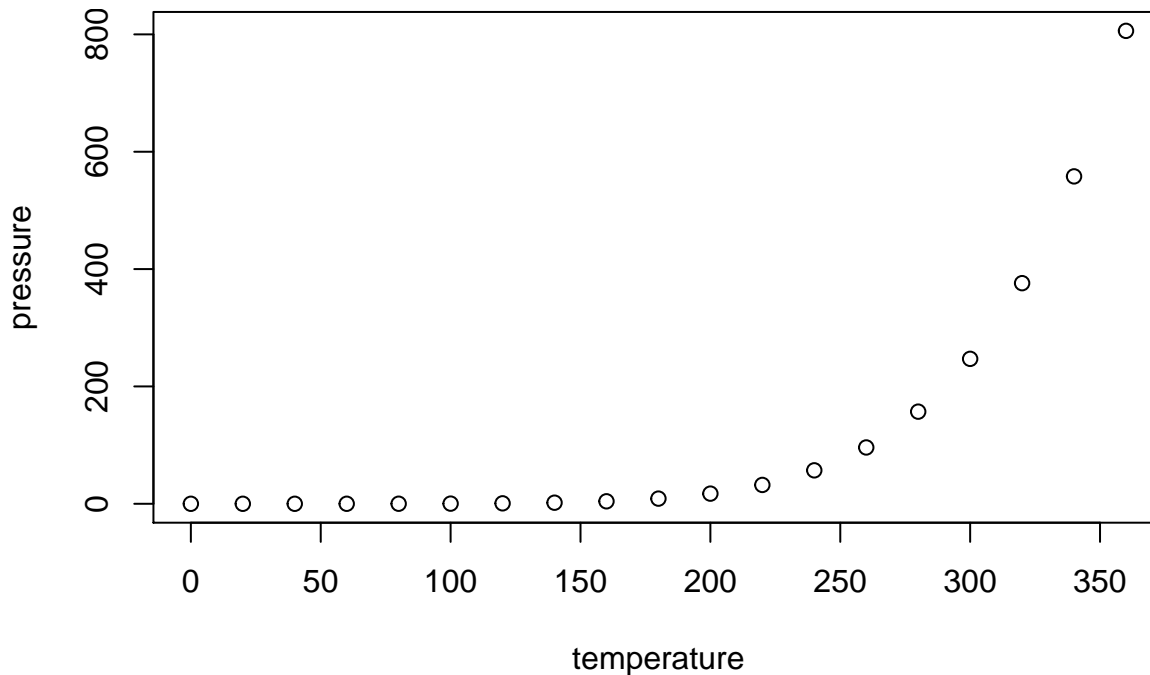
timepoints = paste0("TP", c(1:9))

for(i in 1:length(timepoints)){
  countData_timepoint = countData_timeCourse_datasets[grepl(timepoints[i], names(countData_timeCourse_datasets)), ]
  countData_timepoint_reads = do.call(cbind, lapply(countData_timepoint, function(x) { x$ReadsCPM } ))
  cor(countData_timepoint_reads, method = "spearman")
}

```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.