**[Making a histogram of Transcript lengths](http://www.scott-fay.com/blog/making-a-histogram-of-transcript-lengths-in-r)**

10/8/2012 - http://www.scott-fay.com/blog/making-a-histogram-of-transcript-lengths-in-r

[9 Comments](http://www.scott-fay.com/blog/making-a-histogram-of-transcript-lengths-in-r#comments)

Let's say we have a de novo transcriptome assembly, for example [Trinity](http://trinityrnaseq.sourceforge.net/) output, and we want to see the distribution of transcript lengths.  Let's make a histogram.  
  
For the first time we use it, we need to install the [Biostrings](http://www.bioconductor.org/packages/release/bioc/html/Biostrings.html) package in [R](http://www.r-project.org/).  Biostrings is a part of [Bioconductor](http://www.bioconductor.org/), an open-source set of libraries for working with high-throughput sequencing data.  In an R command window, type:

source("http://bioconductor.org/biocLite.R")  
 biocLite("Biostrings")

Now, load the Biostrings library.  You'll need to do this again in future R terminal instances.

library(Biostrings)

Then, import your transcripts FASTA file as a Biostrings DNAStringSet.

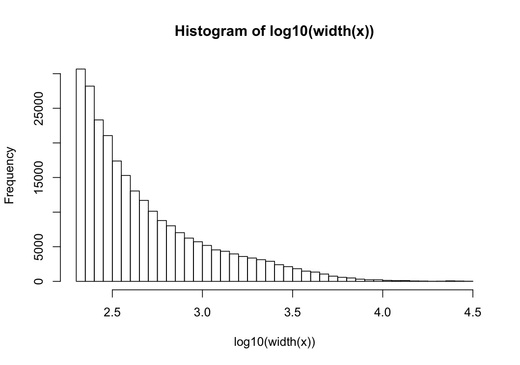
x <- read.DNAStringSet("~/someFolder/Trinity.fasta")

Then, use the generic R function hist() to generate a histogram of sequence lengths.  We generate the sequence lengths with the Biostrings width() function.

hist(width(x), breaks=50)

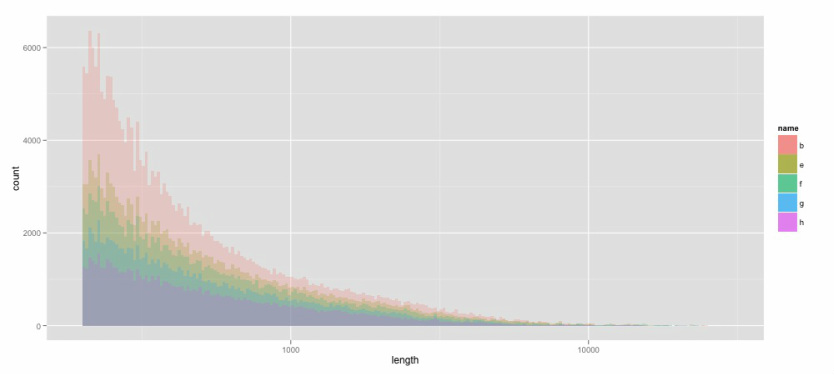
Alternately have a look at the log10 of these lengths:

hist(log10(width(x)), breaks=50)



Now, let's say you want to compare multiple transcripts to evaluate the performance of different assembly runs.  Let's use ggplot's [geom\_histogram](http://docs.ggplot2.org/current/geom_histogram.html) this time.

# load the necessary libraries  
library(Biostrings)  
library(ggplot2)  
  
# load your assemblies; in this case we will compare 5 different Trinity runs  
b <- read.DNAStringSet("~/assemblies/Trinity\_03\_B.fa")  
e <- read.DNAStringSet("~/assemblies/Trinity\_03\_E.fa")  
f <- read.DNAStringSet("~/assemblies/Trinity\_03\_F.fa")  
g <- read.DNAStringSet("~/assemblies/Trinity\_03\_G.fa")  
h <- read.DNAStringSet("~/assemblies/Trinity\_03\_H.fa")  
  
# make data frames, with the "name" of each assembly in one column   
# and the sequence "length" in the other  
b\_frame=data.frame(name="b",length=width(b))  
e\_frame=data.frame(name="e",length=width(e))  
f\_frame=data.frame(name="f",length=width(f))  
g\_frame=data.frame(name="g",length=width(g))  
h\_frame=data.frame(name="h",length=width(h))  
  
# turn these into a single data frame  
all\_frame <- rbind(b\_frame,e\_frame,f\_frame,g\_frame,h\_frame)  
  
# generate a ggplot object  
m <- ggplot(all\_frame, aes(x=length, fill=name)) + geom\_histogram(data=subset(all\_frame,name=="b"), alpha=0.2, binwidth=0.01) + scale\_x\_log10() + geom\_histogram(data=subset(all\_frame,name=="e"), alpha=0.2, binwidth=0.01) + geom\_histogram(data=subset(all\_frame,name=="f"), alpha=0.2, binwidth=0.01) + geom\_histogram(data=subset(all\_frame,name=="g"), alpha=0.2, binwidth=0.01) + geom\_histogram(data=subset(all\_frame,name=="h"), alpha=0.2, binwidth=0.01)  
  
# plot it!  
m



Okay, this gives us a good idea of the different sequence lengths that came out of the assembly.  Let's try looking at this another way by making a density histogram.

# make a density histogram  
ggplot(all\_frame, aes(length, fill=name)) + geom\_density(alpha = 0.2) + scale\_x\_log10() + theme\_bw(base\_size = 32, base\_family = "")