Combination By GC content

xiucheng quek

01 July 2016

This document is written to find the optimized parameters on alignments grouped by their gc content

Read in tables, there are 3 tables. ranked\_gc.tsv contains the cumulative score of the top parameters. combination.index.txt contains the table that maps the combination the the combination\_id in ranked.tsv. seq.index.txt containst the index for the sequence pair

For bin labels, sequences are place in the bin if they have GC content >= lower limit but < than the larger limit. Hence, 10-20 will contains sequence pair with GC >= 10 but <20.

## Dependencies

library(ggplot2)  
library(reshape2)  
library(dplyr)

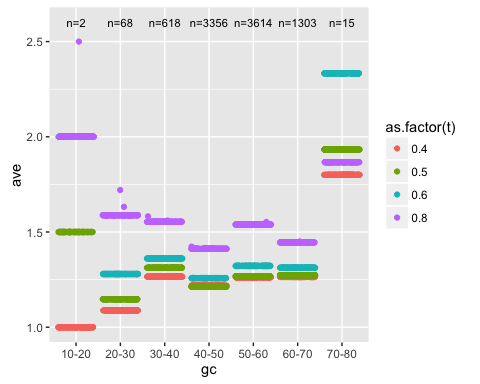
##   
## Attaching package: 'dplyr'  
##   
## The following objects are masked from 'package:stats':  
##   
## filter, lag  
##   
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

rankTable <- read.table('ranked\_gc.tsv', sep='\t', header=TRUE)  
comb <- read.table('combination.index.txt', sep='\t', header=TRUE)  
seq\_pair <- read.table('seq.index.txt', sep='\t', header=TRUE)

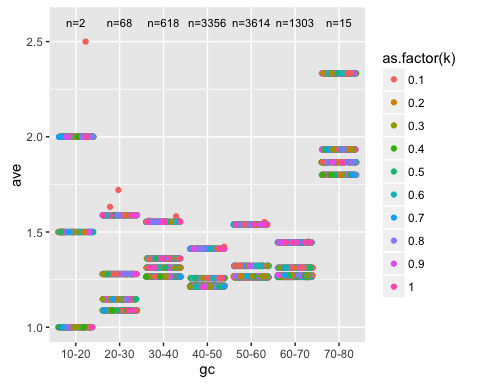
## Get the top 10 sci score

The average rank score ( y-axis, the lower the better) and the x-axis is the gc bin group. The number above represent the number of sequences in each gc bin.

mergedTable <- merge(rankTable, comb, by.x='combination', by.y='index')  
mergedTable$ave <- mergedTable$score / mergedTable$no\_of\_seq  
bin\_sizes <- dplyr::select(mergedTable, no\_of\_seq, gc) %>% distinct  
bin\_sizes$labels<- paste0('n=' , bin\_sizes$no\_of\_seq)  
ggplot(mergedTable, aes(x=gc, y=ave)) + geom\_jitter(aes(color=as.factor(t))) + geom\_text(data=bin\_sizes, aes(x=gc, y=2.6, label=labels), size=3)



ggplot(mergedTable, aes(x=gc, y=ave)) + geom\_jitter(aes(color=as.factor(k))) + geom\_text(data=bin\_sizes, aes(x=gc, y=2.6, label=labels), size=3)

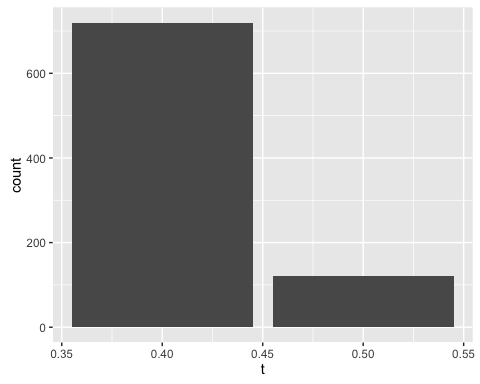


by\_gc <- dplyr::group\_by(mergedTable, gc)   
top\_scores <- dplyr::filter(by\_gc, ave == min(ave))  
top\_scores <- as.data.frame(top\_scores)  
write.table(top\_scores[order(top\_scores$gc),], 'topscore\_gc.tsv')

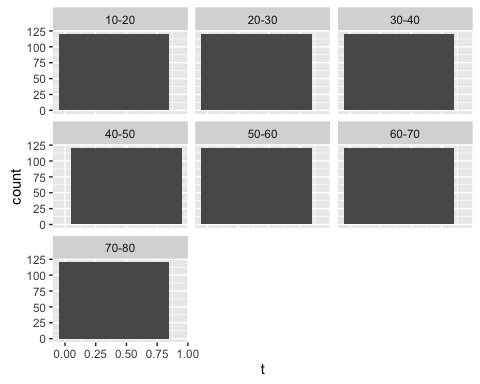
## check occurence of t and k

The only difference between the GC content is the value of t. GC content of sueqnces between 40-50 prefer to t = 0.4.

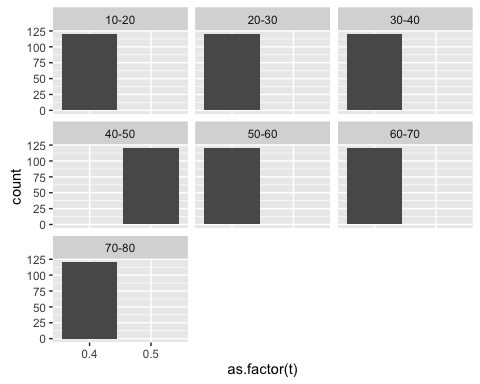
t\_k\_scores <- dplyr::select(top\_scores, gc, t, k)  
ggplot(t\_k\_scores, aes(x=t)) + geom\_bar()



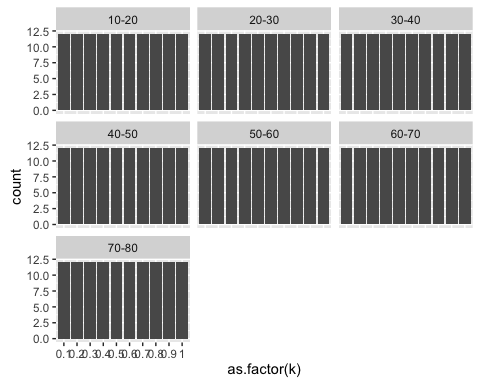
ggplot(top\_scores, aes(x=t)) + geom\_bar() + facet\_wrap(~gc)



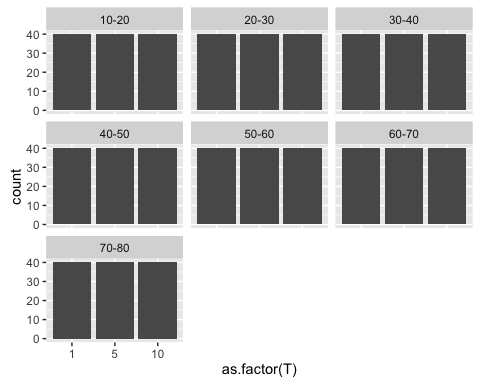
ggplot(top\_scores, aes(x=as.factor(t))) + geom\_bar() + facet\_wrap(~gc)



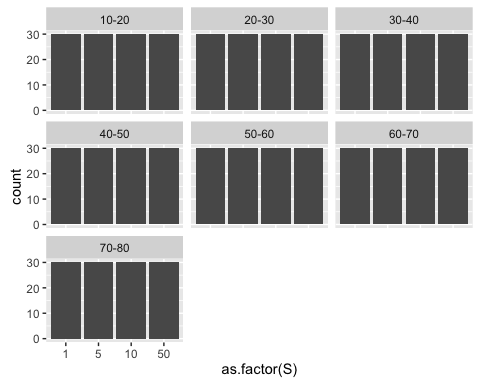
ggplot(top\_scores, aes(x=as.factor(k))) + geom\_bar() + facet\_wrap(~gc)



ggplot(top\_scores, aes(x=as.factor(T))) + geom\_bar() + facet\_wrap(~gc)



ggplot(top\_scores, aes(x=as.factor(S))) + geom\_bar() + facet\_wrap(~gc)



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