Sup_ghana_1

Qian Feng

2018/7/1

##Load libraries

```
library(data.table)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggfortify)
```

```
## Warning: package 'ggfortify' was built under R version 3.4.3
```

```
## Loading required package: ggplot2
```

```
library(ggplot2)
library(Rtsne)
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.4.3
```

```
library(starmie)
library(ggtree)
```

```
## Warning: package 'ggtree' was built under R version 3.4.3
```

```
## Loading required package: treeio
```

```
## Warning: package 'treeio' was built under R version 3.4.3
```

```
## ggtree v1.10.5  For help: https://guangchuangyu.github.io/ggtree
##
## If you use ggtree in published research, please cite:
## Guangchuang Yu, David Smith, Huachen Zhu, Yi Guan, Tommy Tsan-Yuk Lam. ggtree: an R package for visua
```

```
##
## Attaching package: 'ggtree'
```

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

```
library(ape)
```

```
##
## Attaching package: 'ape'

## The following object is masked from 'package:ggtree':
##
##     rotate

## The following objects are masked from 'package:treeio':
##
##     drop.tip, Nnode, Ntip
```

```
library(pheatmap)
library(proxy)
```

```
## Warning: package 'proxy' was built under R version 3.4.4

##
## Attaching package: 'proxy'

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix
```

```
library(stringr)
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.4.3
```

```
library(plotly)
```

```
##
## Attaching package: 'plotly'

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

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

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

```
cols <- c("#a6cee3","#1f78b4","#b2df8a","#33a02c","#fb9a99","#e31a1c","#fdbf6f","#ff7f00","#cab2d6","#6
```

We start by clustering the raw Pilot reads using a python script that makes use of the Usearch software suite.

```
cd
Python /Users/fengqian/Downloads/UniMelb_shared-master/project/scripts/clusterDBLa.py -o /Users/fengqia
```

###Binary Analysis Now we can investigate the isolates based on shared DBLa sequence types. We have extracted 161 isolates from file "Pilot.fasta"(35591 reads), and prveious isolate_information.csv provides the locations of some of isolates (137 isolates,less than 161. At last, 133 out of 161 isolates have location), let's add these location information.

```
isolateInformation <- fread("/Users/fengqian/Downloads/UniMelb_shared-master/ghana_isolate/ghana_isolat
                            , header=TRUE
                            , data.table = FALSE)
#Add in location information
isolatepilot <- fread("/Users/fengqian/Downloads/UniMelb_shared-master/isolateInformation.csv"
                            , header=TRUE
                            , data.table = FALSE)
isolatepilot <- isolatepilot[isolatepilot$Publication=="DayLab_Ghana_Pilot",]
isolatepilot$Isolate <- unlist(lapply(isolatepilot$Isolate
                                      , function(x) {
                                        paste("P",str_split(x,  "_")[[1]][[1]],sep="")}))
isolateInformation <- merge(isolateInformation,isolatepilot,by="Isolate",all.x=TRUE)

isolateInformation <- isolateInformation[isolateInformation$Survey=="pilot",]
isolateInformation$Location[which(isolateInformation$Location %in% NA)]="Ghana_Unknown"

otuTable <- fread("/Users/fengqian/Downloads/UniMelb_shared-master/project/OTU/pilot_upper_renamed_otuTa
                  , data.table = FALSE
                  , header=TRUE)

otuMatrix <- as.matrix(otuTable[,2:ncol(otuTable)])
rownames(otuMatrix) <- otuTable$`#OTU ID`
```

We found a total of 35566 (35566) reads in the combined dataset, which clustered into a total of 17923 (17923). Of these 11607(11607) were only seen in one isolate.

We next perform some filtering. We only investigate isolates that were found to have more than 20 DBLa types. This was found to be a sensible threshelf on having adequetly sequences an isolates VAR repetoir. Furthermore as we are interested in the realtionship between isolates we exclude the singletons from the binary analysis.

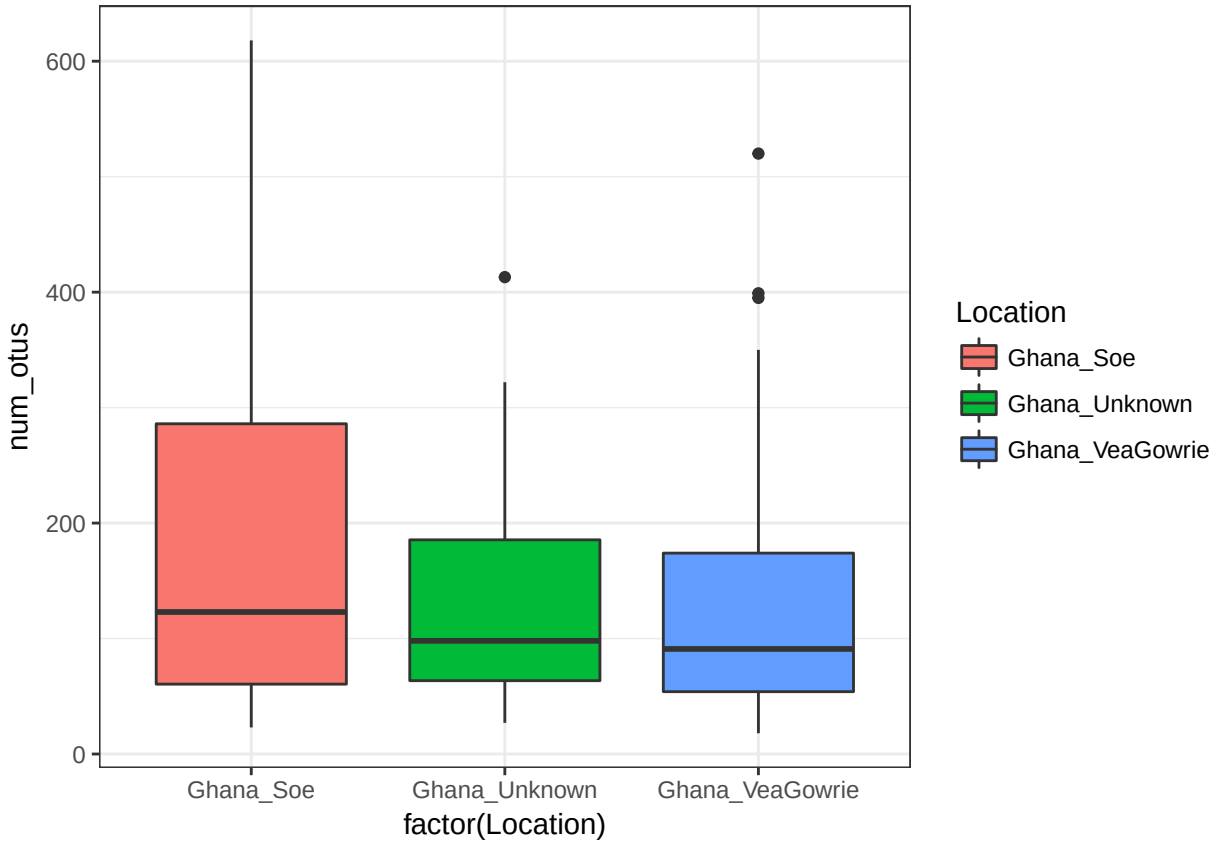
```
#Filter otus that only appear in one isolate and isolates with less than 20 types
MIN_ISOLATE_PER_OTU = 2
MIN_OTUS_PER_ISOLATE = 20
MAX_OTUS_PER_ISOLATE = Inf
otuMatrix <- otuMatrix[, colSums(otuMatrix) >= MIN_OTUS_PER_ISOLATE]
otuMatrix <- otuMatrix[, colSums(otuMatrix) <= MAX_OTUS_PER_ISOLATE]
otumatrixfiltered <- otuMatrix[rowSums(otuMatrix) >= MIN_ISOLATE_PER_OTU, ]
colnames(otumatrixfiltered) <-  unlist(lapply(colnames(otumatrixfiltered)
                                      , function(x) {
                                        str_split(x,  ".MID")[[1]][[1]]}))
```

We can now look at the number of reads per isolate for the different locations in ghana.

```
otu_sums <- data.frame(Isolate=colnames(otumatrixfiltered), num_otus=colSums(otumatrixfiltered)
                       , stringsAsFactors = FALSE)

otu_sums <- merge(otu_sums, isolateInformation, by.x='Isolate', by.y='Isolate'
                  , all.x=TRUE)

median_summary <- otu_sums %>% group_by(Location) %>%
  summarise(n=n(),
            median=median(num_otus),
            max=max(num_otus))
```

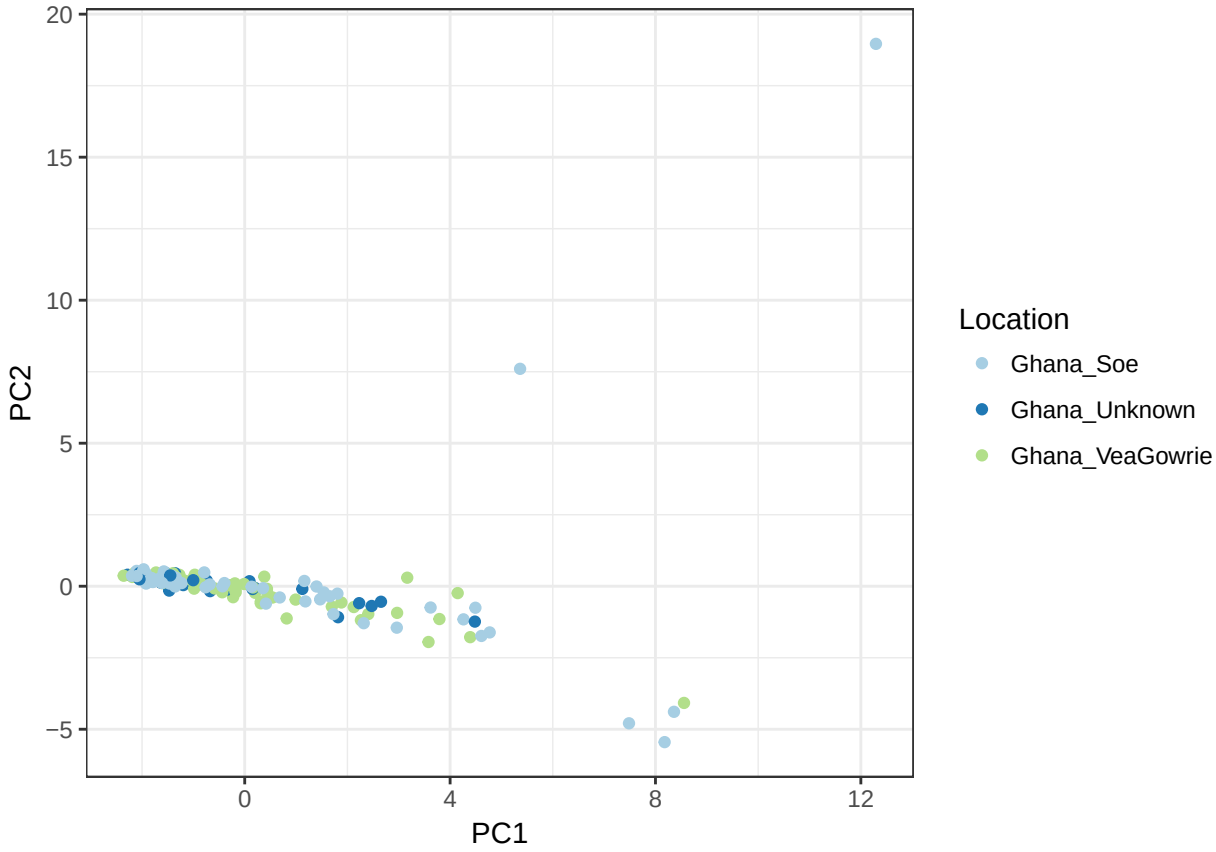
```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
#boxplot plot
gg <- ggplot(otu_sums, aes(factor(Location), num_otus, fill=Location)) + geom_boxplot()
gg <- gg + scale_color_manual(values = cols[1:length(unique(isolateInformation$Location))])
gg <- gg + theme_bw()
gg
```



###PCA

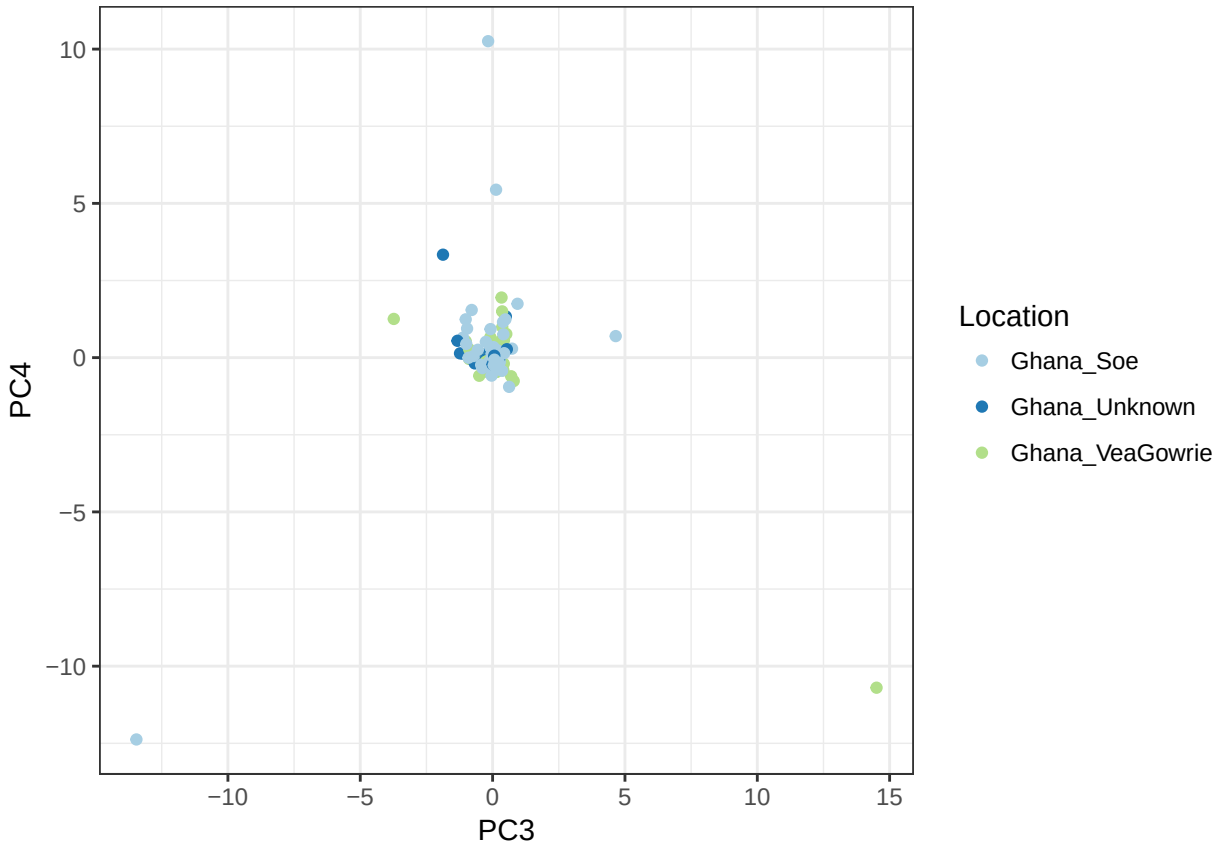
```
otuMatrixfiltered_t <- t(otumatrixfiltered)

pca <- prcomp(otuMatrixfiltered_t)

pca <- data.frame(Isolate = rownames(otuMatrixfiltered_t)
                  , pca$x[, 1:6]
                  , stringsAsFactors = FALSE)

pca <- merge(pca, isolateInformation, by.x='Isolate', by.y='Isolate'
             , all.x=TRUE)

#PCA plot
gg <- ggplot(pca, aes(PC1, PC2, colour=Location)) + geom_point()
gg <- gg + scale_color_manual(values = cols[1:length(unique(isolateInformation$Location))])
gg <- gg + theme_bw()
gg
```



Let's zoom out this PCA plot.

Also worth looking at the 3rd and 4th principal components that appear to split mainly on the different African countries.

```
#PCA plot
gg <- ggplot(pca, aes(PC3, PC4, colour=Location)) + geom_point()
gg <- gg + scale_color_manual(values = cols[1:length(unique(isolateInformation$Location))])
gg <- gg + theme_bw()
gg
```



Let's zoom out this PCA plot.

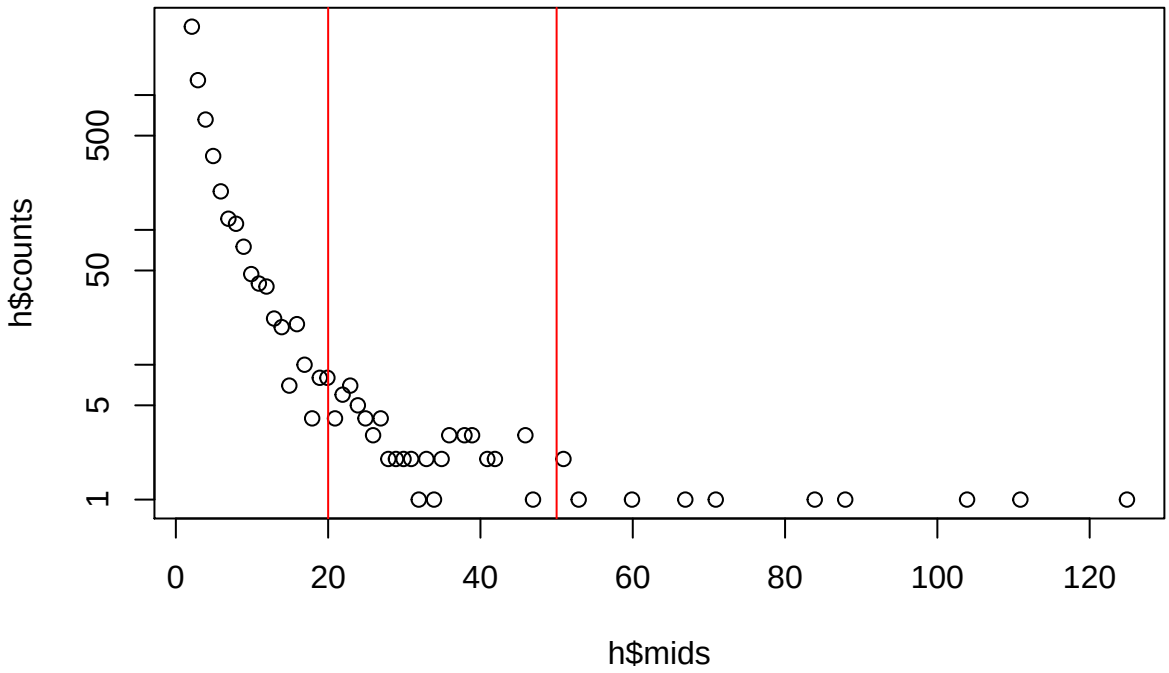
Location doesn't appear to be an issue.

We can also investigate the most conserved DBLa types. First let's look at a histogram of the number of times each DBLa is seen in the global population. It suggests that the majority of types are seen less than 20 times(6228>>83). We then take a closer look at those seen at least 20 times.

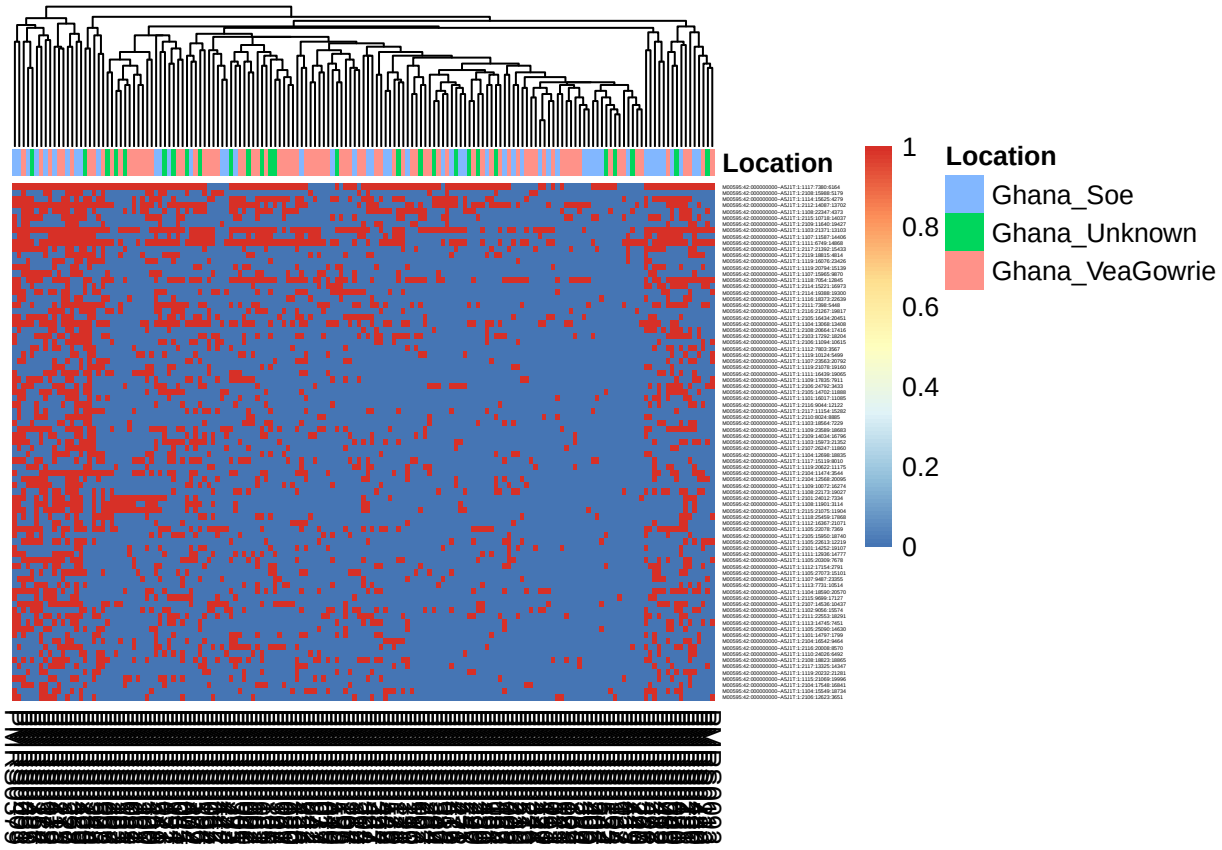
```
h <- hist(rowSums(otumatrixfiltered), breaks=500, plot = FALSE)
plot(h$mids, h$counts, log="y")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 564 y values <= 0 omitted
## from logarithmic plot
```

```
abline(v=c(20,50), col='red')
```




```
majorTypeMatrix <- otumatrixfiltered[rowSums(otumatrixfiltered)>=20,]
col_annotations <- data.frame(Isolate = colnames(majorTypeMatrix),
                              stringsAsFactors = FALSE)
col_annotations <- merge(col_annotations, isolateInformation,
                         by.x="Isolate", by.y="Isolate",
                         all.x=TRUE)
rownames(col_annotations) <- col_annotations$Isolate
col_annotations <- col_annotations[, c("Isolate","Location")]
col_annotations <- col_annotations[order(col_annotations$Location),]
majorTypeMatrix <- majorTypeMatrix[, match(col_annotations$Isolate, colnames(majorTypeMatrix))]
col_annotations$Isolate <- NULL
pheatmap(majorTypeMatrix, cluster_row = FALSE
         , annotation_col = col_annotations
         , show_rownames = TRUE
         , fontsize_row=2
         , show_colnames = TRUE)
```



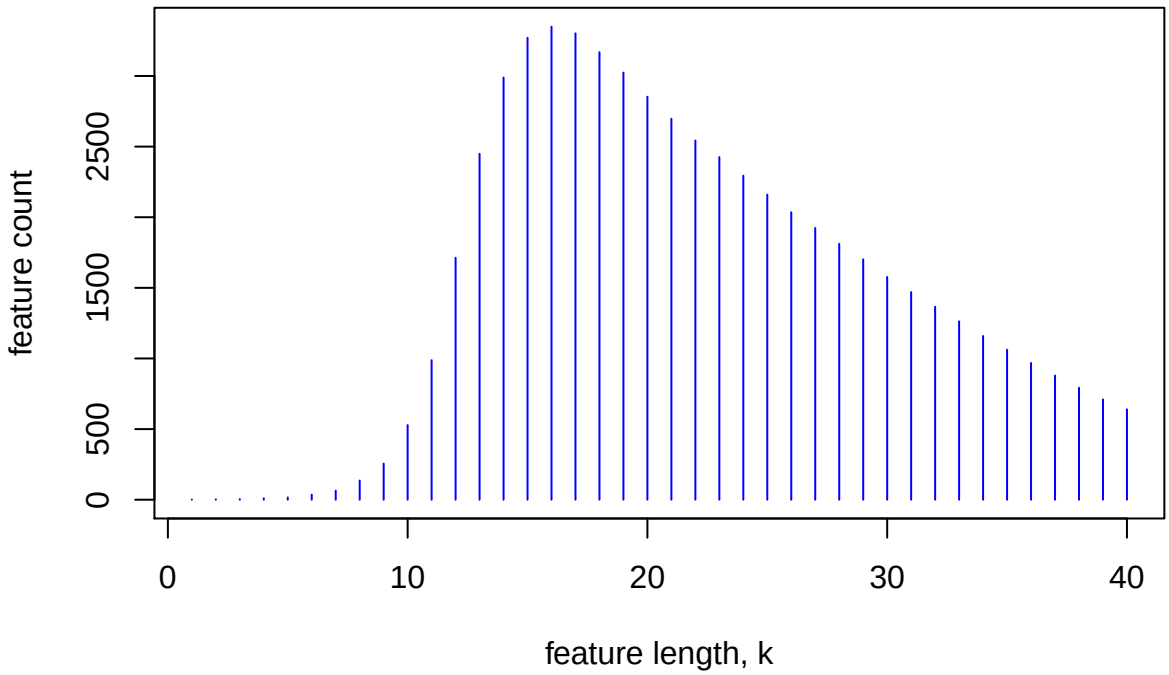
FFP

Try pilot data still. We emply the centroids P3D7.fasta to get the approariate k.

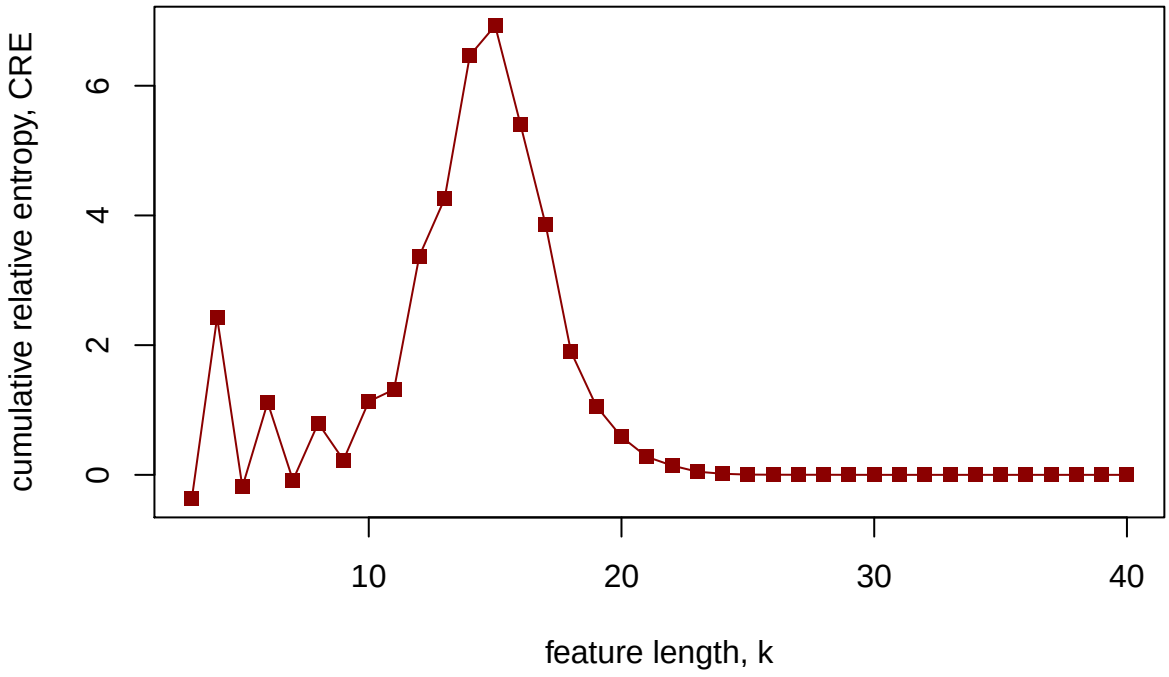
First let look at word usage to get an idea of a lower bound.The peak is in feature count distribution is our lower bound, termed as k_{hman},Then from the cumulative relative entropy curve, get the minimal k_{CREmin} is our upper bound.

We can now attempt to choose an appropriate value of k. First let look at word usage to get an idea of a lower bound.

```
word_usage <- fread("/Users/fengqian/Downloads/UniMelb_shared-master/project/ffp_data/ffp_word_usage.tx
                    data.table = FALSE)
plot(word_usage,xlab="feature length, k",ylab="feature count",type="h",col="blue")
```




```
entropy <- fread("/Users/fengqian/Downloads/UniMelb_shared-master/project/ffp_data/ffp_entropy_profile.
                 data.table = FALSE)
plot(entropy,xlab="feature length, k",ylab="cumulative relative entropy, CRE",pch=15,type="o",col="dark
```



Thus a choice of k=20 appears to be appropriate.

We can now run a script to calculate the ffp distance matrix, its output is .phylip format.

```
python /vlsci/SG0011/qian-feng/UniMelb_shared-master/project/scripts/ffp.py --kmer_length 20 --out /vls
```

Finally a tree was built using fastme v2.1.5 with default parameters. We can now have a look at the resulting tree.

```
ffp <- read.tree("/Users/fengqian/Downloads/UniMelb_shared-master/project/ffp_data/ffp_distance_matrix_
gg <- ggtree(ffp, size=0.3, branch.length = "none", layout="circular") + ggtitle("Pilot Phylogenetic tr
gg
```

Pilot Phylogenetic tree