Vignette of FSBC: Fast string-based clustering for HT-SELEX data

Shintaro Kato and Takayoshi Ono

2020-02-04

Introduction

Systematic Evolution of Ligands by EXponential enrichment (SELEX) is an experimental method for identifying aptamer sequences (Ellington and Szostak 1990, Tuerk and Gold (1990)). The combination of SELEX and Next Generation Sequencer (NGS) allows to obtain a huge number of oligonucleotide sequences from SELEX pools. This enables to search different type of aptamers for different epitopes and search aptamer hidden by oligonucleotides of PCR artifacts and/or bead binding oligonucleotides. However, it is impossible to evaluate all sequences for binding to the target molecules. Thus, clustering method is important for selecting aptamer candidates strategically from such a huge dataset. We developed fast string-based clustering (FSBC) for HT-SELEX data with R (R Core Team 2013) with bioconductor package (Gentleman et al. 2004). In this document, an example to use FSBC package for clustering with HT-SELEX dataset will be shown.

Data and Preprocessing

Sample dataset "sample.fst" includes 1000 oligonucleotide sequences without primer regions. Some oligonucleotide sequences in the data is shown as below.

```
DS <- readDNAStringSet("data/sample.fst")
DS
```

```
##   A DNAStringSet instance of length 1000
##        width seq                                                names
##    [1]    30 ATGGATGGGGGTCGGGGGTCGGGTGGGTGG                     1
##    [2]    30 GCGGGGGGTGCTAGGGCGGAGGTGGGCGTT                     2
##    [3]    30 GCGGGGGGTGCTAGGGCGGAGGTGGGCGTT                     3
##    [4]    30 TGGGGTGGGCGCAGGTGAGGGGGTGGGGGT                     4
##    [5]    30 GCGGGGGGTGCTAGGGCGGAGGTGGGCGTT                     5
##    ...   ... ...
##  [996]    30 TGGGGTGGGCGCAGGTGAGGGGGTGGGGGT                     996
##  [997]    30 GCGGGGGGTGCTAGGGCGGAGGTGGGCGTT                     997
##  [998]    30 TGGGGTGGGCGCAAGTGAGGGGGTGGGGGT                     998
##  [999]    30 TGGGGTGGGCGCAGGTGAGGGGGTGGGGGT                     999
## [1000]    30 TGGGGTGGGCACAGGTGAGGGGGTGGGGAT                     1000
```

Clustering

The following script shows the flow of FSBC to generate clusters from raw data. The flow includes calculation of frequency, calculation of nucleobase ratio, selection of over-represented strings and clustering with selected strings.

```
lmin <- 5
lmax <- 10

# Calculate frequency.
DS.freq    <- fsbc_calc_freq(DS)
# Get probability of nucleobases.
BR         <- fsbc_get_base_ratio(DS)
# Select subsequences.
DF.subseq  <- fsbc_search_subseq(DS.freq, DS.freq@metadata$freq, symbols = BR)
# Generate clusters with selected subsequences.
L.cluster  <- fsbc_seq_cluster(rownames(DF.subseq), DS.freq)
# Add cluster ID to the cluster object.
DS.cluster <- fsbc_label_cluster(L.cluster, DS.freq)
```

The following figure shows the frequency of oligonucleotide sequences.

```
plot(DS.freq@metadata$rank, DS.freq@metadata$freq, pch = 19,
         xlab = "Ranking", ylab = "Frequency")
```

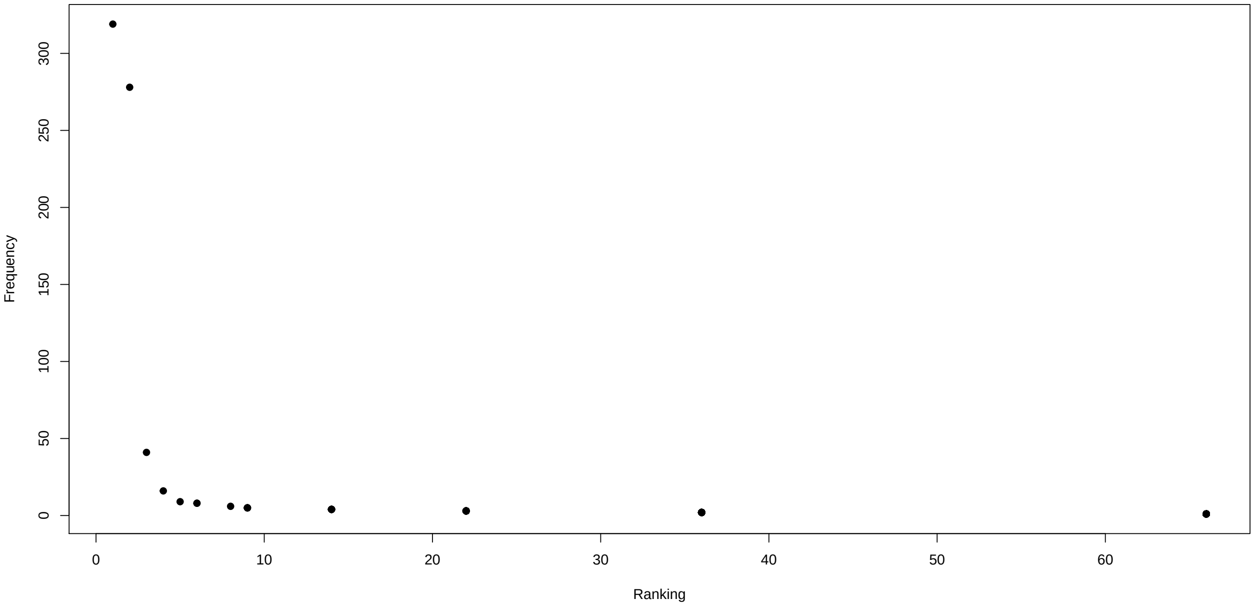


Figure 1: Frequency of oligonucleotide sequences

The following figure shows ratios of nucleobases.

```
barplot(as.vector(BR), col = 1:4, names = names(BR), ylim = c(0,1),
    main = "Nucleobase ratio")
```

Top 12 over-represented strings are shown as below. There are many G-quadruplex structure in the oligonucleotide sequences. But, the selected strings do not include such g-quadruplex sequence. Because the ratio of guanine is quite high with this data, and G-quadruplex was not estimated as not so important.

```
DF.subseq[1:12,]
```

```
##                F     R            P         Z       ZZ  L rank
## CGCAGGTGA    421 0.421 4.139253e-05 2069.1281 5.089707  9    1
## TGCTA        401 0.401 3.864192e-03  202.4186 4.431933  5    2
## CAGGTGA      436 0.436 6.791627e-04  528.4088 3.775269  7    3
```

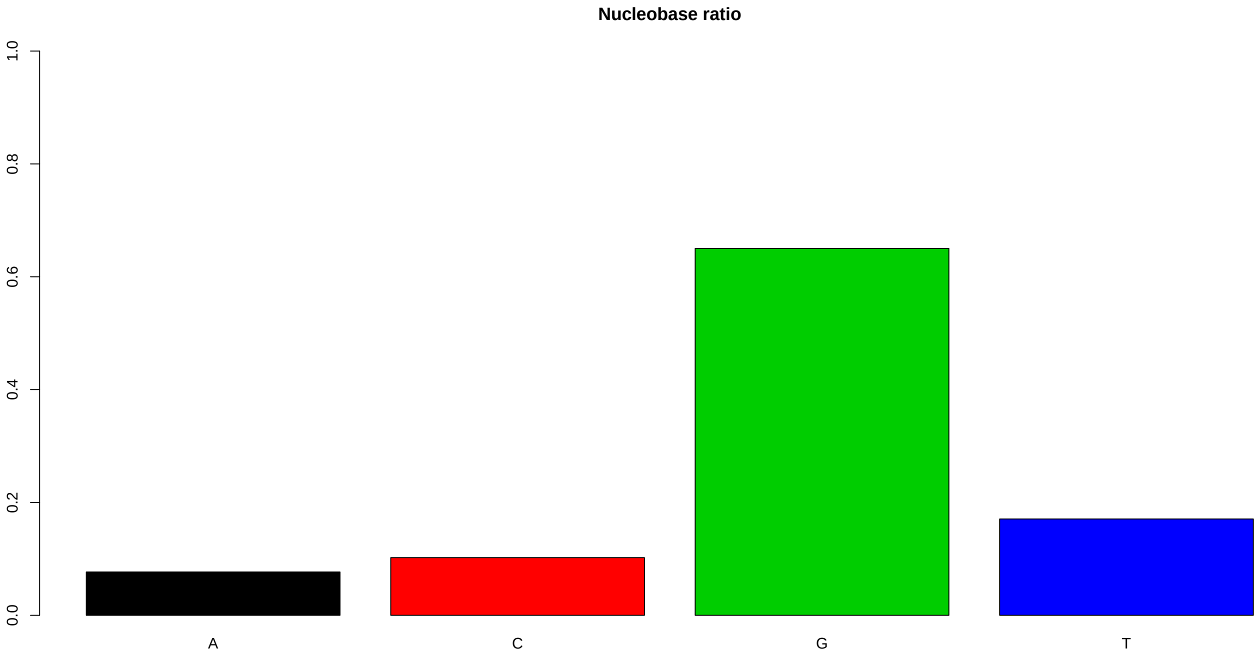


Figure 2: Nucleobase ratio

```
## CGCAGGTGAG 417 0.417 2.569151e-05 2601.4776 3.618246 10    4
## CTAGGGCGGA 414 0.414 2.569151e-05 2582.7608 3.587151 10    5
## GCGCAGGTGA 412 0.412 2.569151e-05 2570.2829 3.566422 10    6
## GTGCTA     398 0.398 2.417405e-03  254.7347 3.390683  6    7
## TGCTAG     396 0.396 2.417405e-03  253.4468 3.369902  6    8
## CGCAGGT    428 0.428 9.040224e-04  449.3992 3.105398  7    9
## CTAGGGC    427 0.427 9.037726e-04  448.4091 3.097004  7   10
## CGCAG      435 0.435 8.797275e-03  144.3314 2.979802  5   11
## GCGCA      431 0.431 8.797275e-03  142.9768 2.945938  5   12
```

The distributions of Z-score and Z^*-score are shown in the following figures.

```
par(mfrow = c(1,2))
boxplot(DF.subseq$Z  ~ DF.subseq$L, main = "Z-score")
boxplot(DF.subseq$ZZ ~ DF.subseq$L, main = "Z*-score")
```

The ratio of selected strings is shown as below.

```
all <- sum(sapply(lmin:lmax, function(i) 4^i))
nrow(DF.subseq) / all
```

```
## [1] 0.0003076351
```

```
freq <- DS.cluster@metadata$freq # Frequency of sequence
cid  <- DS.cluster@metadata$cluster.id # Cluster ranking
plot(cid, freq, pch = 19, ylab = "Frequency", xlab = "Cluster ranking", log = "x")
```

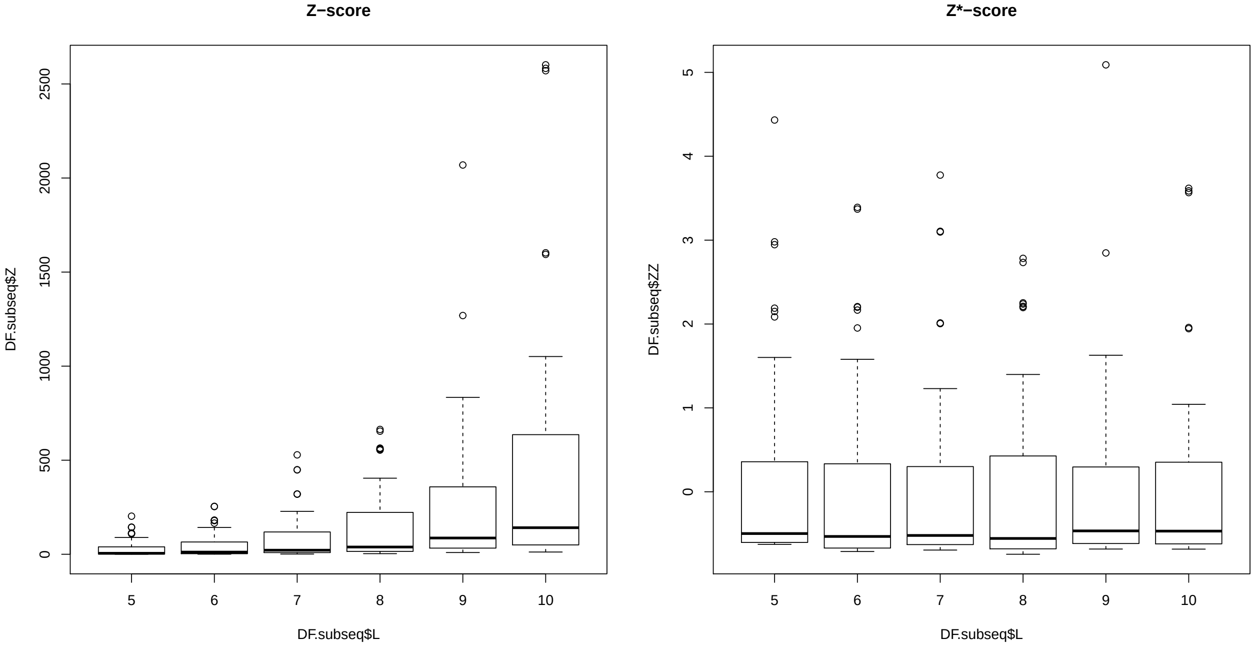
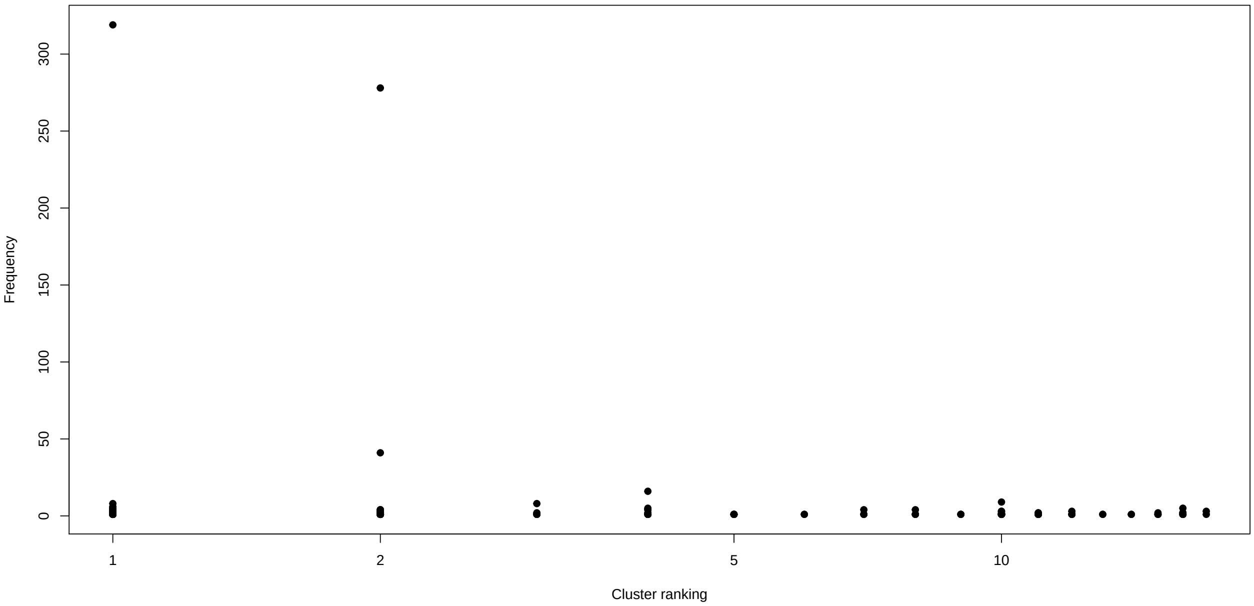


Figure 3: Distribution of Z-score and Z^*-score



The following result shows the number of sequences in each cluster.

```
cluster.div <- sapply(L.cluster, length)
l <- factor(nchar(names(cluster.div)))
bp <- barplot(cluster.div, names = 1:length(cluster.div), col = l, ylab = "Number of unique sequences")
legend("topright", title = " Length of strings ", levels(l), col = 1:nlevels(l), pch = 15)
text(bp[,1], cluster.div, names(cluster.div), pos = 3, xpd = T)
```

```
cluster.freq <- by(DS.cluster@metadata$freq, DS.cluster@metadata$cluster.id, sum)
l <- factor(nchar(names(L.cluster)))
bp <- barplot(cluster.freq, names = 1:length(cluster.freq), col = l, ylab = "Number of unique sequences
```

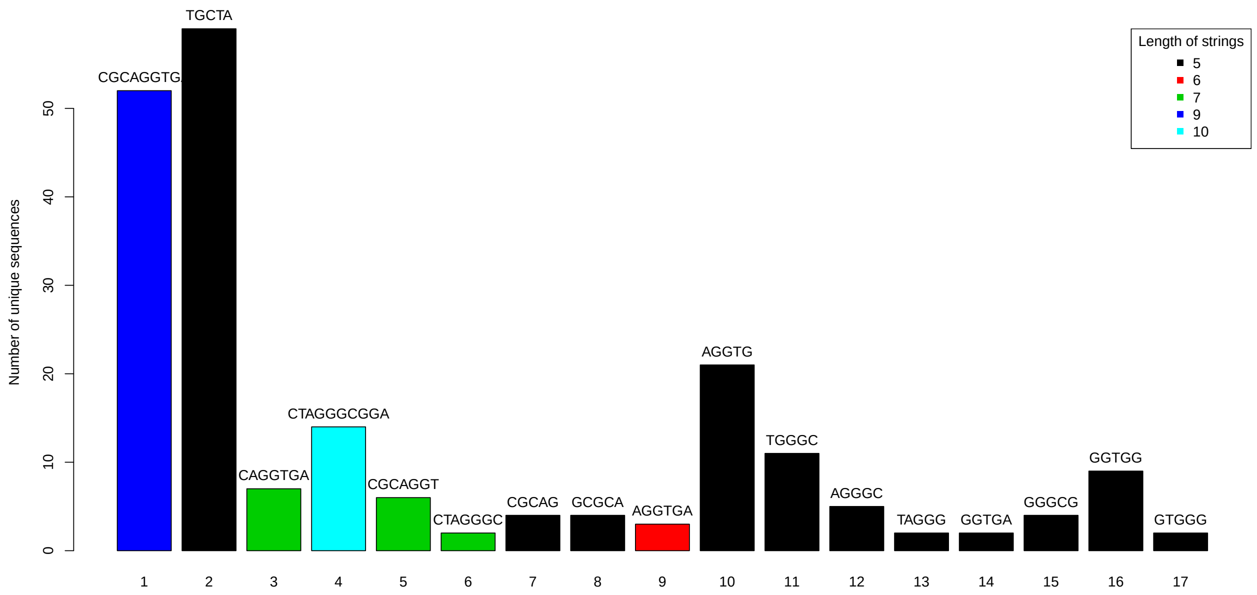
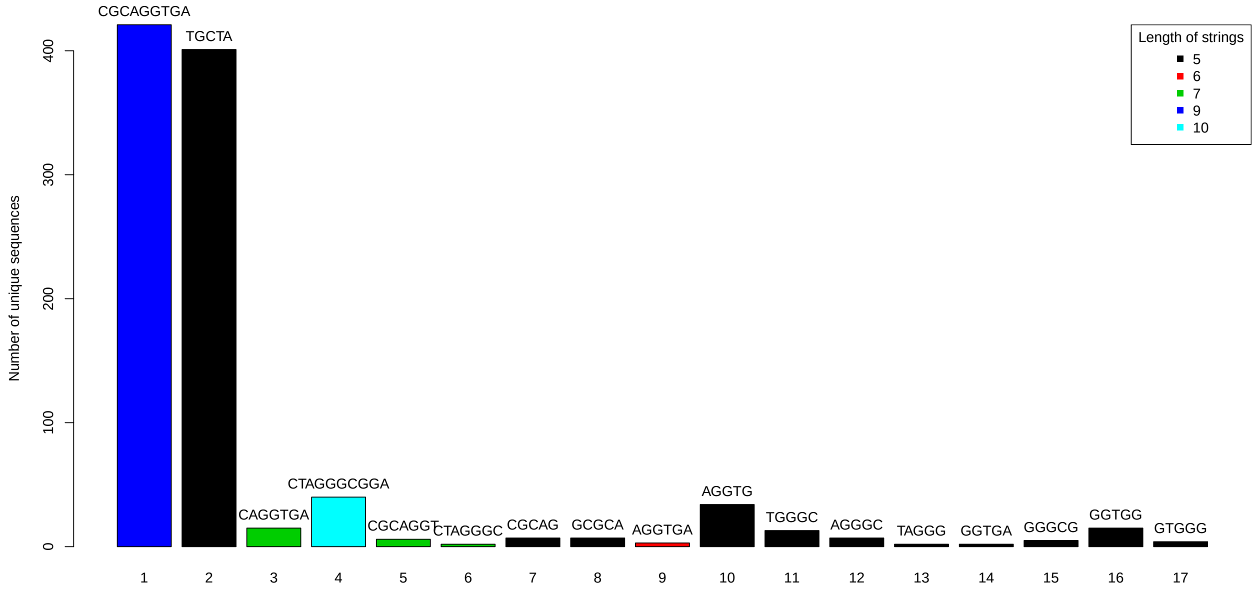


Figure 4: Clustering Results

```
legend("topright", title = " Length of strings ", levels(l), col = 1:nlevels(l), pch = 15)
text(bp[,1], cluster.freq, names(L.cluster), pos = 3, xpd = T)
```



The following result shows the sequence from top 5 clusters.

```
sapply(L.cluster, head, 1) %>% head(., 5)
```

```
## $CGCAGGTGA
##   A DNAStringSet instance of length 1
##     width seq                                                    names
## [1]    30 TGGGGTGGGCGCAGGTGAGGGGGTGGGGGT                        R:1.F:319
##
## $TGCTA
##   A DNAStringSet instance of length 1
##     width seq                                               names
## [1]    30 GCGGGGGGTGCTAGGGCGGAGGTGGGCGTT                    R:2.F:278
##
## $CAGGTGA
##   A DNAStringSet instance of length 1
##     width seq                                               names
## [1]    30 TGGGGTGGGCACAGGTGAGGGGGTGGGGGT                    R:6.F:8
##
## $CTAGGGCGGA
##   A DNAStringSet instance of length 1
##     width seq                                               names
## [1]    30 GCGGGGGGCGCTAGGGCGGAGGTGGGCGTT                    R:4.F:16
##
## $CGCAGGT
##   A DNAStringSet instance of length 1
##     width seq                                               names
## [1]    30 TGGGGTGGGCGCAGGTAAGGGGATGGGGGT                    R:66.F:1
```

Reference

Ellington, Andrew D, and Jack W Szostak. 1990. "In Vitro Selection of Rna Molecules That Bind Specific Ligands." *Nature* 346 (6287). Nature Publishing Group: 818.

Gentleman, Robert C, Vincent J Carey, Douglas M Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, et al. 2004. "Bioconductor: Open Software Development for Computational Biology and Bioinformatics." *Genome Biology* 5 (10). BioMed Central: R80.

R Core Team. 2013. *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/.

Tuerk, Craig, and Larry Gold. 1990. "Systematic Evolution of Ligands by Exponential Enrichment: RNA Ligands to Bacteriophage T4 Dna Polymerase." *Science* 249 (4968). JSTOR: 505–10.