Sequence Analysis

Liliana Patricia Calderón Bernal
Gonzalo Daniel Garcia
Ainhoa-Elena Leger

14/4/2021

Load necessary packages.

```r
# Call TraMineR library
library(TraMineR)
# Call other required libraries
library(ggplot2)
library(grDevices)
library(graphics)
library(foreign)
library(cluster)
library(Hmisc)
library(TraMineRextras)
library(WeightedCluster)
library(RColorBrewer)
library(colorspace)
```

Exercise 1

1) Input the Dataset 2

[Sol.]

```r
data2 <- read.csv("SFS2018_Data2.csv", na.strings=c(".",".a",".b"))
```

2) Define a sequence object with elements in data columns 2:61 and alphabet 1:6, using the following state names and labels

   1 SNP "Single, childless",
   2 SBP "Single, child b/separat.",
   3 SAP "Single, child a/separat.",
   4 UNP "Union, childless",
   5 UBP "Union, child b/separat.",
   6 UAP "Union, child a/separat."

[Sol.]

```r
# Create a vector for the state labels
seqlab <-c("Single, childless",
           "Single, child b/separat.",
           "Single, child a/separat.",
           "Union, childless",
           "Union, child b/separat.",
           "Union, child a/separat.")
```

```r
# Create a vector of short state names (default would be alphabet labels)
sllist <- c("SNP","SBP","SAP","UNP", "UBP", "UAP")
# Define Color palette
color1 <-  sequential_hcl(6, palette = "SunsetDark", rev= TRUE)
###  Generate sequence object
seqObj2 <- seqdef(data2,
                  var=2:61,
                  alphabet=c(1:6),
                  cpal=color1,
                  states=sllist,
                  labels=seqlab)
### Retrieve information from sequence object
summary(seqObj2)
names(seqObj2)
```

3) Display (print) the first 10 sequences in extended and compact form

[Sol.]

Extended form:

```r
#display the first 5 sequences, and sequence elements 1-20 (STS format - default).
print(seqObj2[1:10, ], format ="STS")
```

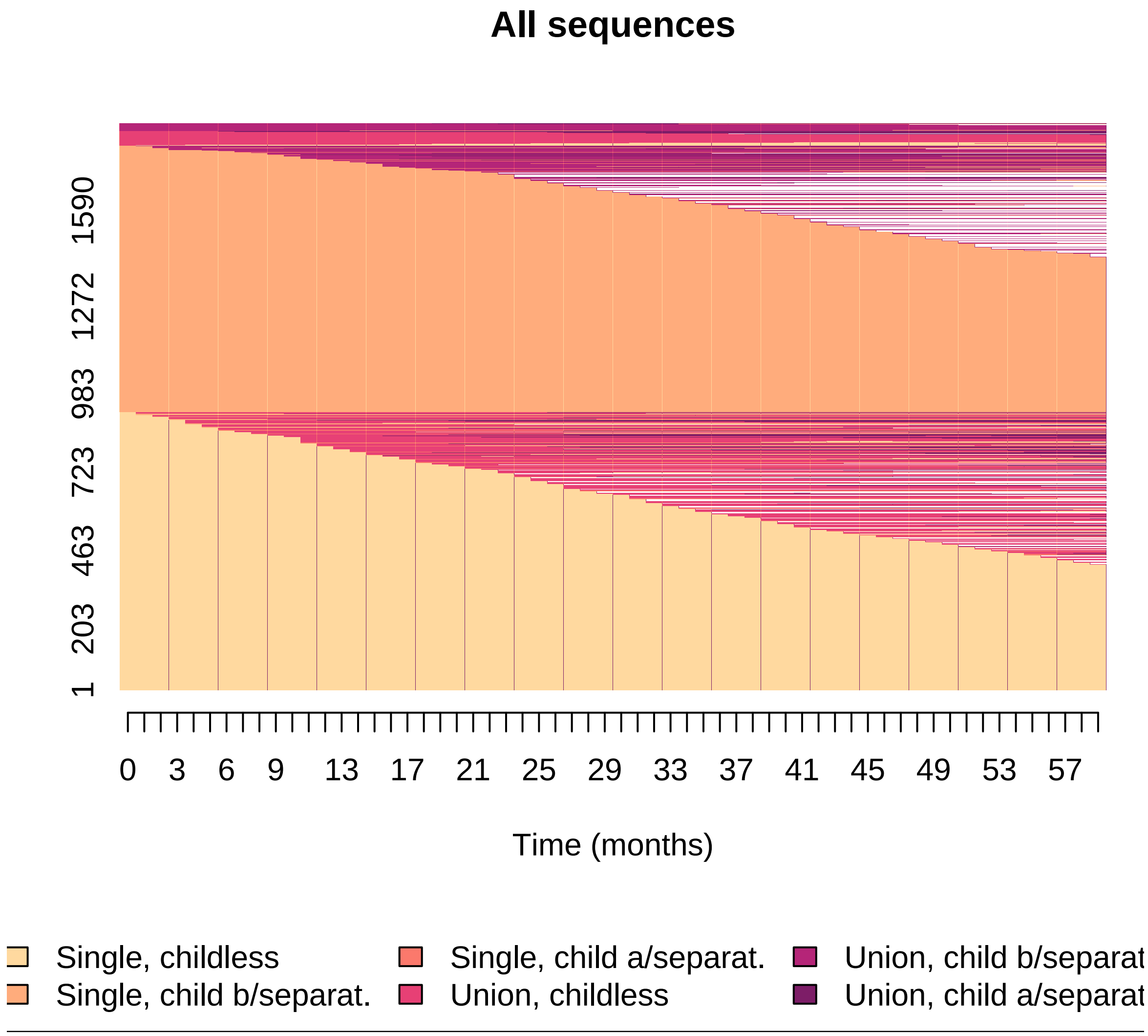
Compact form:

```r
#display the first 5 sequences, and sequence elements 1-20 (SPS format)
print(seqObj2[1:10, ], format ="SPS")
```

4) Plot a full representation of sequences, and order them from the first state

[Sol.]

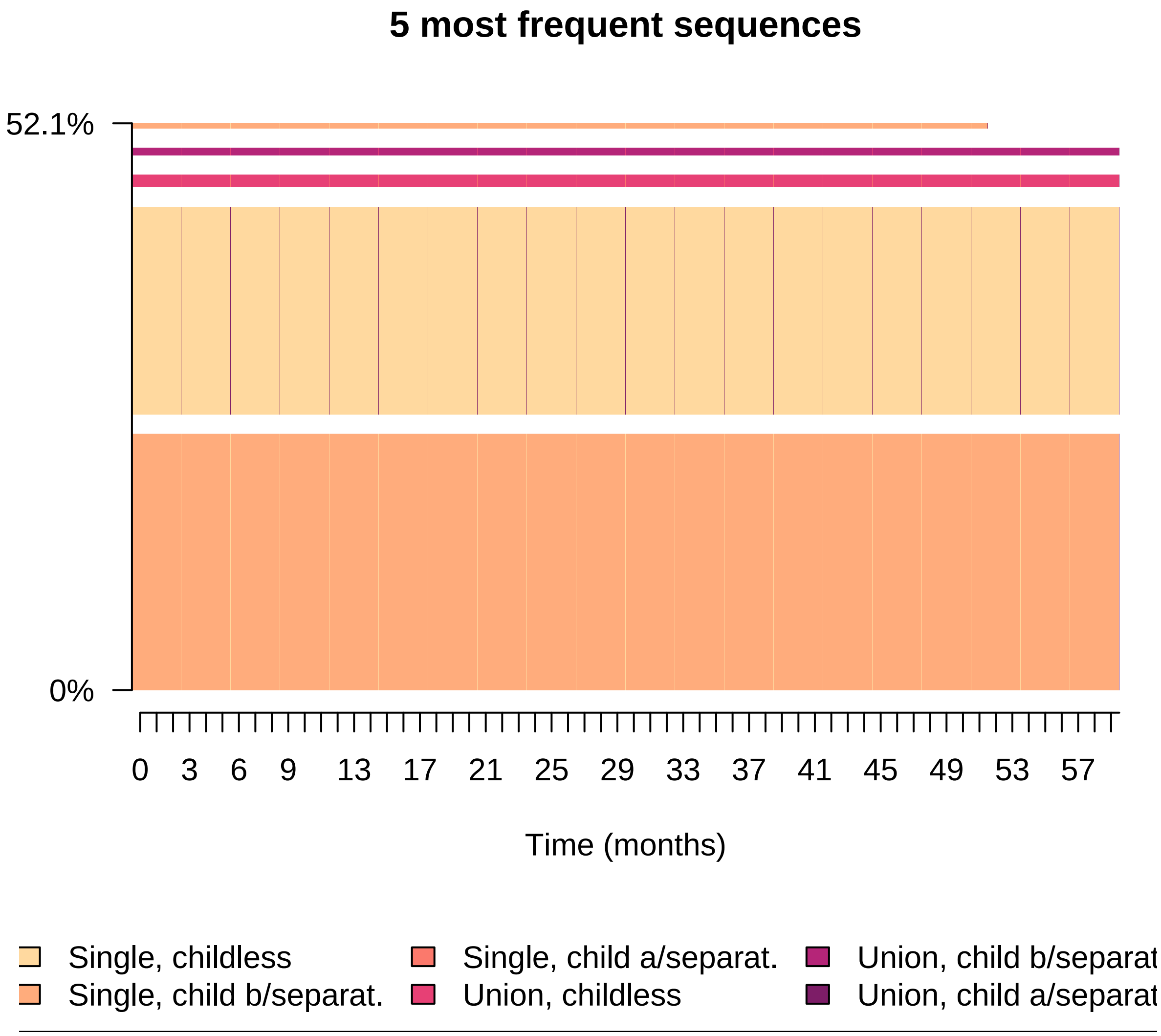
```r
# X-axis for exercise
xtlab=seq(0,60, by=1)
#All sequences -sequence index plot (sorted - first state)
par(mfrow=c(2,1))
seqIplot(seqObj2, with.legend=TRUE, main= "All sequences",
         xtlab=xtlab, xlab="Time (months)", ylab=NA, yaxis=TRUE,
         border=NA, sortv="from.start")
```



5) Plot the 5 most frequent sequences. Comment the plot

[Sol.]

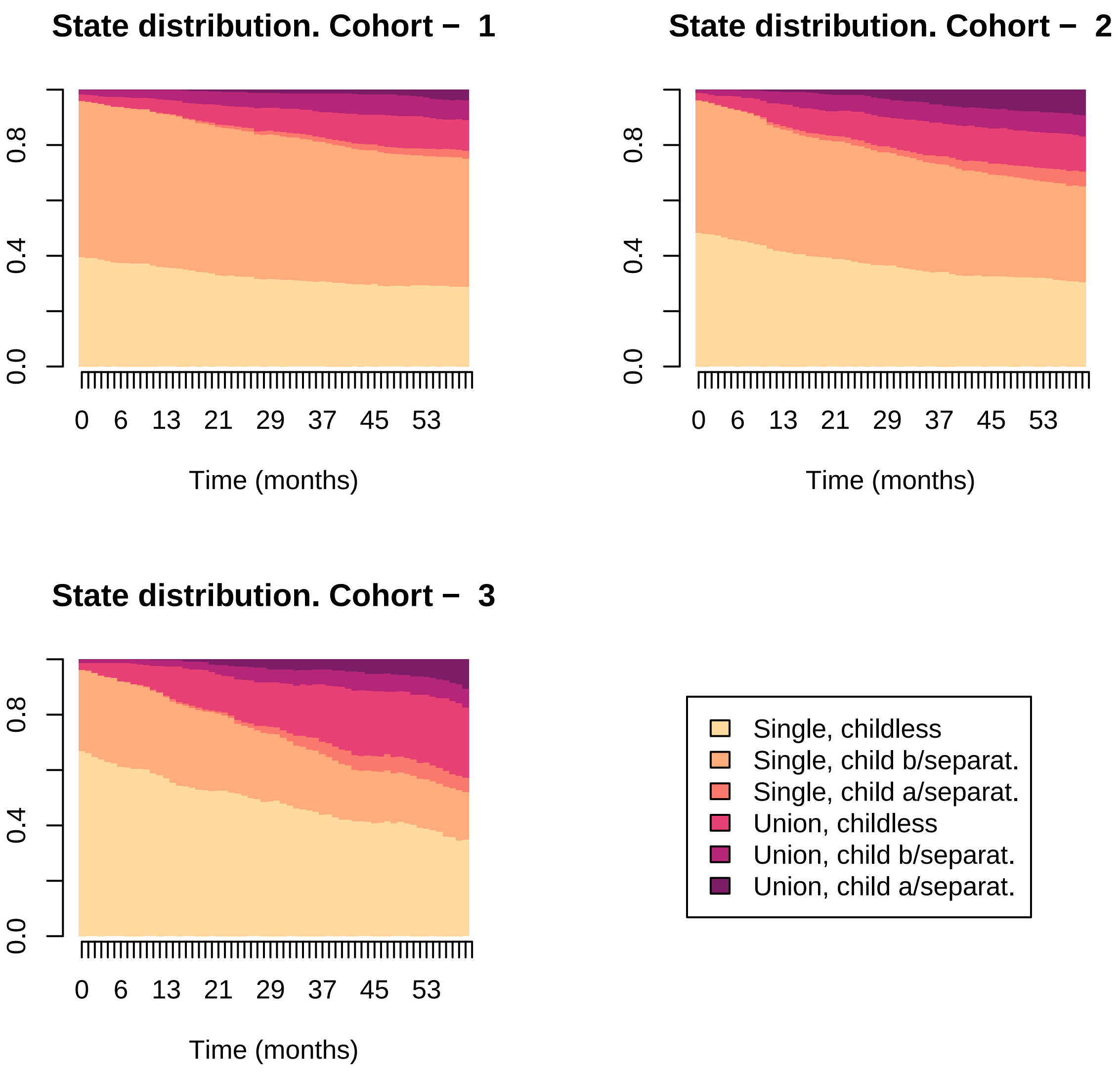
```
par(mfrow=c(2,1))
seqfplot(seqObj2, idxs=1:5, main="5 most frequent sequences",
         with.legend=TRUE, border=NA,
         ylab=NA, xlab="Time (months)", xtlab=xtlab)
```



6) Create a state distribution plot for each birthcohort (BIRTHCOH). What are the cross-cohort differences in the distribution of states overtime?

[Sol.]

```
seqdplot(seqObj2, group=data2$BIRTHCOH, with.legend=TRUE,
         main= "State distribution. Cohort", use.layout=FALSE,
         border=NA, xtlab=xtlab, ylab=NA, xlab="Time (months)")
```



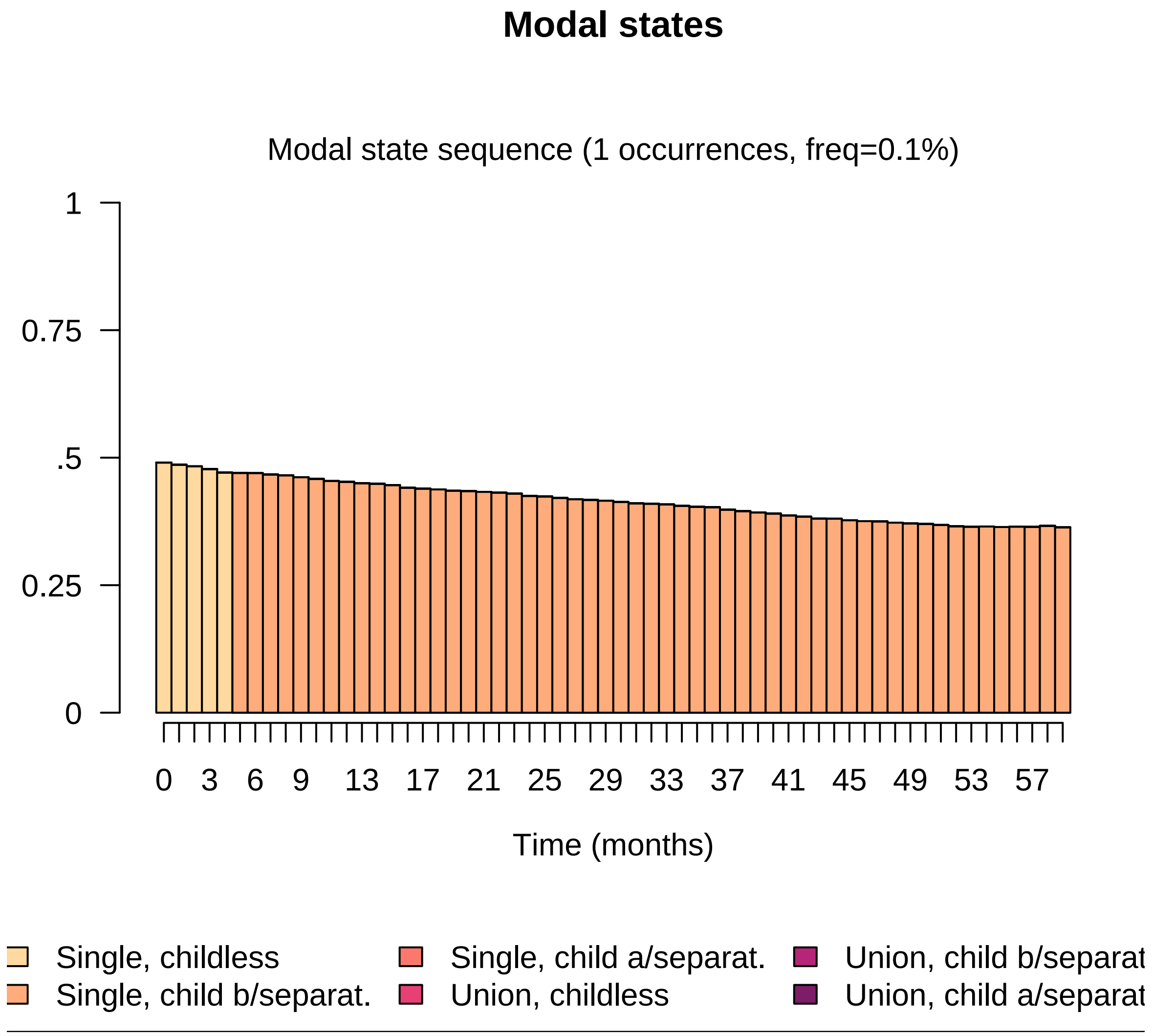
From the plot we can observe that cohort 3 has the biggest proportion of childless status, for both Single and Union types.

If cohorts 1 to 3 follow a time disposition, we can observe that more recent cohorts have higher proportions of children after separations. We can infer that more recent cohorts not only are more prone to not having children, but also are more prone of having children even after the couple dissolutes.

7) What are the most frequent states one and five years after break-up? Use a modal state plot for illustration.

[Sol.]

```
par(mfrow=c(1,1))
seqmsplot(seqObj2, with.legend=TRUE, main="Modal states",
          xtlab=xtlab, ylab=NA, xlab="Time (months)")
```



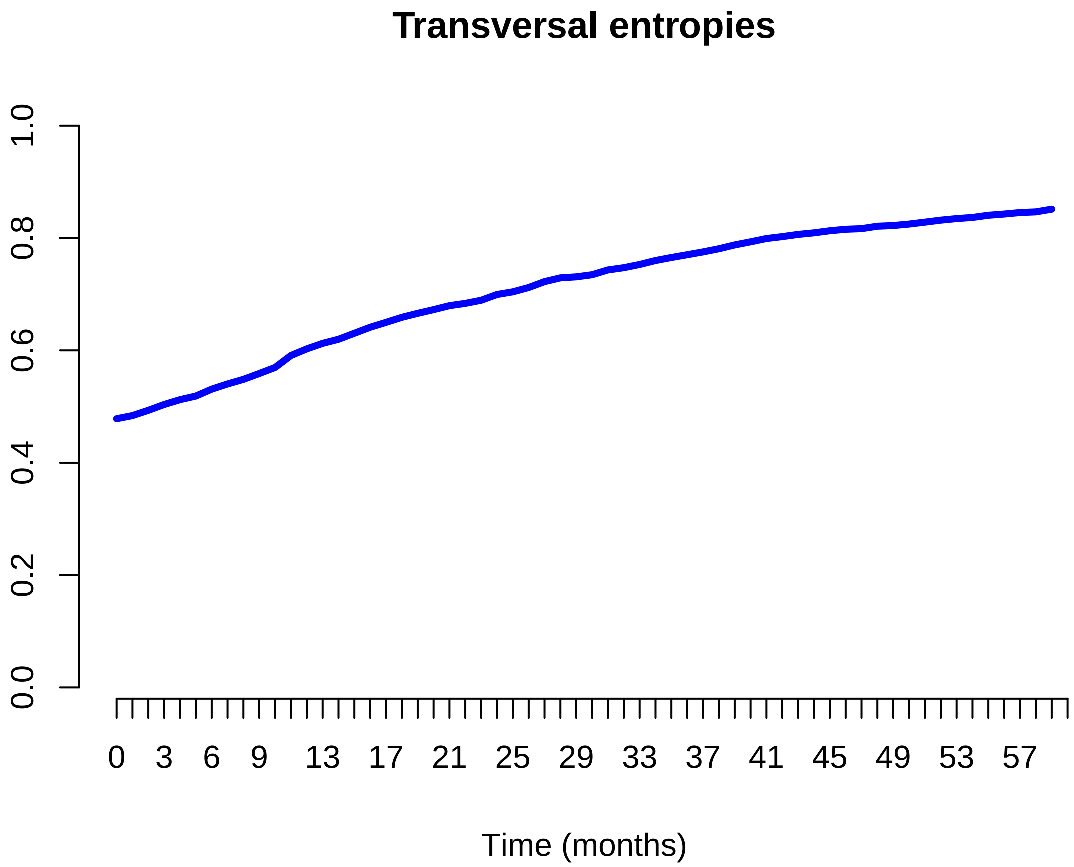
As can be seen, "Single, child b/separat." is the more frequent state after month 4. Before that, "Single, childless" is the frequent one.

This behavior was observable in the plot from the previuos question, but we have to remark that these frequent states are going to be different for each cohort.

8) Assess the cross-sectional state diversity plotting a measure of entropy. At what time after separation is the cross-sectional diversity of the states at its highest?

[Sol.]

```r
# Plot the transversal entropies in each position of the sequence
seqHtplot(seqObj2, with.legend=FALSE, main= "Transversal entropies",
          use.layout=FALSE, border=NA,xtlab=xtlab, ylab=NA, xlab="Time (months)")
```

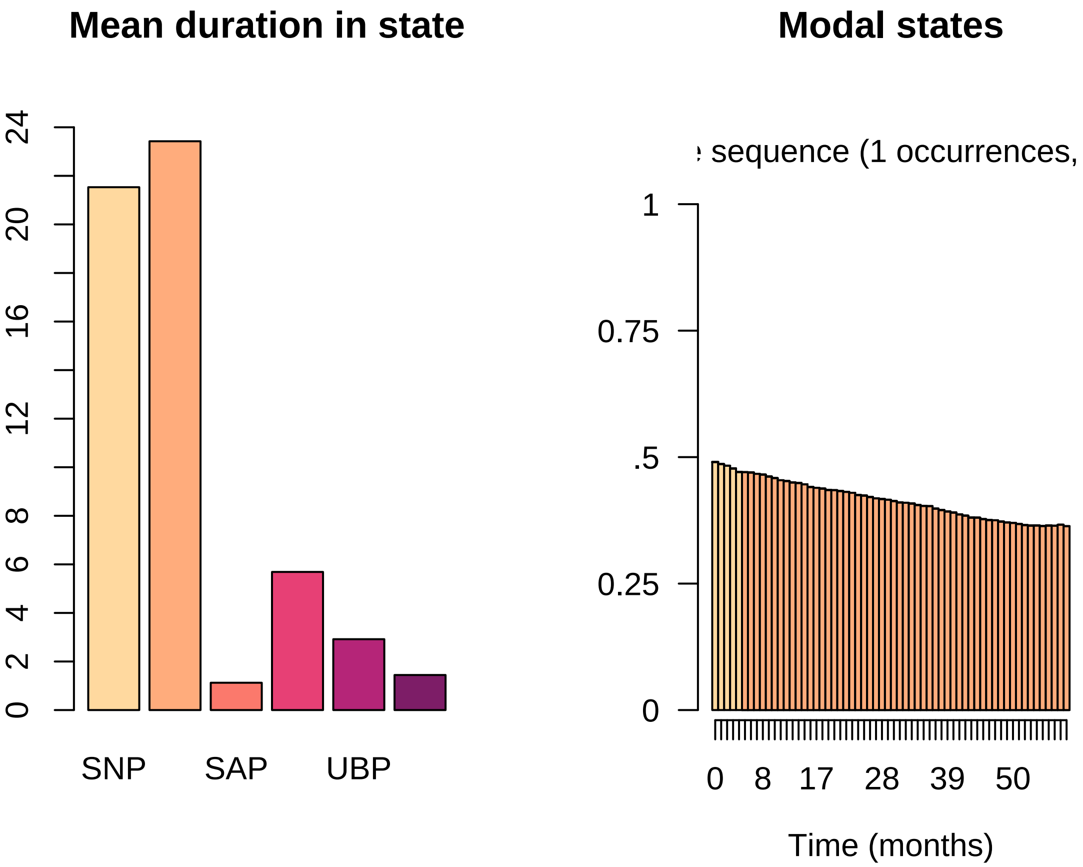


The plot shows that the entropy measure keeps increasing with time, having the peak at month 60 (the end of the observation time). We think that this is a result of having different ages in the individual sequence: even though we are analyzing the data from a life-course perspective, that 60-months could represent many different ages (and stages) in a woman's life, then creating a bias in the plot.

9) Display side by side in a same plot area the mean times spent in each of the states and the sequence of modal states.

[Sol.]

```
par(mfrow = c(1, 2))
# Plot the mean time spent in eache state
seqmtplot(seqObj2, with.legend=FALSE, main= "Mean duration in state",
          ylab=NA, ylim=c(0,25), yaxis=F)
axis(2, at=seq(from=0, to=25, by=2))
# Plot modal states in each position of the sequence
seqmsplot(seqObj2, with.legend=FALSE, main="Modal states", xtlab=xtlab,
          ylab=NA, xlab="Time (months)")
```



10) Compute the (overall) transition rate matrix. What is the largest transition rate between two different states?

[Sol.]

```
seqtrate(seqObj2)
```

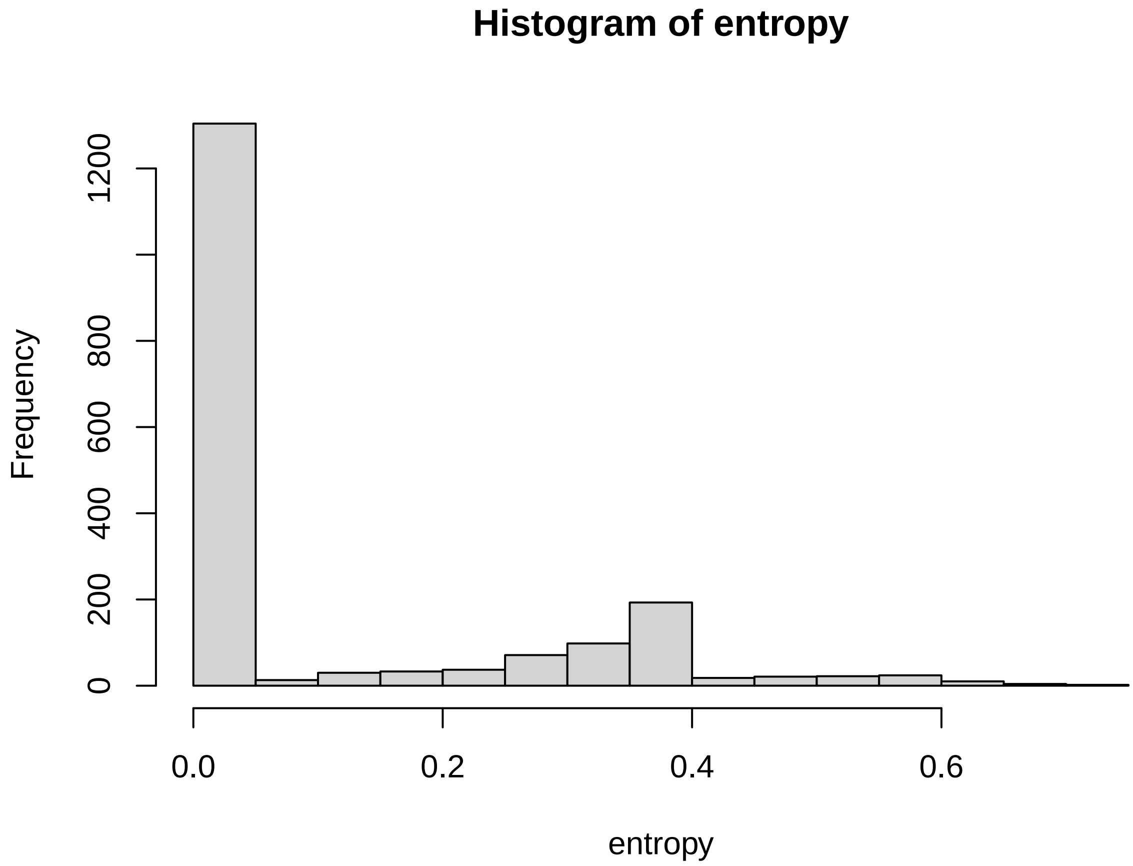
The largest transition rate between two different states is the one that goes from SNP to UNP: from Single with no children to Union with no children. The second largest is the one that reverses that states from UNP to SNP. This means that going in and out from Unions is more probable than moving to other states, which relates to the fact that states where children are present are less frequent through cohorts.

11) Compute the sequence length, the number of transitions, the number of subsequences and the longitudinal entropy

[Sol.]

```
# Sequence lenght - number of elements with valid cases (print results for first five sequences)
length <-seqlength(seqObj2)
length[1:5]
# Number of transitions between state episodes in each sequence (print results for first five sequences
transn <-seqtransn(seqObj2)
transn[1:5]
# Number of subsequences contained in a sequence
subseq <- seqsubsn(seqObj2)
table(subseq)
```

```
# Longitudinal or within-sequence entropy
entropy <- seqient(seqObj2)
par(mfrow=c(1,1))
hist(entropy)
```



12) Using summary(), look at the min, max, mean, median and quartiles of the distribution of each of the computed longitudinal characteristics.

[Sol.]

Summary of Sequence lenght:

```
print(summary(length))
```

Summary of Number of transitions between state episodes:

```
print(summary(transn))
```

Summary of Number of subsequences contained in a sequence:

```
print(summary(subseq))
```

Summary of Entropy:

```
print(summary(entropy))
```

Exercise 2

1) Input the Dataset 2

[Sol.]

```
data2 <- read.csv("SFS2018_Data2.csv", na.strings=c(".",".a",".b"))
```

2) Define a sequence object with elements in data columns 2:61 and alphabet 1:6, using the following state names and labels

   1 SNP "Single, childless",
   2 SBP "Single, child b/separat.",
   3 SAP "Single, child a/separat.",
   4 UNP "Union, childless",
   5 UBP "Union, child b/separat.",
   6 UAP "Union, child a/separat."

[Sol.]

```
#vector for the state labels
seqlab <-c("Single, childless",
           "Single, child b/separat.",
           "Single, child a/separat.",
           "Union, childless",
           "Union, child b/separat.",
           "Union, child a/separat.")

#vector of short state names (default would be alphabet labels)
sllist <- c("SNP","SBP","SAP","UNP", "UBP", "UAP")

###  Generate sequence object
seqObj2 <- seqdef(data2,
                  var=2:61,
                  alphabet=c(1:6),
                  cpal=color1,
                  states=sllist,
                  labels=seqlab)
```

3) Compute the matrix of pairwise distances - OM with constant costs - between all sequences and display the results for the first 5 sequences.

[Sol.]

```
#OM with CONSTANT subcosts (OM with indel=1, subs=2)
Matrix.OM.Const <- seqdist(seqObj2, method="OM", indel=1, sm="CONSTANT")
#display matrix
print(Matrix.OM.Const[1:5,1:5])
```

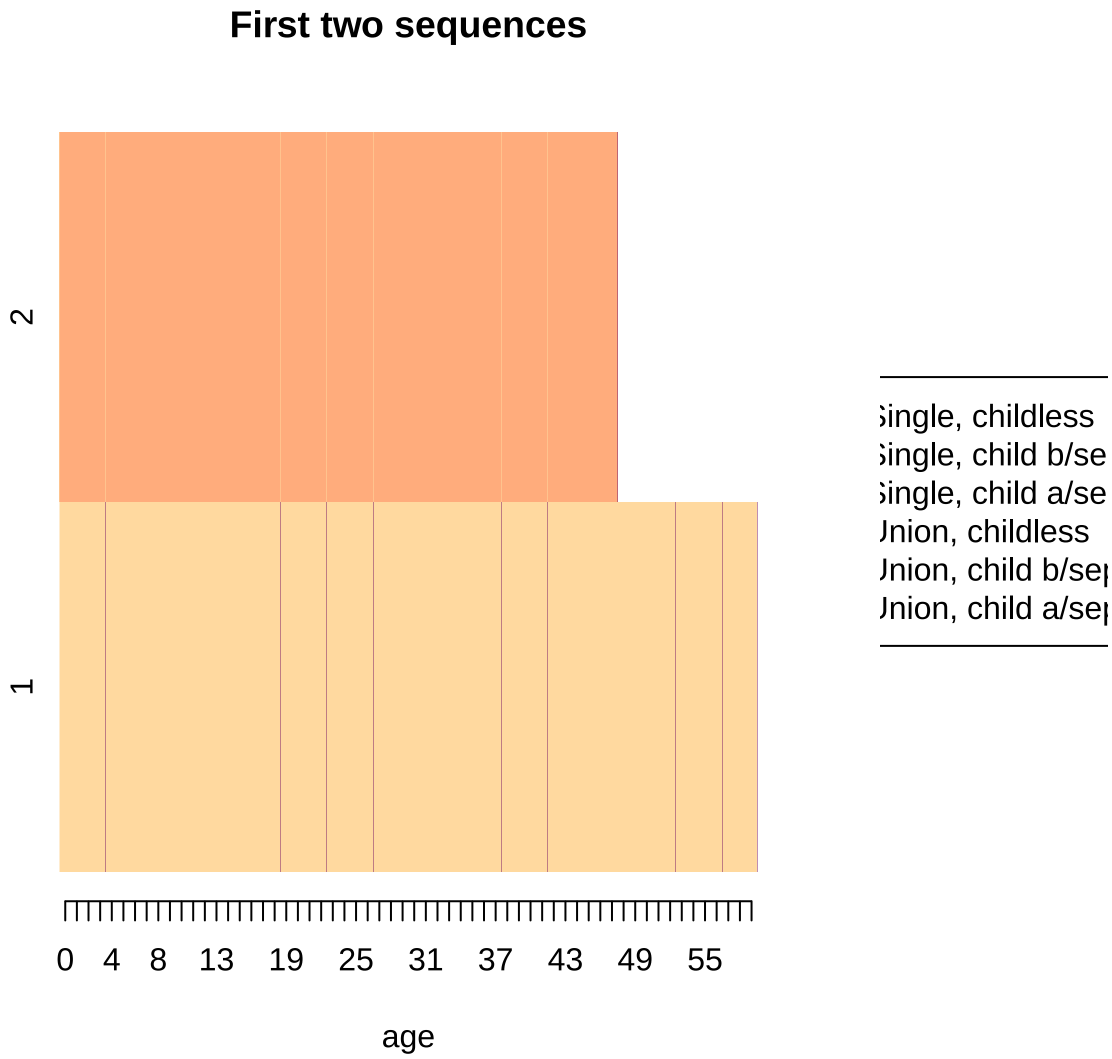
4) Plot the first 2 sequences and check that the OM distance is the number of non matching positions between them.

[Sol.]

```
#display the first 5 sequences, and sequence elements 1-20 (SPS format)
print(seqObj2[1:2, ], format ="SPS")

# Sequence
# 1 (SBP,48)
# 2 (SNP,60)
```

```
#All sequences -sequence index plot (sorted - first state)
xtlab=seq(0,60, by=1)
seqIplot(seqObj2[1:2, ], with.legend="right", main= "First two sequences",
         xtlab=xtlab, xlab="age", ylab=NA, yaxis=TRUE, sortv="from.start")
```



```
# 48*2 + 12 = 108
```

5) Check data that the LCS distance provides the same (non-normalized) distances as OM with indel=1 and a constant substitution cost of 2

[Sol.]

```
#Longest common subsequence
Matrix.LCS <- seqdist(seqObj2,method="LCS")
#display matrix
print(Matrix.LCS[1:5,1:5])

#Compare
```

```
print(Matrix.OM.Const[1:5,1:5])
```

6) Define a substitution cost matrix reflecting what (according to your prior knowledge) are the distances between two states (i.e. customize state-dependent substitution costs)

[Sol.]

We start by calculating the transition matrix, rounded to the 3rd decimal:

```
round(seqtrate(seqObj2),3)
```

We will try to reflect the ordinal character of this matrix by:

- setting the cost of remaining in the same state as 0
- the higher expected cost will be 10 (since dividing 1 by the transition probability would give us an `Inf`)
- the higher the probability, the lower the cost.
- for all other costs, we subtract the 3rd decimal of the transition probability to the maximum cost

We get the following state-dependent subcost matrix:

```
#OM with customized state-dependent subcosts
submatrix <- matrix(c( 0,10,9,2,10,10,
                       10,0,10,10,6,10,
                       10,10,0,10,10,6,
                       4,10,10,0,10,1,
                       10,6,10,10,0,5,
                       10,10,6,10,10,0), nrow = 6, ncol = 6, byrow = TRUE)

print(submatrix)
```

7) Compute the OM dissimilarity matrix using the previously derived substitution. Set the indel cost as half the maximum substitution cost.

[Sol.]

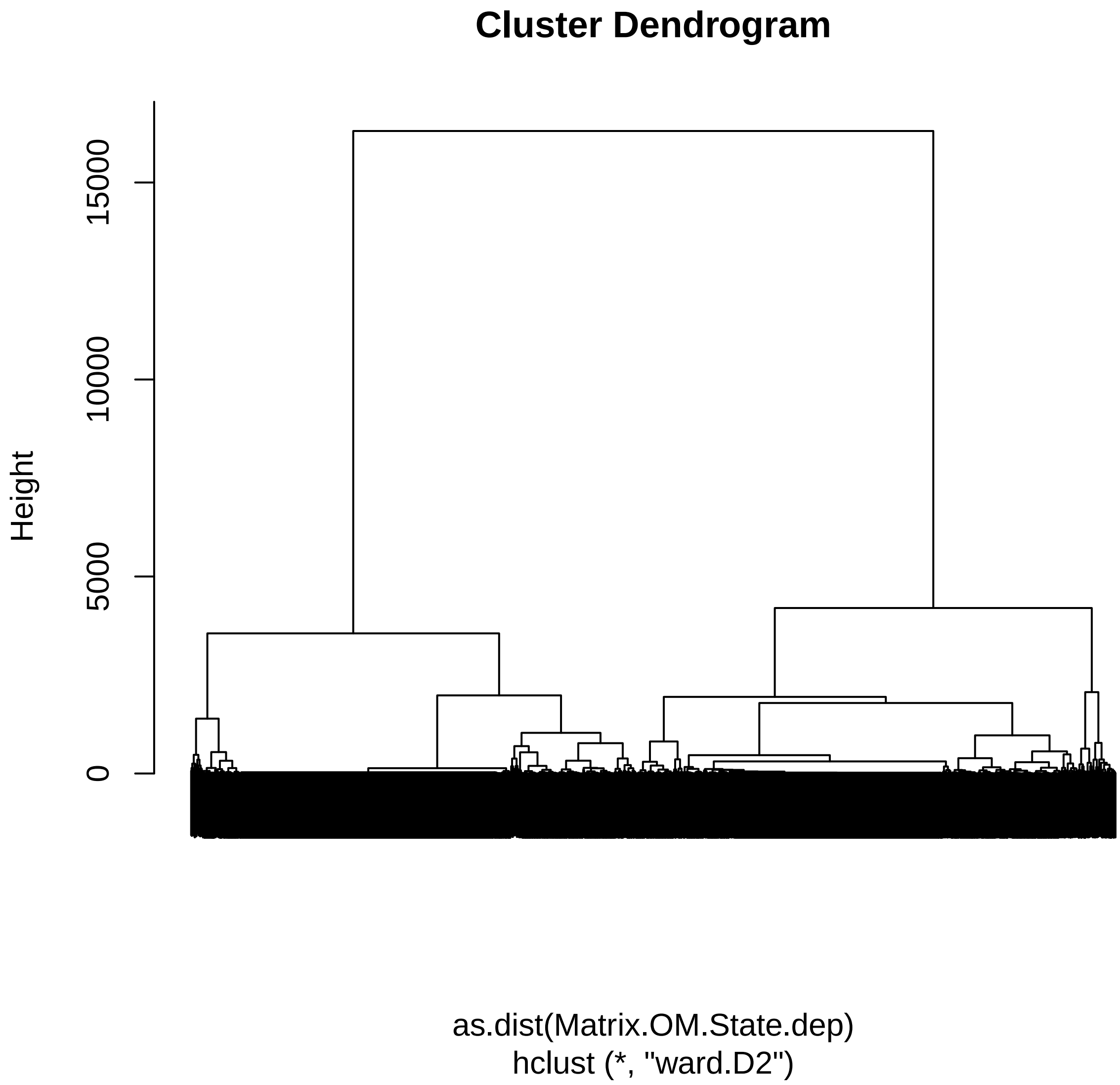
```
Matrix.OM.State.dep <- seqdist(seqObj2, method="OM", indel=5, sm=submatrix)
#display matrix
print(Matrix.OM.State.dep[1:5,1:5])
```

8) From the previously computed OM dissimilarity matrix, create a hierarchical cluster tree object with Ward method. Display the hierarchical tree

[Sol.]

```
# cluster sequences using the OM distances with state-dependent costs and Ward method
ward.OM <- hclust(as.dist(Matrix.OM.State.dep), method = "ward.D2")

###dendogram
# plot basic dendograms
plot(ward.OM, labels=FALSE)
```



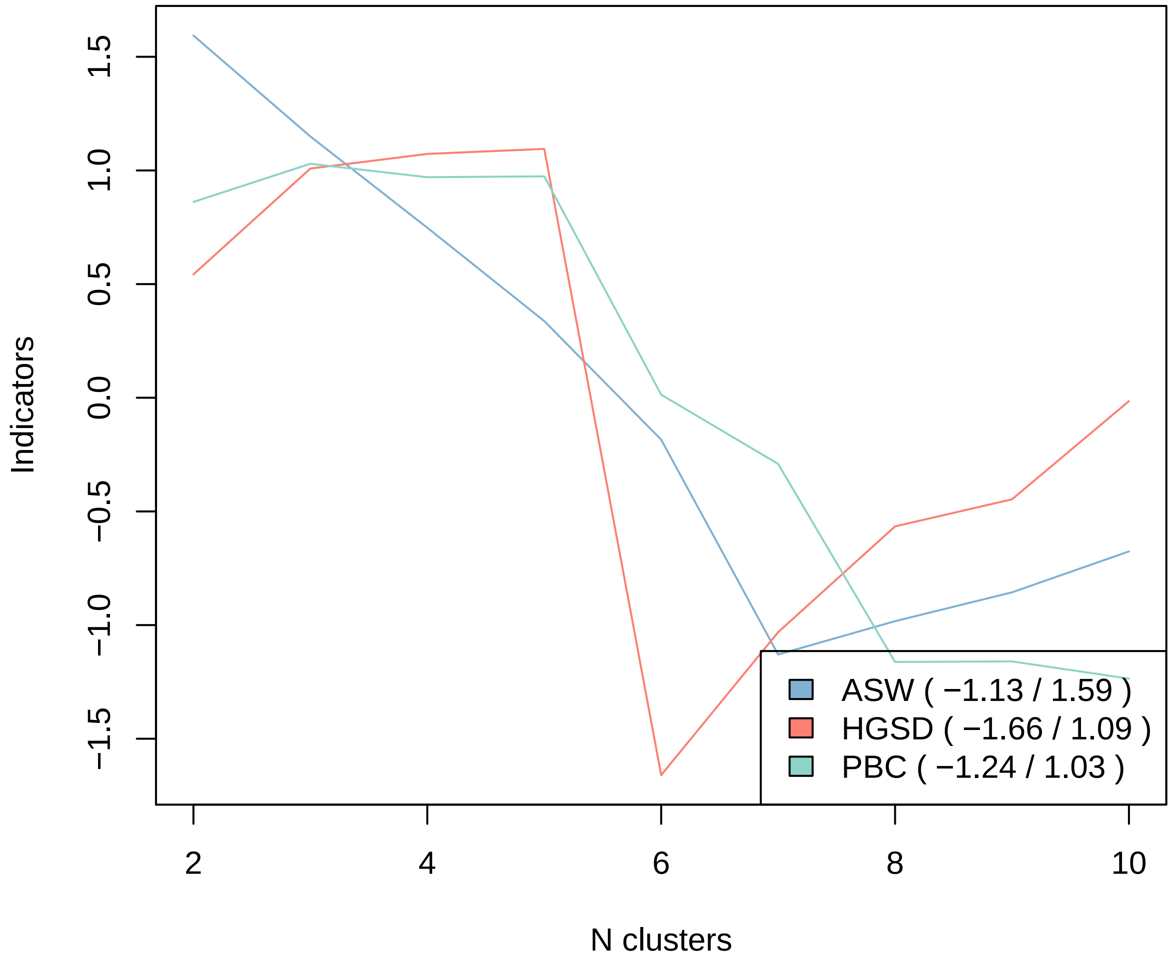
9) Calculate appropriate cluster cut-off criteria. Assess what is an empirically optimal cluster solution.

[Sol.]

```
### Generate an object with 1-10 cluster solutions for each prior anal
wardrange.OM <- as.clustrange(ward.OM, diss=Matrix.OM.State.dep, ncluster=10)

### show cluster cut-off measure values - indicate three optimal cluster solutions
summary(wardrange.OM, max.rank=3)

### plot ASW, HGSD and PBC
plot(wardrange.OM, stat=c("ASW", "HGSD", "PBC"), norm="zscore")
```

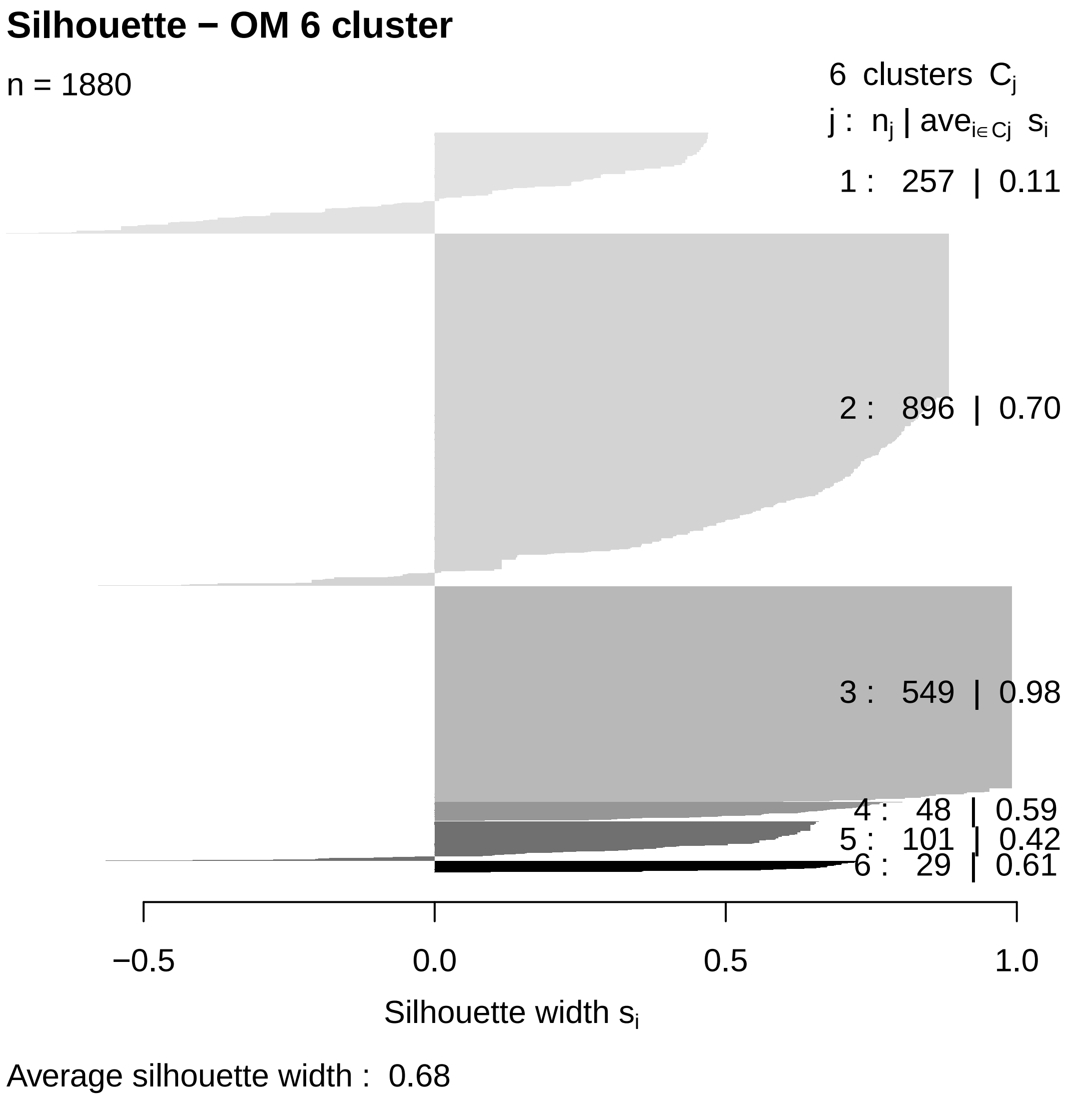


10) Select the six-cluster solution from the Ward analysis, check cluster consistency, and label the clusters by looking at the full sequence index plots (or the relative frequency version) by cluster.

[Sol.]

```r
### store cluster solutions with best empirical fits
#OM
wardrange.OM.6 <- cutree(ward.OM , k=6)

### cluster consistency (plot silhouette widths)
#OM 5-cluster solution
silh.OM.6 <- silhouette(wardrange.OM.6, dmatrix = Matrix.OM.State.dep)
summary(silh.OM.6)
plot(silh.OM.6, main= "Silhouette - OM 6 cluster", border=NA,
     col=c("#E2E2E2", "#D3D3D3", "#B8B8B8", "#969696", "#707070", "black"))
```



11) Repeat steps 8-10 using an OM with transition rates as substitution costs, and 1 as indel costs.

[Sol.]

Since we are already using the transition rates matrix as a proxy of substitution costs, now we are lowering the indel cost from 5 to 1 and comparing results.

```
Matrix.OM.State.dep.indel1 <- seqdist(seqObj2, method="OM", indel=1, sm=submatrix)
#display matrix

# cluster sequences using the OM distances with state-dependent costs and Ward method
ward.OM.indel1 <- hclust(as.dist(Matrix.OM.State.dep.indel1), method = "ward.D2")


### Generate an object with 1-10 cluster solutions for each prior anal
wardrange.OM.indel1 <- as.clustrange(ward.OM.indel1, diss=Matrix.OM.State.dep.indel1, ncluster=10)
```

```
### store cluster solutions with best empirical fits
# OM
wardrange.OM.6.indel1 <- cutree(ward.OM.indel1 , k=6)

### cluster consistency (plot silhouette widths)
# OM 6-cluster solution
silh.OM.6.indel1 <- silhouette(wardrange.OM.6.indel1, dmatrix = Matrix.OM.State.dep.indel1)
```

12) Compare the results between the OM and the DHD approaches

[Sol.]

By lowering the indel costs, we get the following best fit for number of clusters:

```
### plot ASW, HGSD and PBC: indel cost = 5
plot(wardrange.OM, stat=c("ASW", "HGSD", "PBC"), norm="zscore")
```

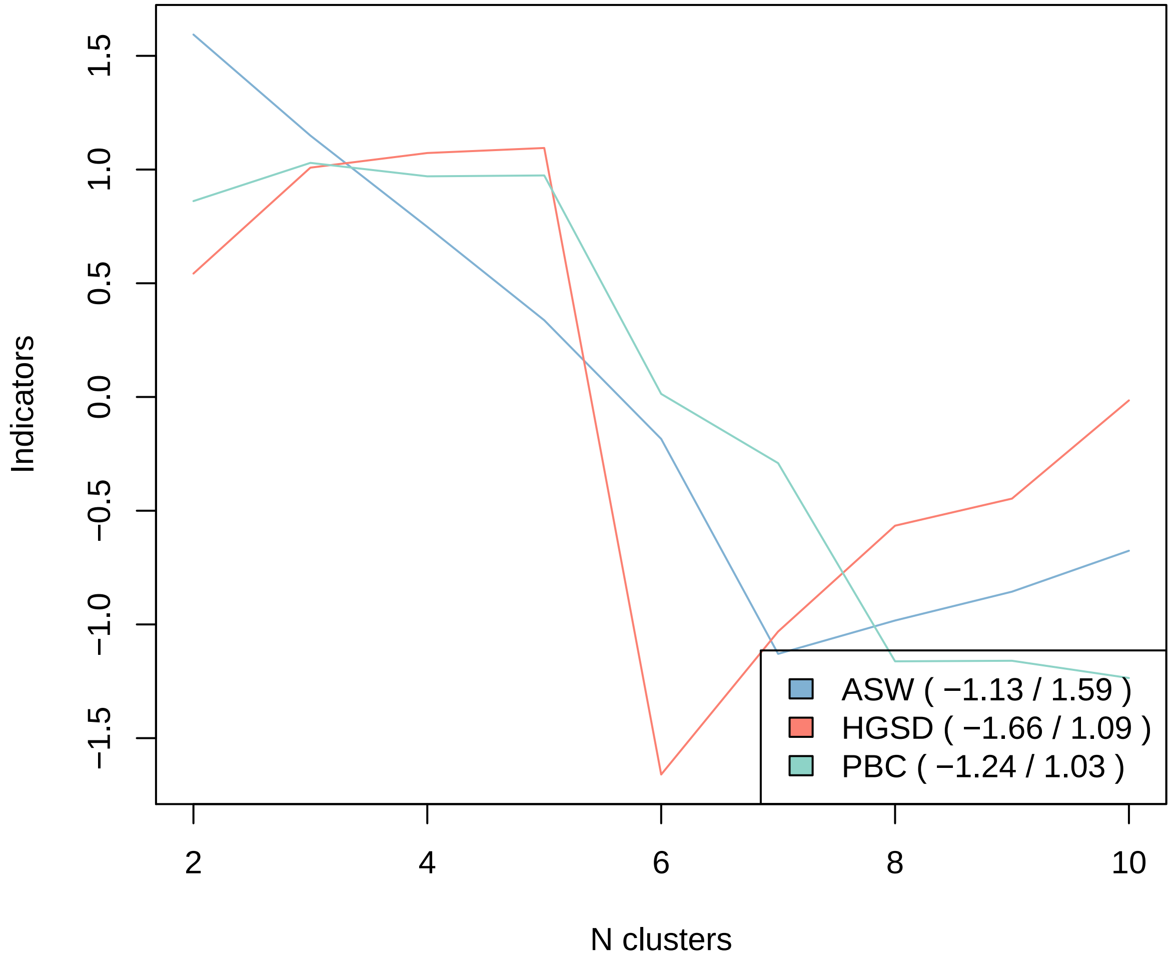


Figure 1: Indel cost = 5

```r
### plot ASW, HGSD and PBC: indel cost = 1
plot(wardrange.OM.indel1, stat=c("ASW", "HGSD", "PBC"), norm="zscore")
```

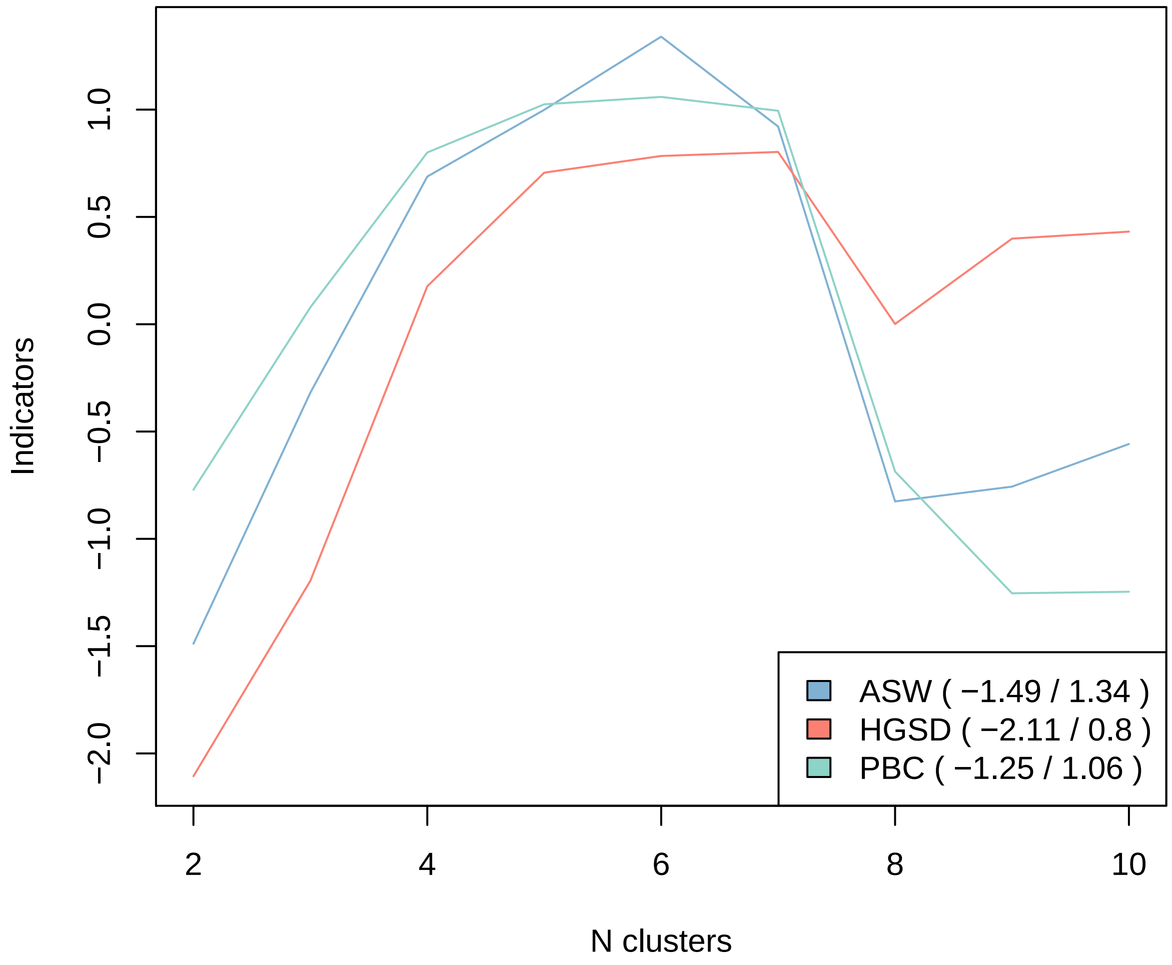
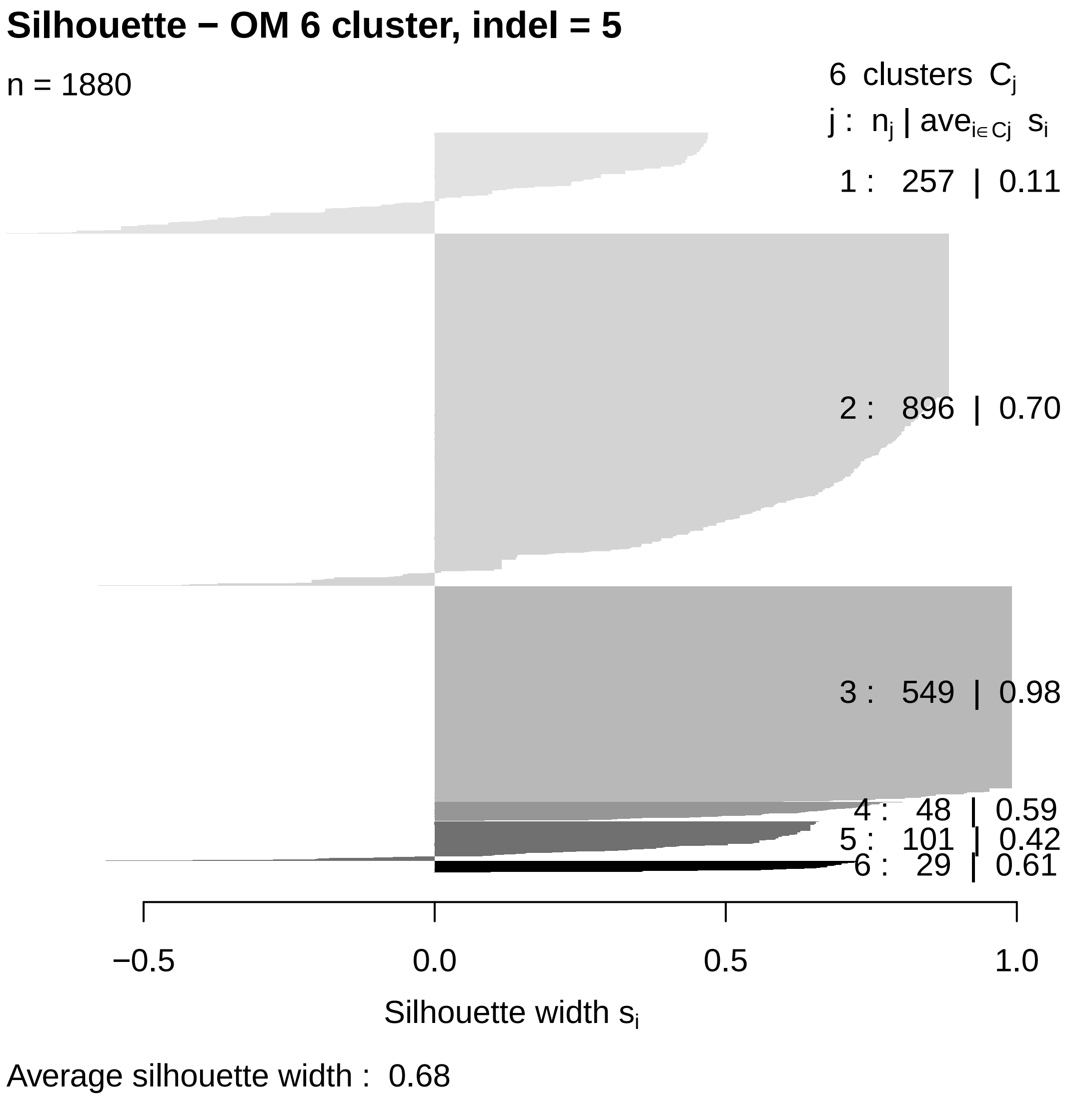


Figure 2: Indel cost = 1

When indel costs = 5 the best fit seems to be around 5 clusters, and when we lower the indel costs to 1, the best fit seems to be around 6 clusters.

If we force the solution to be a 6-cluster one, we get the following silhouette fittings:

```r
### cluster consistency: indel cost = 5
plot(silh.OM.6, main= "Silhouette - OM 6 cluster, indel = 5", border=NA,
     col=c("#E2E2E2", "#D3D3D3", "#B8B8B8", "#969696", "#707070", "black"))
```



```
### cluster consistency: indel cost = 1
plot(silh.OM.6.indel1, main= "Silhouette - OM 6 cluster, indel = 1", border=NA,
     col=c("#E2E2E2", "#D3D3D3", "#B8B8B8", "#969696", "#707070", "black"))
```

When we have indel costs = 5, the first cluster does not have the best fitting. But when we lower the costs down to 1, the 6-cluster solution is a much better fitting.

We understand that if we use the transition rates matrix as a substitution matrix cost, we need lower indel costs than usual to have better fitting for clustering.

References

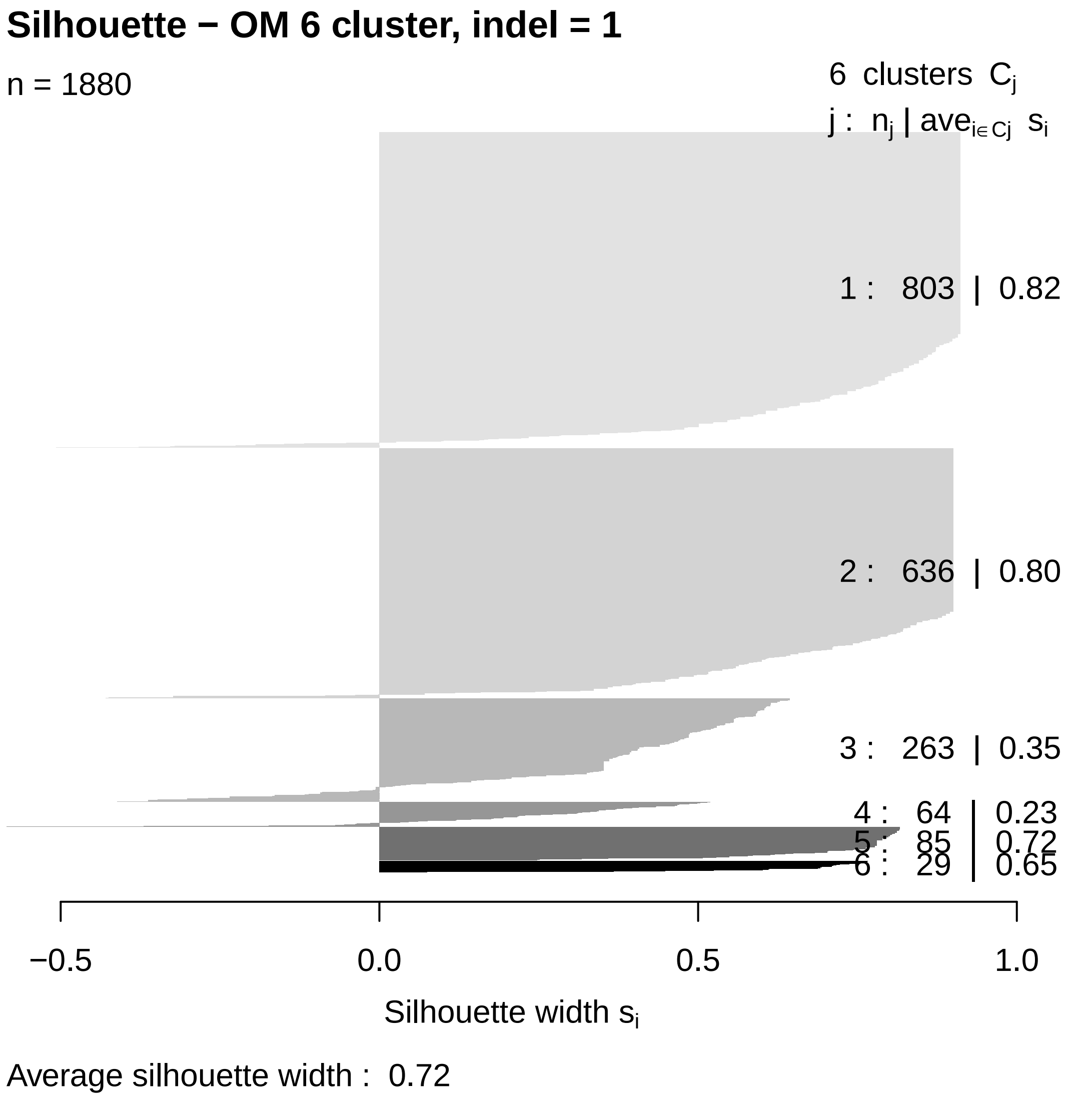


Figure 3: Silhoutte Indel cost = 1