

TraMineR

```
library(TraMineR)

##
## TraMineR stable version 2.2-6 (Built: 2023-02-03)
## Website: http://traminer.unige.ch
## Please type 'citation("TraMineR")' for citation information.
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr  0.3.5
## v tibble  3.1.8      v dplyr  1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
library(cluster)

tinderMessagesDf <- read.csv("Datasets/Sequential Data - Tinder messages/messages.db.csv")

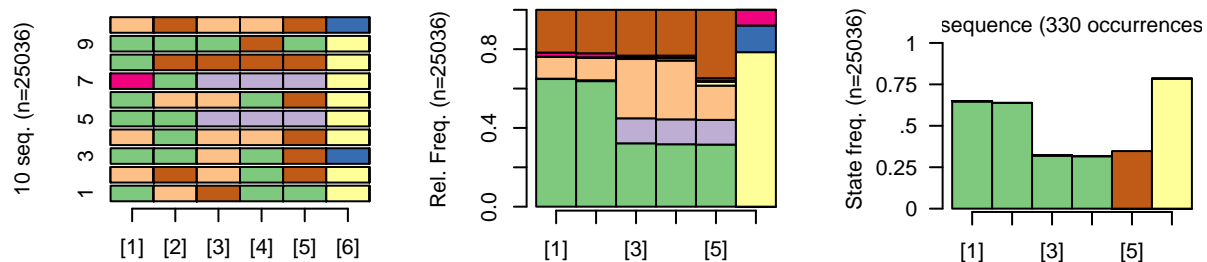
Make a sequence for each conversationid made of each message.type
tinderMessagesSeq <- tinderMessagesDf %>%
  group_by(conversationid) %>%
  summarise(seq = paste(tail(message.type, 6), collapse = ","))
traminerSeq <- seqdef(tinderMessagesSeq$seq, stsep = ",")

## [>] found missing values ('NA') in sequence data
## [>] preparing 25036 sequences
## [>] coding void elements with '%' and missing values with '*'
## [>] 7 distinct states appear in the data:
##      1 = aff.nor
##      2 = aff.off
##      3 = chunk
##      4 = end
##      5 = phone
##      6 = piss.off
##      7 = questn
## [>] state coding:
```

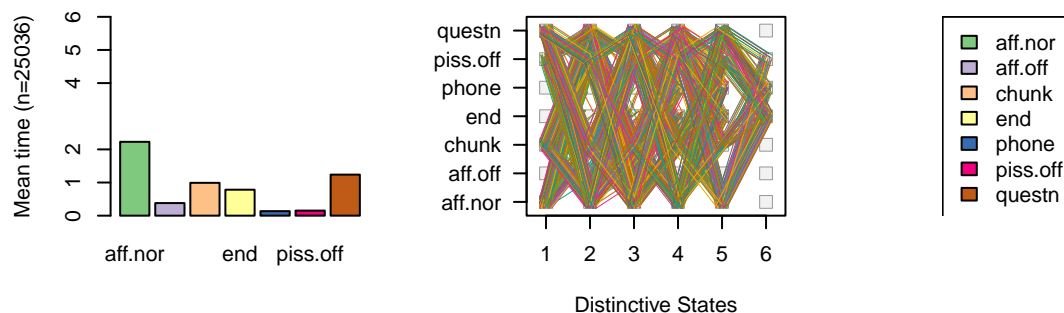
```
##      [alphabet] [label] [long label]
##      1 aff.nor    aff.nor  aff.nor
##      2 aff.off    aff.off  aff.off
##      3 chunk      chunk    chunk
##      4 end        end      end
##      5 phone      phone    phone
##      6 piss.off   piss.off  piss.off
##      7 questn     questn    questn
## [>] 25036 sequences in the data set
## [>] min/max sequence length: 2/6
```

Visualize the sequences

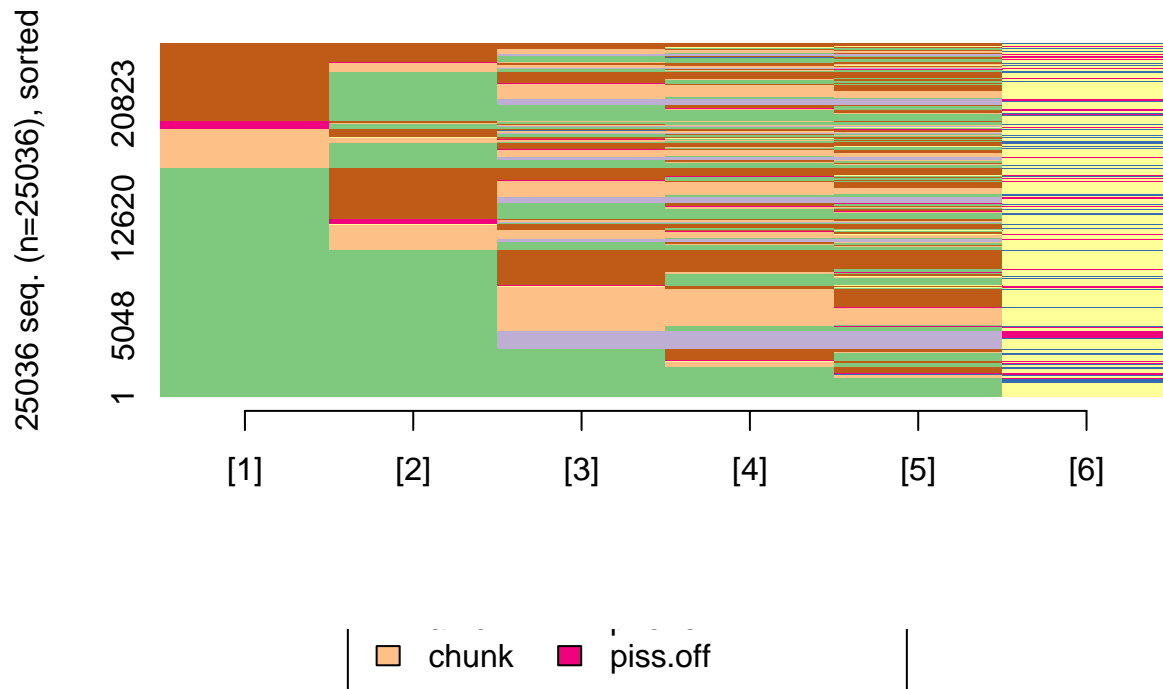
```
# Interesting to disable borders with border = NA too
par(mfrow = c(2, 3))
seqplot(traminerSeq, type = "i", with.legend = FALSE)
seqplot(traminerSeq, type = "d", with.legend = FALSE)
seqplot(traminerSeq, type = "ms", with.legend = FALSE)
seqplot(traminerSeq, type = "mt", with.legend = FALSE)
seqplot(traminerSeq, type = "pc", with.legend = FALSE)
seqlegend(traminerSeq)
```



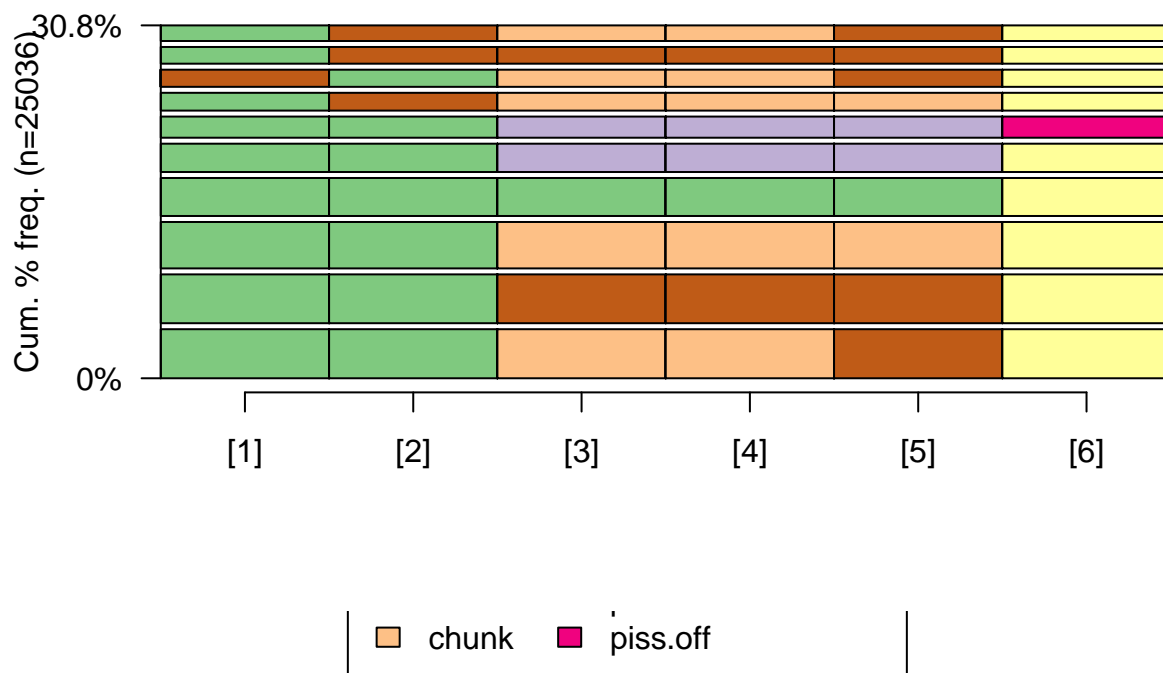
n = 25036



```
# Do not run this with httdpd
seqIplot(traminerSeq, sortv = "from.start")
```

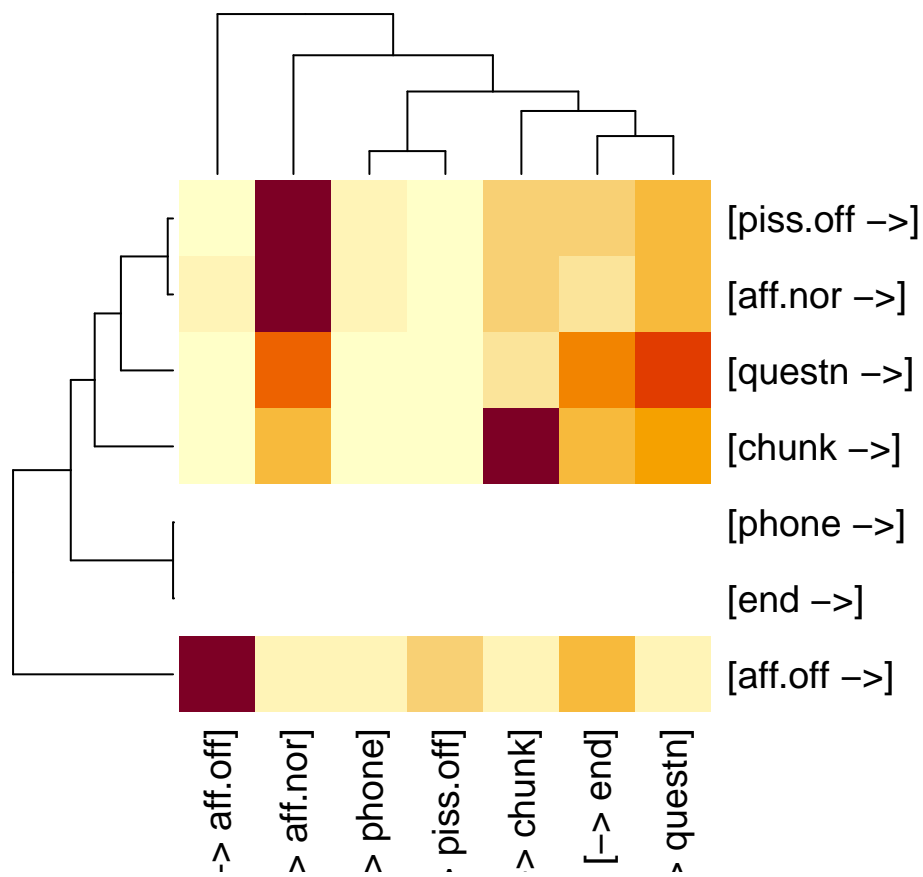


```
seqplot(traminerSeq, type = "f")
```



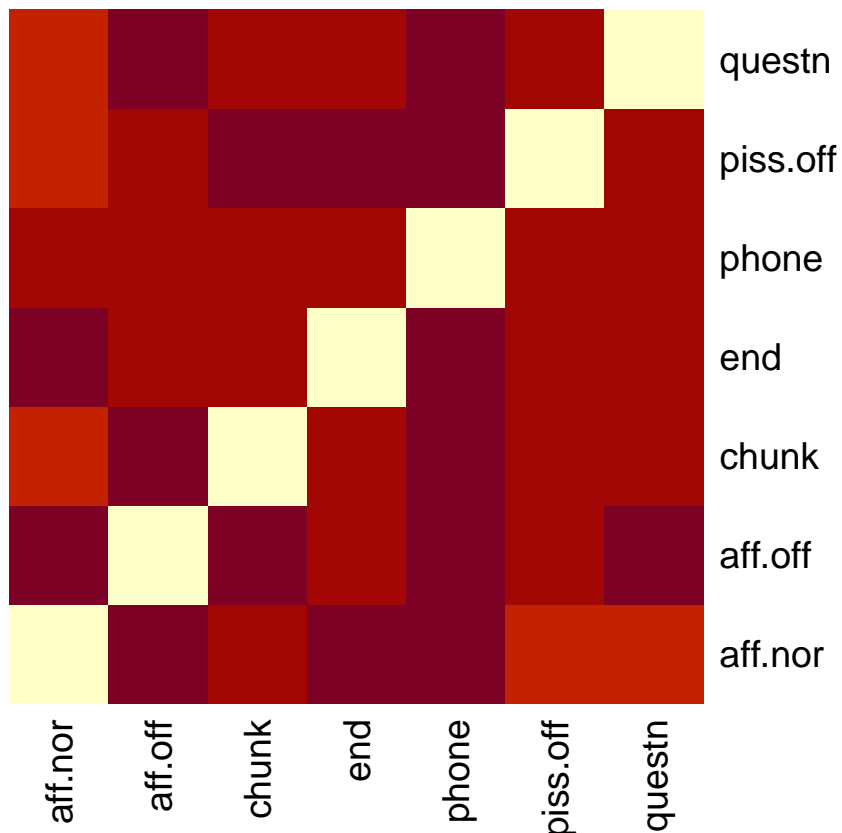
```
heatmap(seqtrate(traminerSeq), )
```

```
## [>] computing transition probabilities for states aff.nor/aff.off/chunk/end/phone/piss.off/questn .
```



```
submat <- seqsubm(traminerSeq, method = "TRATE")

## [>] creating substitution-cost matrix using transition rates ...
## [>] computing transition probabilities for states aff.nor/aff.off/chunk/end/phone/piss.off/questn .
# Make a heatmap
heatmap(submat, Rowv = NA, Colv = NA, )
```



```
submat
```

```
##      aff.nor  aff.off   chunk    end    phone piss.off  questn
## aff.nor  0.000000 1.963282 1.678562 1.895833 1.964680 1.532790 1.529460
## aff.off  1.963282 0.000000 1.986610 1.815936 1.991485 1.823746 1.977936
## chunk    1.678562 1.986610 0.000000 1.841016 1.982537 1.842879 1.713964
## end      1.895833 1.815936 1.841016 0.000000 2.000000 1.876570 1.745137
## phone    1.964680 1.991485 1.982537 2.000000 0.000000 1.959039 1.972711
## piss.off 1.532790 1.823746 1.842879 1.876570 1.959039 0.000000 1.802174
## questn   1.529460 1.977936 1.713964 1.745137 1.972711 1.802174 0.000000
```

```
sampleSeq <- seqdef(sample(tinderMessagesSeq$seq, size = 2500), stsep = ",")
```

```
## [>] found missing values ('NA') in sequence data
## [>] preparing 2500 sequences
## [>] coding void elements with '%' and missing values with '*'
## [>] 7 distinct states appear in the data:
##      1 = aff.nor
##      2 = aff.off
##      3 = chunk
##      4 = end
##      5 = phone
##      6 = piss.off
```

```

##      7 = questn
## [>] state coding:
##      [alphabet] [label] [long label]
##      1 aff.nor    aff.nor aff.nor
##      2 aff.off    aff.off aff.off
##      3 chunk      chunk   chunk
##      4 end        end     end
##      5 phone      phone   phone
##      6 piss.off   piss.off piss.off
##      7 questn     questn  questn
## [>] 2500 sequences in the data set
## [>] min/max sequence length: 2/6
# Use traminer to cluster the sequences
library(cluster)
dist.om1 <- seqdist(sampleSeq, method = "OM", indel = 1, sm = "TRATE")

## [>] 2500 sequences with 7 distinct states
## [>] Computing sm with seqcost using  TRATE
## [>] creating substitution-cost matrix using transition rates ...
## [>] computing transition probabilities for states aff.nor/aff.off/chunk/end/phone/piss.off/questn .
## [>] 490 distinct  sequences
## [>] min/max sequence lengths: 2/6
## [>] computing distances using the OM metric
## [>] elapsed time: 0.232 secs

clusterward1 <- agnes(dist.om1, diss = TRUE, method = "ward")
# plot(clusterward1, which.plot = 2)
cl1.4 <- cutree(clusterward1, k = 4)
cl1.4fac <- factor(cl1.4, labels = paste("Type", 1:4))

```

```
seqIplot(sampleSeq, group = cl1.4fac, sortv = "from.start")
```

