The Demonstration of myTraMineR functions

Xinming Mia Dai

2023-12-14

```
# install.packages("myTraMineR", repos = NULL, type="source")
library(myTraMineR)
library(TraMineR)
data(mvad)
mvad.alphab <- c("employment", "FE", "HE", "joblessness", "school", "training")
mvad.seq <- seqdef(mvad, 17:86, xtstep = 6, alphabet = mvad.alphab)
```

```
seqfplot(mvad.seq, idxs = 1:20)
```

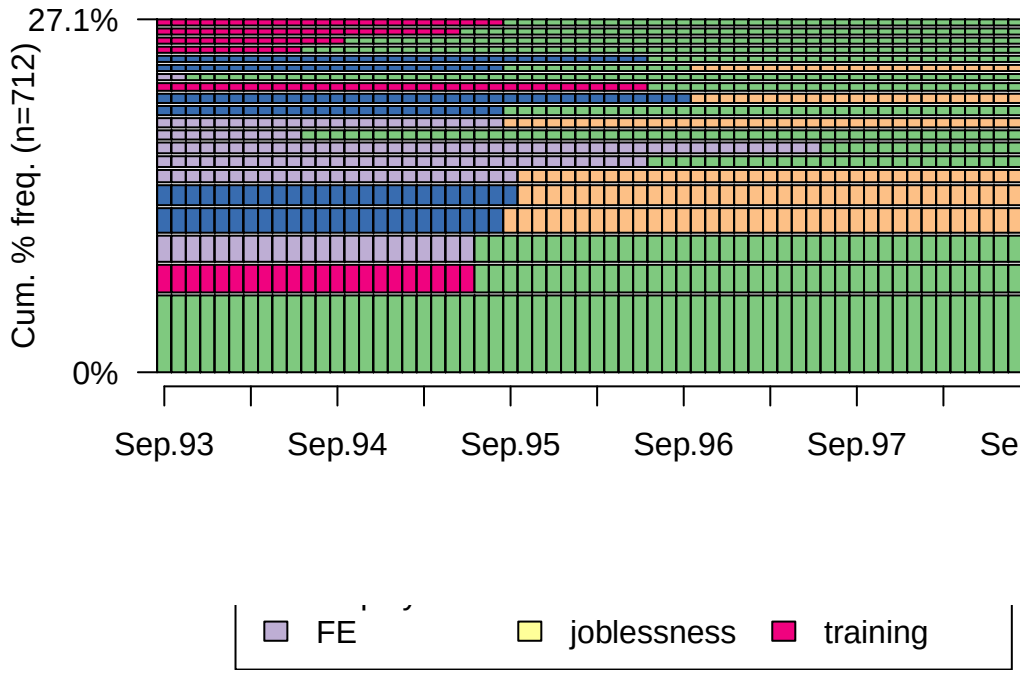

Figure 1: Frequence plot

Figure 1 shows that the first 20 most frequent sequences represent 27.1% of the total. Now, let's consider a situation: if a sequence differs by only one state from the most common sequence, which consists solely of the "employment" state, this sequence won't be shown in the plot. Several cases are shown in figure 2. The difference may be caused by repeating error—noise, so we are interested in plotting this sequence in the frequency plot as well. In other words, we would like to replace sequences with their representatives and then draw a frequency plot.

Note that it will be important to examine whether small differences among sequences are true or not in the exploratory data analysis process. If users observe a significant improvement in frequency when considering small differences as not true, then in the later analysis, users should consider using the new dataset obtained

by replacing these sequences with their representatives.

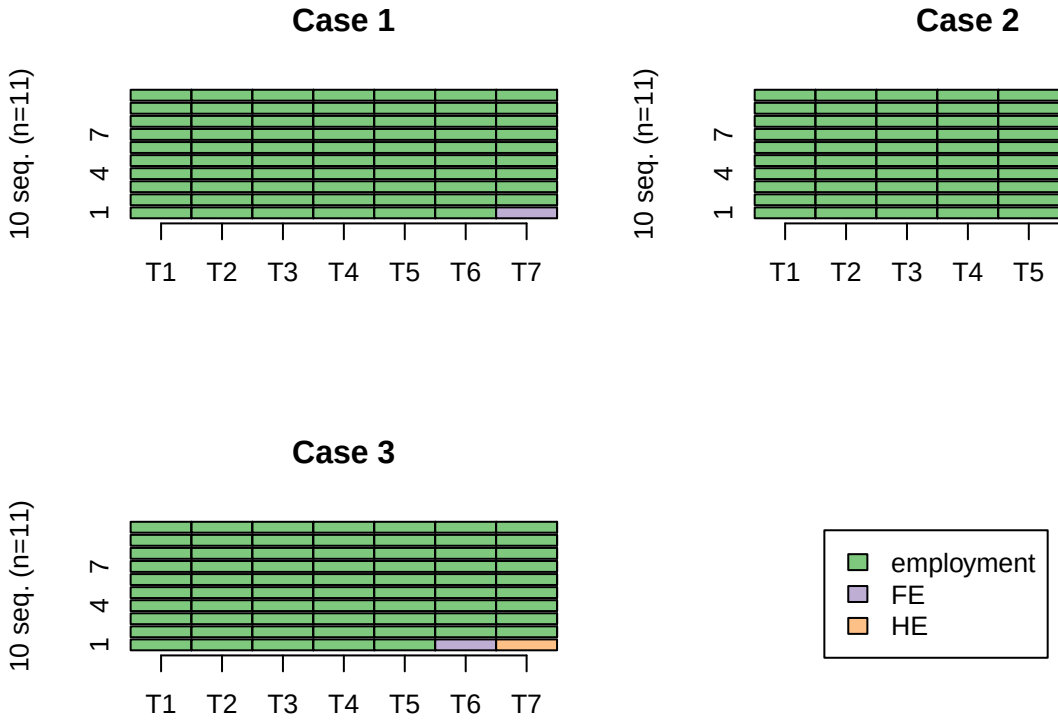

Figure 2: In case 1, the first sequence only differs from the most frequent sequence by one state. In case 2, the sequence, which consists solely of the "employment" state, is still most frequent, and the first two sequences differ from it by one state. In case 3, the first sequence only differs from the most frequent sequence by two states.

Function 1: The Extension of `seqfplot()`

Figure 3 shows that if we consider two sequences different by one Hamming distance as the same, the first 20 most frequent sequences represent 31.6% of the total. This number increased by 16.6% compared to Figure 1.

```
seqdistance <- seqdist(mvad.seq, method="HAM")
clusters <- seqcluster(seqdistance, h=1.5, cmethod='complete')
alphabet <- c("employment", "FE", "HE", "joblessness", "school", "training")
mvad.replaced <- seqrep_replace(mvad, clusters, var=17:86, alphabet, idxs=1:20, xtsetp=6)
seqfplot(mvad.replaced, idxs=1:20)
```

We cut the hierarchical tree to enforce the maximal distance within clusters to separate the 20 most frequent sequences. Next, we identified the representative—specifically, the mode—for each cluster and used these representatives to replace every element in the clusters. This new dataset, named `mvad.replaced`, is also a `stslist` object. Finally, we generated a frequency plot for `mvad.replaced`.

1. `seqdist()` is used to calculate distance matrix using Hamming distance. This is already in `TraMineR`. Users can switch to other distance measurements here.
2. `seqcluster()` uses complete linkage clustering and returns a data frame containing cluster information. There are three columns in the returned data frame—index, cluster, and Freq. It shows which cluster a data point belongs to and the cluster size. Users have the flexibility to switch to other agglomeration methods and specify heights where the dendrogram should be cut. For example, if a user chooses `h = 2.5`, then two sequences differing by two states will be considered the same, as illustrated in case 3 in Figure 3.

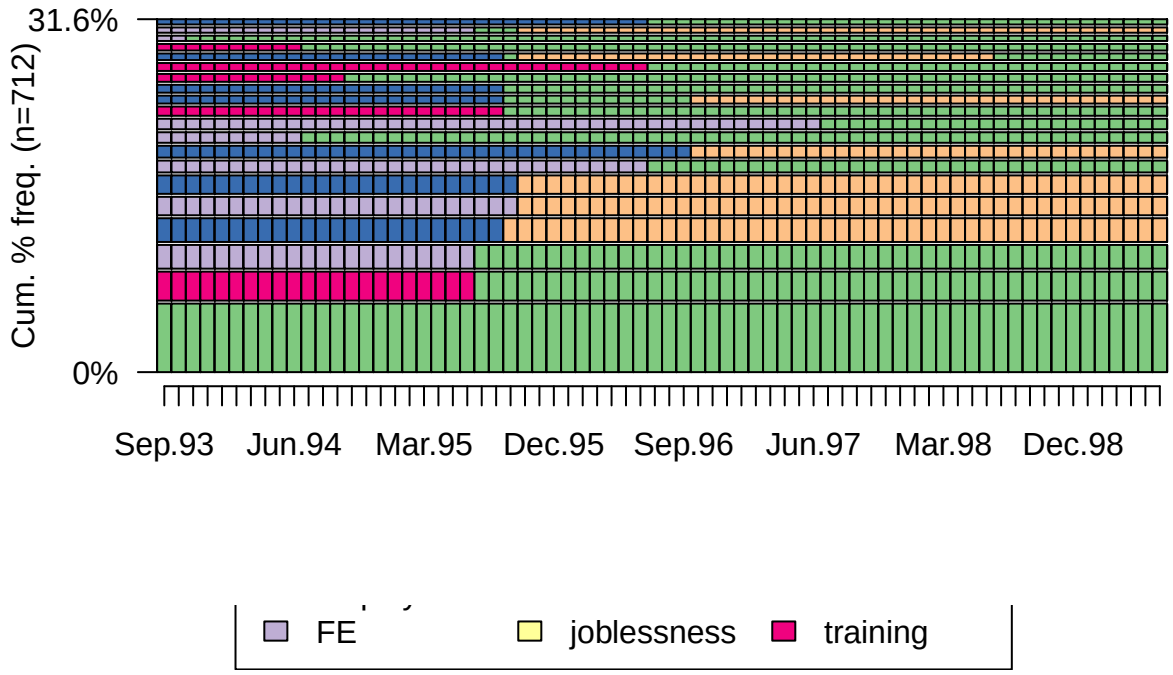

Figure 3: Frequency plot for mvad.replaced dataset.

3. `seqrep_replace()` is similar as `seqdef()`. It returns a `stslist` object but with elements replaced by their cluster representatives.

If significant improvement on the frequency is observed, then the new `stslist` object, `mvad.replaced`, should be considered in the later analysis.

Use `help()` to check the usage of two functions—`seqcluster()` and `seqrep_replace()` that we created in `myTraMineR` package.

Cluster sequences

Description

This function clusters sequences using hierarchical cluster and returns group memberships of sequences.

Usage

```
seqcluster(seqdistance, h = 1.5, cmethod = "complete")
```

Arguments

`seqdistance`	A distance matrix or a distance array returned by TraMineR::seqdist() function.
`h`	Numeric scalar or vector with heights where the tree should be cut. The default is 1.5.
`cmethod`	The agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC). The default clustering method is "complete", complete-linkage clustering.

Value

`seqcluster` returns a data frame with three columns—the index of original data, group memberships, and group size.

Examples

```
data(mvad)
mvad.alphab <- c("employment", "FE", "HE", "joblessness", "school", "tra
mvad.seq <- seqdef(mvad, 17:86, xtstep = 6, alphabet = mvad.alphab)
seqdistance <- seqdist(mvad.seq, method="HAM")
clusters <- seqcluster(seqdistance, h=1.5, cmethod='complete')
```

R: Replace sequences with their representatives ▾ seqdist < >

seqrep_replace {myTraMineR} R Documentation

Replace sequences with their representatives

Description

Similar as `seqdef()`. It returns a `stslist` object but with elements replaced by their cluster representatives.

Usage

```
seqrep_replace(data, clusters, var = NULL, alphabet, idxs = 1:10, ...)
```

Arguments

`data`	A data frame, matrix, or character string vector containing sequence data (tibble will be converted with as.data.frame).
`clusters`	A data frame, containing three columns—the index of original data, group memberships, and group size. Or the cluster data frame returned by `seqcluster` function.
`var`	The list of columns containing the sequences. Default is NULL, i.e. all the columns. The function detects automatically whether the sequences are in the compressed (successive states in a character string) or extended format.
`alphabet`	Optional vector containing the alphabet (the list of all possible states). Use this option if some states in the alphabet don't appear in the data or if you want to reorder the states. The specified vector MUST contain AT LEAST all the states appearing in the data. It may possibly contain additional states not appearing in the data. If NULL, the alphabet is set to the distinct states appearing in the data as returned by the seqstatl function. See details.
`idxs`	A integer or an array of integers. The Default is 1:10, meaning replacing sequences in the 10 largest clusters with their representatives. If idxs=0, then all clusters will be replaces.
`...`	options passed to the `seqdef` function for handling input data that is not in STS format.

Value

An object of class `stslist`.

Function2: Get the cut height

`getheight()` returns the cut height for cutting a hierarchical clustering tree. The returned cut height `h` will satisfy that the first a few most frequent groups, specified by `idxs`, will represent at least `fq` of the total.

```
getheight(seqdistance, fq=0.3)
```

```
## $h
## [1] 5.5
##
## $frequency
## [1] 0.3300562
```

```
getheight(seqdistance, fq=0.3, idxs=1:100)
```

```
## There is no need to cluster.
```

```
## $h
## [1] 0
##
## $frequency
## [1] 0.4522472
```

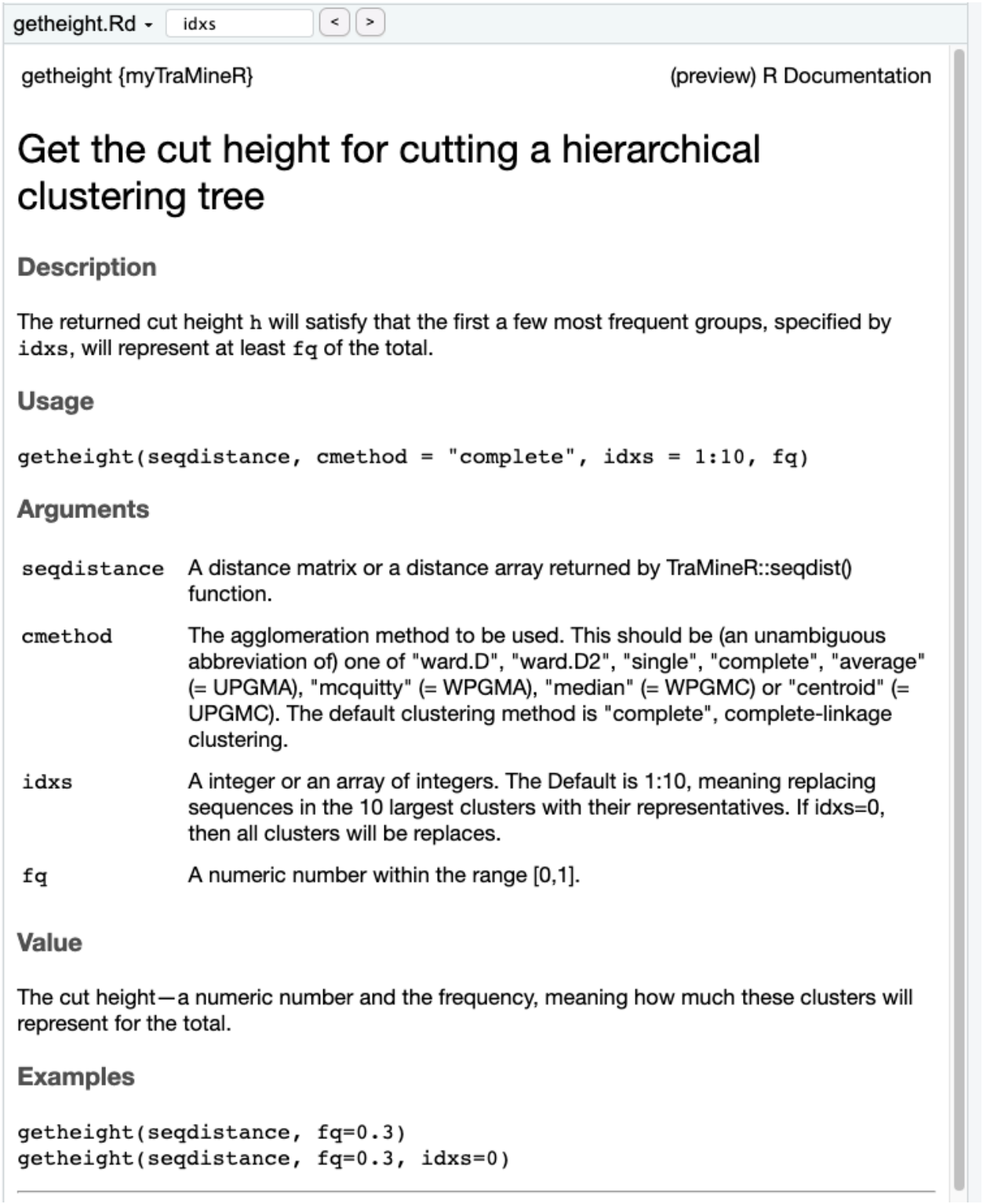

Figure 4: getheight