TraMineR: A toolbox for exploring and rendering sequences

Gilbert Ritschard

Institute for Demographic and Life Course Studies, University of Geneva
and NCCR LIVES: Overcoming vulnerability, life course perspectives
http://mephisto.unige.ch/traminer

Deuxièmes Rencontres R, Lyon, June 27-28, 2013

Outline

TraMineR

- Trajectory Miner in R: a toolbox for exploring, rendering and analyzing categorical sequence data

TraMineR, Why?

- TraMineR primary aim: Answer questions from social sciences
    - where sequences (succession of states or events) describe life trajectories
- Examples of questions:
    - Do life courses obey some social norm?
        - Which are the standard trajectories?
        - What kind of departures do we observe from those standards?
        - How do life course patterns evolve over time?
    - Why are some people more at risk to follow a chaotic trajectory or stay stuck in a state?
        - How does the trajectory complexity evolve across birth cohorts?
    - How is the life trajectory related to sex, social origin and other cultural factors?

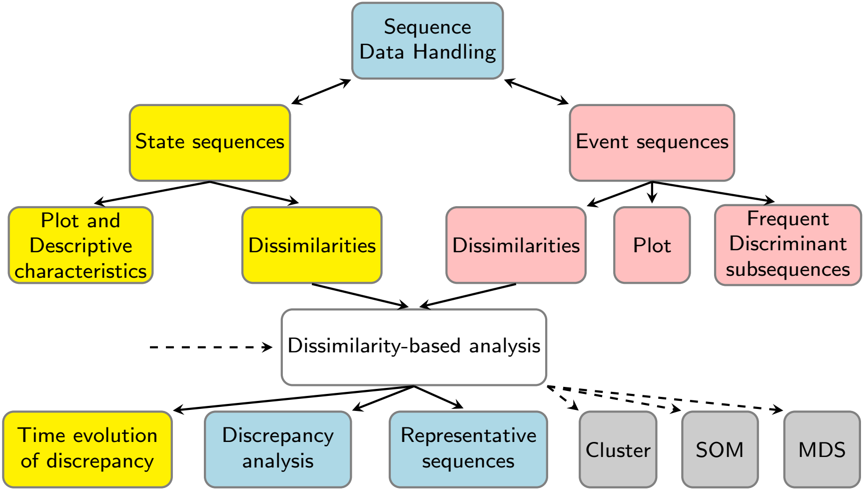
What TraMineR offers to answer those questions

- Various graphics and descriptive measures of individual sequences.
- Tools for computing pairwise dissimilarities between sequences which open access to plenty of advanced statistical and data analysis tools
  - Clustering and principal coordinate analysis (MDS)
  - Discrepancy analysis (ANOVA and regression trees)
  - Identification of representative sequences (trajectory-types)
  - ...
- Tools for mining frequent and discriminant event subsequences

TraMineR's features

- Handling of longitudinal data and conversion between various sequence formats
- Plotting sequences (distribution plot, frequency plot, index plot and more)
- Individual longitudinal characteristics of sequences (length, time in each state, longitudinal entropy, turbulence, complexity and more)
- Sequence of transversal characteristics by position (transversal state distribution, transversal entropy, modal state)
- Other aggregated characteristics (transition rates, average duration in each state, sequence frequency)
- Dissimilarities between pairs of sequences (Optimal matching, Longest common subsequence, Hamming, Dynamic Hamming, Multichannel and more)
- Representative sequences and discrepancy measure of a set of sequences
- ANOVA-like analysis and regression tree of sequences
- Rendering and highlighting frequent event sequences
- Extracting frequent event subsequences
- Identifying most discriminating event subsequences
- Association rules between subsequences

The TraMineR Swiss knife


Sequence Data Handling

State sequences

Event sequences

Plot and Descriptive characteristics

Dissimilarities

Dissimilarities

Plot

Frequent Discriminant subsequences

Dissimilarity-based analysis

Time evolution of discrepancy

Discrepancy analysis

Representative sequences

Cluster

SOM

MDS

Other programs for sequence analysis

- Optimize (Abbott, 1997)
  - Computes optimal matching distances
  - No longer supported
- TDA (Rohwer and Pötter, 2002)
  - free statistical software, computes optimal matching distances
- Stata, SQ-Ados (Brzinsky-Fay et al., 2006)
  - free, but licence required for Stata
  - optimal matching distances, visualization and a few more
  - See also the add-ons by Brendan Halpin `http://teaching.sociology.ul.ie/seqanal/`
- CHESA free program by Elzinga (2007)
  - Various metrics, including original ones based on non-aligning methods
  - Turbulence
- No equivalent package in R.
  - Packages such as those provided by Bioconductor are specifically devoted to biological issues.
  - `arulesSequences` mining of association rules (Zaki, 2001)

Sequence data

Sequence data

- Multiple cases (n cases)
- For each case a sorted list of (categorical) values

- Example:

$$\begin{array}{lllllll} 1: & a & a & d & d & c & \\ 2: & a & b & b & c & c & d \\ 3: & b & c & c & & & \\ . & . & . & . & . & & \end{array}$$

Longitudinal data

TraMineR is primarily intended for longitudinal data

Longitudinal data

- Repeated observations on units observed over time (Beck and Katz, 1995).
- "A dataset is longitudinal if it tracks the same type of information on the same subjects at multiple points in time". (http://www.caldercenter.org/whatis.cfm)
- "The defining feature of longitudinal data is that the multiple observations within subject can be ordered" (Singer and Willett, 2003)

Longitudinal data: Where do they come from?

- Individual follow-ups: Each important event is recorded as soon as it occurs (medical card, cellular phone, weblogs, ...).
- Panels: Periodic observation of same units
- Retrospective data (biography): Depends on interviewees' memory
- Matching data from different sources (successive censuses, tax data, social security, population registers, acts of marriages, acts of deaths, ...)

  Examples: Wanner and Delaporte (2001), censuses and population registers,

  Perroux and Oris (2005), 19th Century Geneva, censuses, acts of marriage, registers of deaths, register of migrations.

State sequences: an example

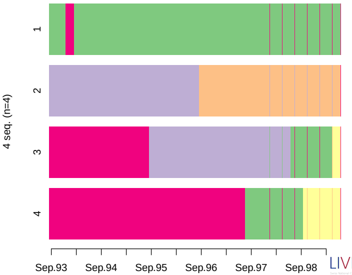
- Transition from school to work, (McVicar and Anyadike-Danes, 2002)

  Monthly states: EM = employment, TR = training, FE = further education, HE = higher education, SC = school, JL = joblessness

```
  Sequence
1 EM-EM-EM-EM-TR-TR-EM-EM-EM-EM-EM-EM-EM-EM-EM-EM-EM-EM-EM-EM-EM-EM-EM-EM-EM-E
2 FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-FE-F
3 TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-FE-F
4 TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-TR-T
```

- Compact representation

```
    Sequence
[1] (EM,4)-(TR,2)-(EM,64)
[2] (FE,36)-(HE,34)
[3] (TR,24)-(FE,34)-(EM,10)-(JL,2)
[4] (TR,47)-(EM,14)-(JL,9)
```



Types of categorical sequences

Nature of sequences

Depends on

- Chronological order?
  - If yes, we can study timing and duration.
- Information conveyed by position j in the sequence
  - If position is a time stamp, differences between positions reflect durations.
- Nature of the elements of the alphabet
  - states, transitions or events, letters, proteins, ...

State versus event sequences

- An important distinction for chronological sequences is between
  state sequences and event sequences
    - A State, such as ‘living with a partner’ or ‘being unemployed’, lasts the whole unit of time
    - An event, such as ‘moving in with a partner’ or ‘ending education’, does not last but provokes a state change, possibly in conjunction with other events.

State versus event sequences: examples

Time stamped events

Sandra	Ending education in 1980	Start working in 1980
Jack	Ending education in 1981	Start working in 1982

- There can be simultaneous events (see Sandra)
- Elements at same position do not occur at same time

State sequence view

year	1979	1980	1981	1982	1983
Sandra	Education	Education	Employed	Employed	Employed
Jack	Education	Education	Education	Unemployed	Employed

- Only one state at each observed time
- Position conveys time information: All states at position 2 are states in 1980.

Sequencing, timing and duration

- For chronological sequences (with time dimension)
- The following three aspects are of interest:
    - Sequencing: Order in which the different elements occur.
    - Timing: When do the different elements occur?
    - Duration: How long do we stay in the successive states?
- Event sequences: Most useful when concern is sequencing.
- State sequences: Most useful when concern is duration.
- Both may be useful for timing questions.

The ‘mvad’ data set

- McVicar and Anyadike-Danes (2002)’s study of school to work transition in Northern Ireland.
- dataset distributed with the TraMineR library.
- 712 cases (survey data).
- 72 monthly activity statuses (July 1993-June 1999)
- States are:

EM	Employment
FE	Further education
HE	Higher education
JL	Joblessness
SC	School
TR	Training.

- 14 additional (binary) variables
- The follow-up starts when respondents finished compulsory school (16 years old).

mvad variables

1	id	unique individual identifier
2	weight	sample weights
3	male	binary dummy for gender, 1=male
4	catholic	binary dummy for community, 1=Catholic
5	Belfast	binary dummies for location of school, one of five Education and Library Board areas in Northern Ireland
6	N.Eastern	"
7	Southern	"
8	S.Eastern	"
9	Western	"
10	Grammar	binary dummy indicating type of secondary education, 1=grammar school
11	funemp	binary dummy indicating father's employment status at time of survey, 1=father unemployed
12	gcse5eq	binary dummy indicating qualifications gained by the end of compulsory education, 1=5+ GCSEs at grades A-C, or equivalent
13	fmpr	binary dummy indicating SOC code of father's current or most recent job,1=SOC1 (professional, managerial or related)
14	livboth	binary dummy indicating living arrangements at time of first sweep of survey (June 1995), 1=living with both parents
15	jul93	Monthly Activity Variables are coded 1-6, 1=school, 2=FE, 3=employment, 4=training, 5=joblessness, 6=HE
	⋮	"
86	jun99	"

The mvad sequences are in STS form

- The `mvad` sequences are organized in STS form, i.e., each sequence is given as a (row) vector of consecutive states.

```
head(mvad[, 17:22])
```

```
       Sep.93     Oct.93     Nov.93     Dec.93   Jan.94   Feb.94
1  employment employment employment employment training training
2          FE         FE         FE         FE       FE       FE
3    training   training   training   training training training
4    training   training   training   training training training
5          FE         FE         FE         FE       FE       FE
6 joblessness   training   training   training training training
```

- There are many other ways of organizing sequences data and TraMineR supports most of them.

Creating the state sequence object

- General TraMineR philosophy: Storing all reusable information on a set of sequences into a sequence object.
- Most TraMineR functions for state sequences require a state sequence object as input argument.
- The state sequence object contains
  - the sequences
  - and their attributes (alphabet, labels, colors, weights, ...)
- Hence, we first have to create this object

Starting TraMineR and creating a state sequence object

- Load `TraMineR` and the `mvad` data.

```
library(TraMineR)
data(mvad)
```

- Check the alphabet (from Sept 93 to June 99; i.e., positions 17 to 86: We skip July-August 93)

```
(mvad.alph <- seqstatl(mvad[, 17:86]))
```

```
[1] "employment"  "FE"           "HE"           "joblessness" "school"
[6] "training"
```

- Create the ‘state sequence’ object

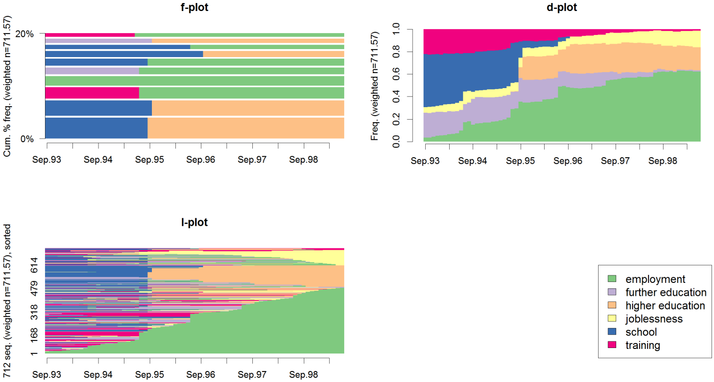
```
mvad.lab <- c("employment", "further education", "higher education",
     "joblessness", "school", "training")
mvad.shortlab <- c("EM", "FE", "HE", "JL", "SC", "TR")
mvad.seq <- seqdef(mvad[, 17:86], alphabet = mvad.alph,
     states = mvad.shortlab, labels = mvad.lab, weights = mvad$weight,
     xtstep = 6)
```

Main sequence object attributes and seqdef arguments

Attribute name	Description	Argument	Default	Retrieve/Set
	input format	`informat=`	`"STS"`	
alphabet	list of states	`states=`	from input data	`alphabet()`
cpal	color palette	`cpal=`	from RColorBrewer	`cpal()`
labels	long state labels	`labels=`	from input data	`stlab()`
cnames	position names	`cnames=`	from input data	`names()`
xtstep	jumps between tick marks	`xtstep=`	`1`	
row.names	row (sequence) labels	`id=`	from input data	`rownames()`
weights	optional case weights	`weights=`	`NULL`	
	missing handling	`left=`	`NA`	
	"	`gaps=`	`NA`	
	"	`right=`	`"DEL"`	

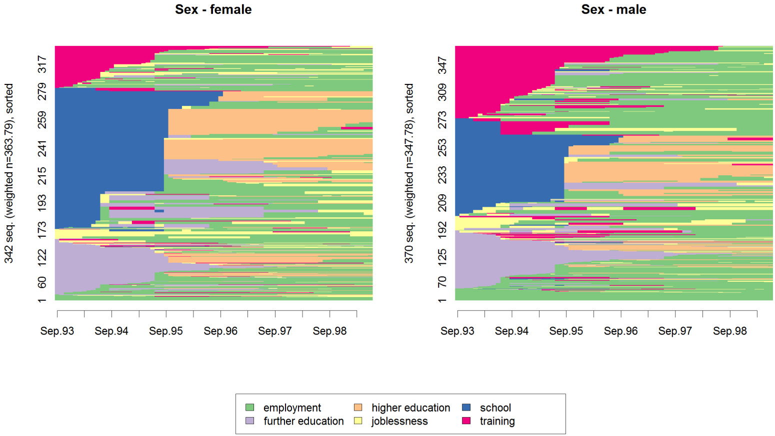
Rendering sequences

```
seqfplot(mvad.seq, withlegend = FALSE, title = "f-plot", border = NA)
seqdplot(mvad.seq, withlegend = FALSE, title = "d-plot", border = NA)
seqIplot(mvad.seq, withlegend = FALSE, title = "I-plot", sortv = "from.end")
seqlegend(mvad.seq, position = "bottomright", fontsize = 1.2)
```



Rendering sequences by group (sex)

```
seqIplot(mvad.seq, group = mvad$male, sortv = "from.start",
    title = "Sex")
```



Characterizing set of sequences

- Sequence of cross-sectional measures (modal state, between entropy, ...)

id	t_1	t_2	t_3	$\cdots$
1	B	B	D	$\cdots$
2	A	B	C	$\cdots$
3	B	B	A	$\cdots$

- Summary of longitudinal measures (within entropy, transition rates, mean duration ...)

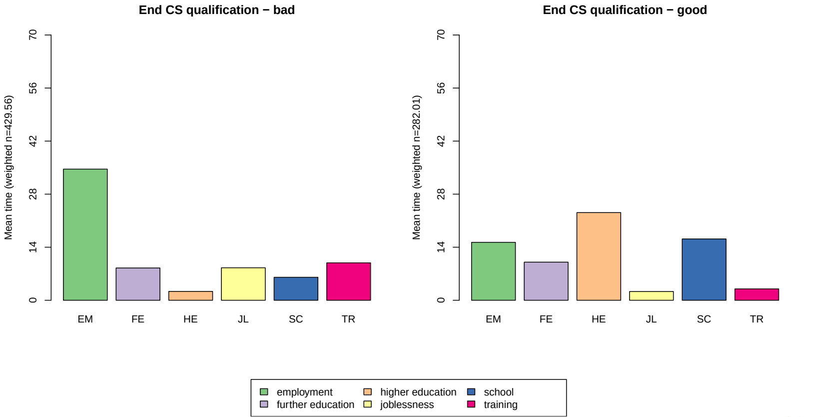
id	t_1	t_2	t_3	$\cdots$
1	B	B	D	$\cdots$
2	A	B	C	$\cdots$
3	B	B	A	$\cdots$

- Other global characteristics: sequence medoid, diversity of sequences, ...

Mean time in each state

by qualification gained at end of compulsory school

```
seqmtplot(mvad.seq, group = mvad$gcse5eq, title = "End CS qualification")
```



Sequence of cross-sectional distributions

For bad qualification at end of compulsory school, 9 months

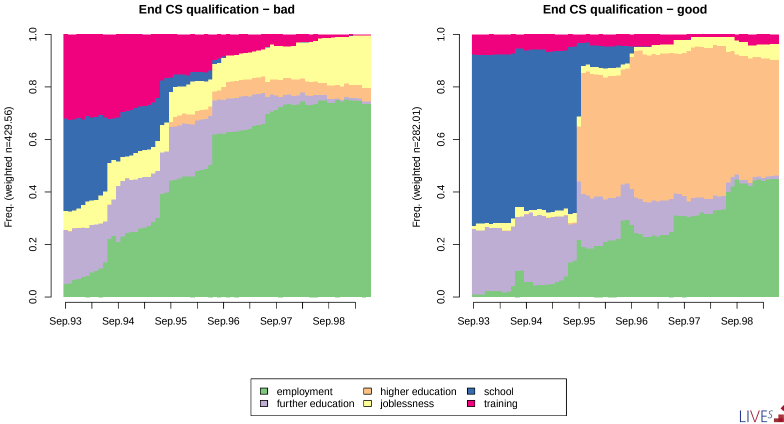
```
seqstatd(mvad.seq[mvad$gcse5eq == "bad", 6:15])
```

```
    [State frequencies]
   Feb.94 Mar.94 Apr.94 May.94 Jun.94 Jul.94 Aug.94 Sep.94 Oct.94 Nov.94
EM   0.08  0.094  0.100   0.11   0.13   0.22   0.23  0.211  0.231  0.244
FE   0.18  0.181  0.176   0.17   0.16   0.13   0.14  0.212  0.211  0.209
HE   0.00  0.000  0.000   0.00   0.00   0.00   0.00  0.000  0.000  0.000
JL   0.10  0.093  0.093   0.11   0.11   0.16   0.15  0.094  0.091  0.084
SC   0.33  0.316  0.316   0.31   0.28   0.17   0.16  0.167  0.171  0.171
TR   0.31  0.316  0.315   0.31   0.32   0.32   0.32  0.316  0.295  0.292

     [Valid states]
   Feb.94 Mar.94 Apr.94 May.94 Jun.94 Jul.94 Aug.94 Sep.94 Oct.94 Nov.94
N     430    430    430    430    430    430    430    430    430    430

     [Entropy index]
   Feb.94 Mar.94 Apr.94 May.94 Jun.94 Jul.94 Aug.94 Sep.94 Oct.94 Nov.94
H    0.82   0.83   0.83   0.84   0.85   0.87   0.87   0.86   0.86   0.86
```

Sequence of cross-sectional distributions (chronogram)

by qualification gained at end of compulsory school

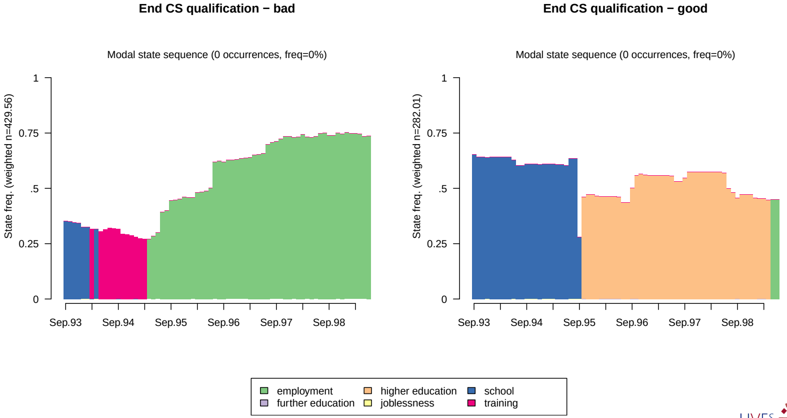
```
seqdplot(mvad.seq, group = mvad$gcse5eq, title = "End CS qualification",
     border = NA)
```



Sequence of modal states

by qualification gained at end of compulsory school

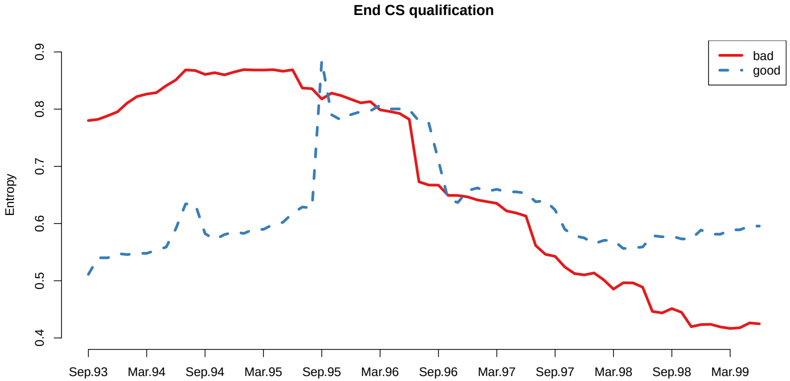
```
seqmsplot(mvad.seq, group = mvad$gcse5eq, title = "End CS qualification",
     border = NA)
```



Cross-sectional entropies

Time evolution of the Cross-sectional state diversity

```
seqplot.tentrop(mvad.seq, title = "End CS qualification",
     group = mvad$gcse5eq)
```



Longitudinal Characteristics

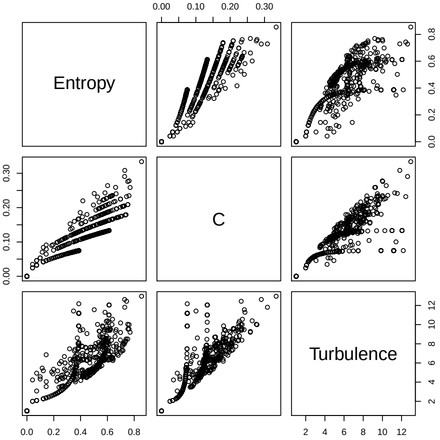
- Characteristics of individual sequences

`seqlength()`	length of the sequence
`seqtransn()`	number of transitions
`seqsubsn()`	number of sub-sequences
`seqdss()`	list of the distinct successive states (DSS)
`seqdur()`	list of the durations in the states of the DSS
`seqistatd()`	time in each state (longitudinal distribution)
`seqient()`	Longitudinal entropy
`seqST()`	Turbulence (Elzinga and Liefbroer, 2007)
`seqici()`	Complexity index (Gabadinho et al., 2011)

Complexity of the sequences

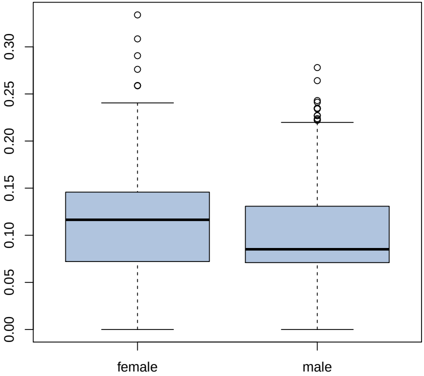
- To evaluate the complexity of a sequence we may consider
- Longitudinal entropy
  - does not account for the sequencing of the states (AABB and ABAB have same entropy)
- Turbulence (Elzinga and Liefbroer, 2007)
  - composite measure based on
    - the number of sub-sequences of the DSS sequence
    - the variance of the durations of the successive states
  - sensitive to state sequencing
- Index of complexity (Gabadinho et al., 2010, 2011)
  - composite measure based on
    - the number of transitions
    - the longitudinal entropy
  - sensitive to state sequencing

Comparing the measures



Distribution of complexity by sex

```
boxplot(mvad.cplx ~ mvad$male, col = "lightsteelblue")
```



Summary of available distances

Distance	Method	Position-wise	Additional arguments
Count of common attributes			
Simple Hamming	`HAM`	Yes	
Longest Common Prefix	`LCP`	Yes	
Longest Common Suffix	`RLCP`	Yes	
Longest Common Subsequence	`LCS`	No	
Edit distances			
Optimal Matching	`OM`	No	Insertion/deletion costs (`indel`) and substitution costs matrix (`sm`)
Hamming	`HAM`	Yes	substitution costs matrix (`sm`)
Dynamic Hamming	`DHD`	Yes	substitution costs matrix (`sm`)

Other distances

- There exist many other distances which will be made available in TraMineR in a near future.
    - Distances based on counts of common subsequences (Elzinga, 2003; Liefbroer and Elzinga, 2012; Oh and Kim, 2004)
    - Euclidean or Chi-squared distances between within-sequence state distributions, including over successive periods (Deville and Saporta, 1983; Grelet, 2002)
    - Variants of Optimal Matching (Hollister, 2009; Halpin, 2010)
    - OM of transitions instead of states (Biemann, 2011)

Dissimilarity matrix

- TraMineR provides the `seqdist` function

```
## OM distances with custom indel and substitution
## costs used by McVicar and Anyadike-Danes (2012).
subm.custom  <- matrix(
         c(0,1,1,2,1,1,
           1,0,1,2,1,2,
           1,1,0,3,1,2,
           2,2,3,0,3,1,
           1,1,1,3,0,2,
           1,2,2,1,2,0),
           nrow = 6, ncol = 6, byrow = TRUE,
           dimnames = list(mvad.shortlab, mvad.shortlab))
mvad.dist <- seqdist(mvad.seq, method="OM", indel=4, sm=subm.custom)
dim(mvad.dist)
```

```
[1] 712 712
```

Dissimilarity matrix

```
print(mvad.seq[1:4, ], format = "SPS")
```

```
    Sequence
[1] (EM,4)-(TR,2)-(EM,64)
[2] (FE,36)-(HE,34)
[3] (TR,24)-(FE,34)-(EM,10)-(JL,2)
[4] (TR,47)-(EM,14)-(JL,9)
```

```
mvad.dist[1:4, 1:6]
```

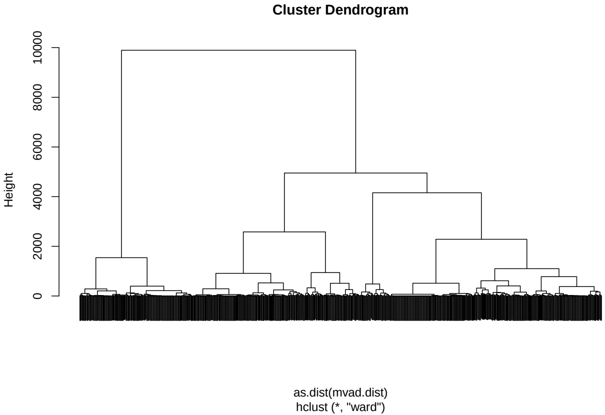
```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0   72   60   63   72   33
[2,]   72    0   86  135   11  104
[3,]   60   86    0   71   97   49
[4,]   63  135   71    0  135   32
```

Cluster analysis

- Can run any clustering method which accepts a dissimilarity matrix as input.
- Many solutions in R:
- For hierarchical clustering
  - `hclust()` base function (can account for weights)
  - Package `cluster` (does not support weights!):
    - `agnes()`: agglomerative nesting (average, UPGMA WPGMA, ward, beta-flexible, ...)
    - `diana()`: divisive partitioning
- For PAM (partitioning around medoids) and other direct methods
  - Packages: `cluster`, `fastclust`, `flashClust`, ...
  - `WeightedCluster` (Studer, 2013)

Example: Hierarchical clustering (Ward)

```
mvad.clusterward <- hclust(as.dist(mvad.dist), method = "ward",
     members = mvad$weight)
plot(mvad.clusterward, labels = FALSE)
```



PAM clustering

- PAM much faster, but must set *a priori* number k of clusters.
- `WeightedCluster` offers nice tools to help selecting k.
- $k = 4$ was found to be good choice.
- PAM with function `wcKMedoids` from `WeightedCluster`

```
library(WeightedCluster)
set.seed(4)
pam.mvad <- wcKMedoids(mvad.dist, k = 4, weight = mvad$weight)
```

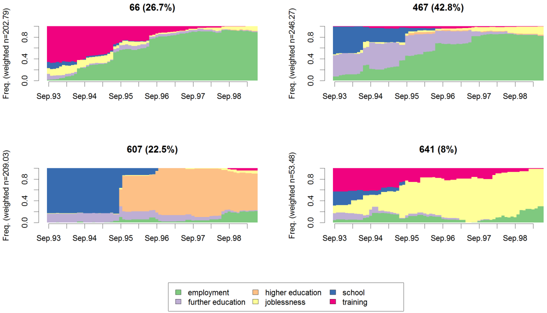
- Cluster membership is in `pam.mvad$clustering`

```
mvad.cl4 <- pam.mvad$clustering
table(mvad.cl4)
```

```
mvad.cl4
 66 467 607 641
190 305 160  57
```

Labeling the PAM clusters

```
seqdplot(mvad.seq, group = group.p(mvad.cl4), border = NA)
```

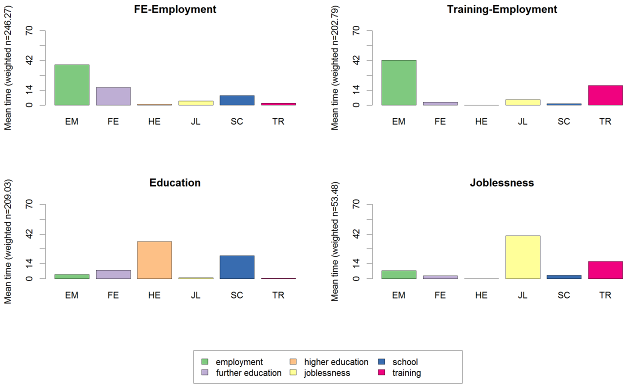


- Rearranging cluster order and defining labels

```
cl4.labels <- c("FE-Employment", "Training-Employment", "Education",
    "Joblessness")
mvad.cl4.factor <- factor(mvad.cl4, levels = c(467, 66, 607,
    641), labels = cl4.labels)
```

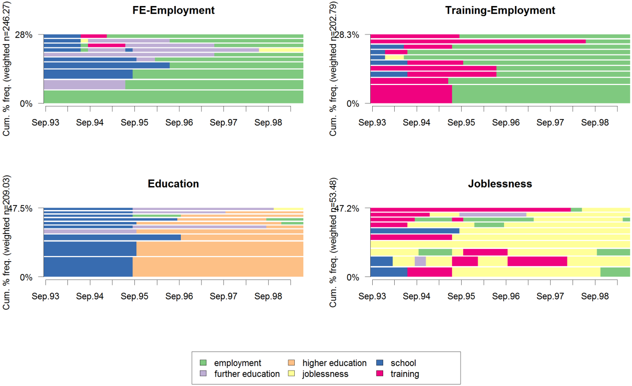
Mean time in each state

`seqmtplot(mvad.seq, group = mvad.cl4.factor)`



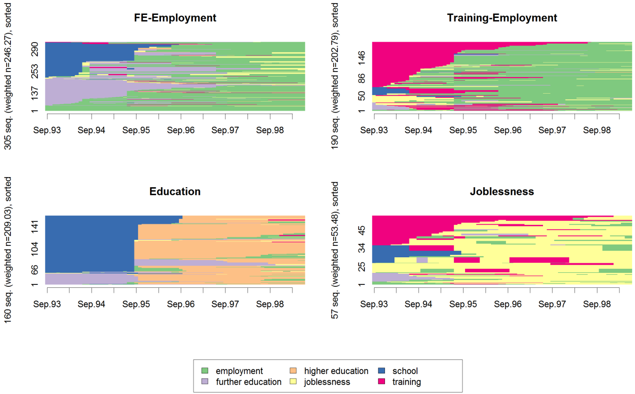
Most frequent sequences

```
seqfplot(mvad.seq, group = mvad.cl4.factor, border = NA)
```



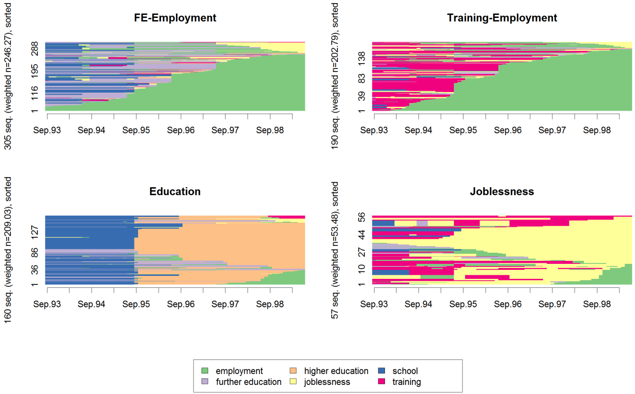
Individual sequences (sorted by states from start)

```
seqIplot(mvad.seq, group = mvad.cl4.factor, sortv = "from.start")
```



Sorted by states from the end

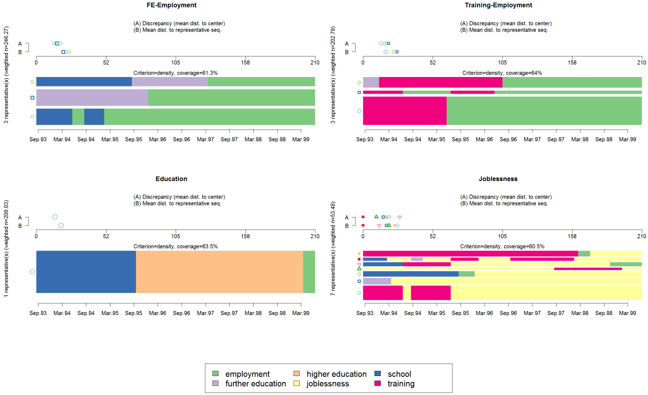
```
seqIplot(mvad.seq, group = mvad.cl4.factor, sortv = "from.end")
```



Representative sequences (Gabadinho et al., 2011)

Smallest set of patterns with given percentage of sequences in their neighborhood

```
seqrplot(mvad.seq, group = mvad.cl4.factor, dist.matrix = mvad.dist,
    trep = 0.6, sim = 0.15, border = NA, cex.legend = 1.5)
```



Discrepancy of sequences

- Sum of squares SS can be expressed in terms of distances between pairs

$$\begin{aligned} SS &= \sum_{i=1}^{n}(y_i-\bar{y})^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(y_i-y_j)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=i+1}^{n}d_{ij} \end{aligned}$$

- Setting d_{ij} equal to OM, LCP, LCS ... distance, we get SS.
- From which we can measure the dispersion with the pseudo-variance SS/n.
- And run ANOVA analyses (Studer et al., 2011, 2010, 2009).

Computing the dispersion

- For the whole set of sequences

```
dissvar(mvad.dist)
```

```
[1] 32.06
```

- By cluster (`dissvar.grp` from library `TraMineRextras`)

```
data.frame(Dispersion = dissvar.grp(mvad.dist, group = mvad.cl4.factor))
```

```
                    Dispersion
FE-Employment            18.60
Training-Employment      17.89
Education                15.90
Joblessness              27.14
```

Analysis of sequence discrepancy

- Running an ANOVA-like analysis for `gcse5eq`

```
da <- dissassoc(mvad.dist, group = mvad$gcse5eq, R = 1000)

print(da)
```

ANOVA output

```
Pseudo ANOVA table:
         SS  df     MSE
Exp     1952   1 1952.4
Res    20871 710   29.4
Total  22823 711   32.1

Test values  (p-values based on 1000 permutation):
                 t0 p.value
Pseudo F   66.41934   0.001
Pseudo Fbf 67.37188   0.001
Pseudo R2   0.08555   0.001
Bartlett    0.14693   0.339
Levene      0.77397   0.403

Inconclusive intervals:
0.00383  <  0.01  <  0.0162
0.03649  <  0.05  <  0.0635

Discrepancy per level:
        n discrepancy
bad   452       29.76
good  260       28.53
Total 712       32.06
```

Growing a sequence regression tree

```
dt <- seqtree(mvad.seq ~ male + Grammar + funemp + gcse5eq +
     fmpr + livboth, weighted = FALSE, data = mvad, diss = mvad.dist,
     R = 5000)
```

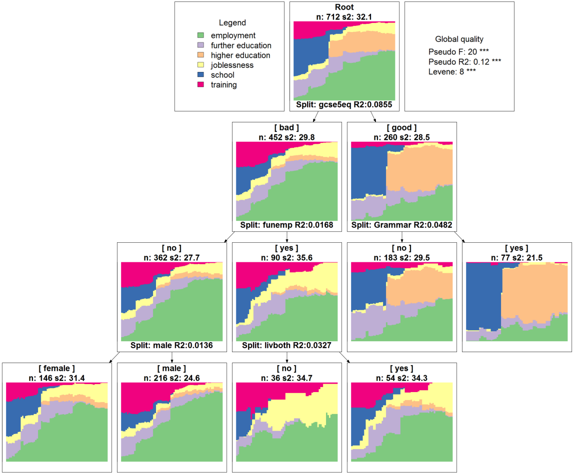
Graphical tree

- The graphical rendering uses Graphviz `http://www.graphviz.org/`

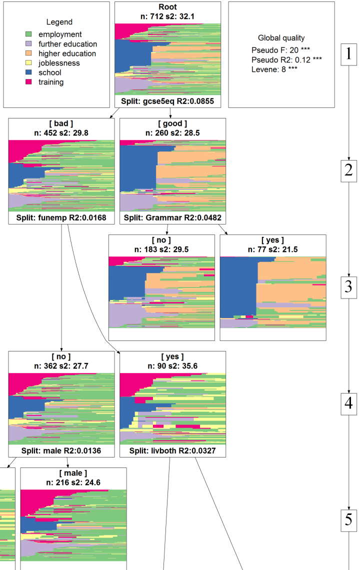
```
seqtreedisplay(dt, filename = "fg_mvadseqtree.png",
      type = "d", border = NA)
```

- The plot is produced as a png file and displayed with the default program associated to this extension.

Graphical Tree



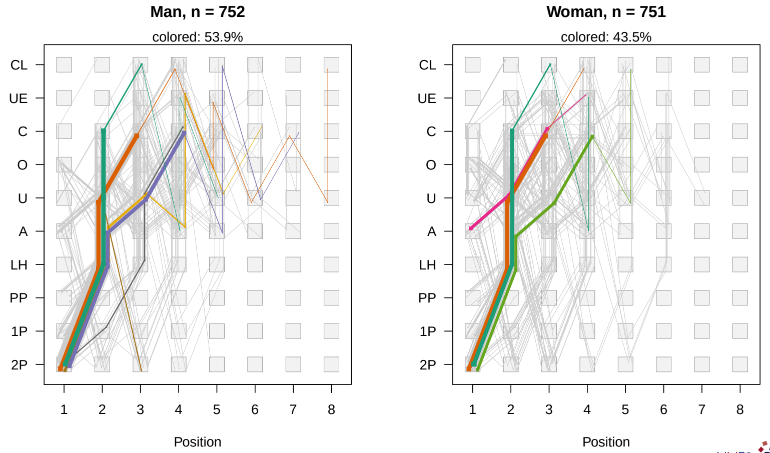
Graphical Tree, using I-plots and `showdepth=TRUE`



Rendering event sequences

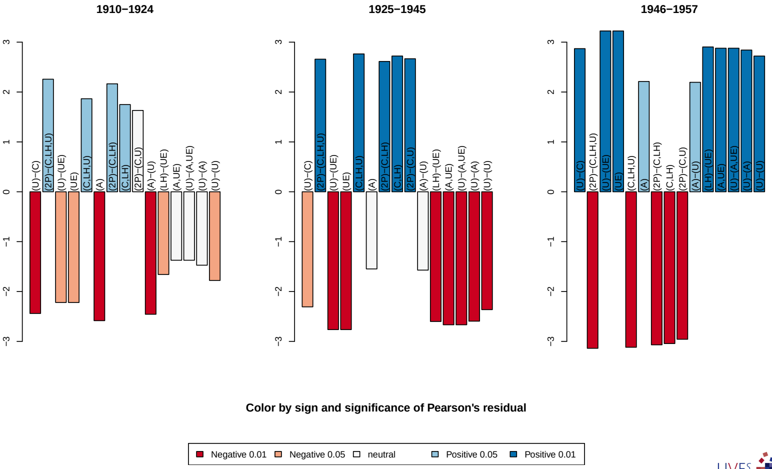
Swiss cohabitational trajectories, data from 2002 SHP biographical survey

Plot of event sequences, patterns with at least 5% support are colored



Event sequences: discriminating sub-sequences
By birth cohort

Pearson's residuals by decreasing discrimination power



TraMineR was made possible thanks to SNF

- Developed within the SNF (Swiss National Fund for Scientific Research) project Mining event histories: Towards new insights on personal Swiss life courses 1/2007-1/2011
- ... development goes on within IP 14 methodological module of the NCCR LIVES: Overcoming vulnerability: Life course perspectives (`http://www.lives-nccr.ch`) .

TraMineR, Who?

- Under supervision of a scientific committee:
  - Gilbert Ritschard (Statistics for social sciences)
  - Alexis Gabadinho (Demography)
  - Nicolas S. Müller (Sociology, Computer science)
  - Matthias Studer (Economics, Sociology)
- Additional members of the development team:
  - Reto Bürgin (Statistics)
  - Emmanuel Rousseaux (KDD and Computer science)

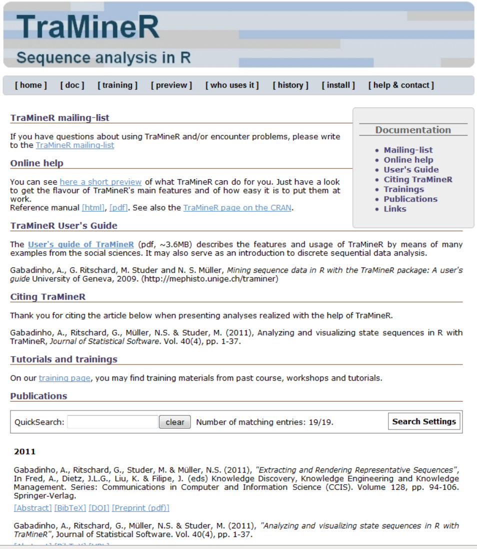
both PhD students within NCCR LIVES IP-14

Other packages by the TraMineR team

- `TraMineRextras` additional less stabilized functions
- `PST` (Probability suffix trees) by Alexis Gabadinho
- `WeightedCluster` (Studer, 2013)
- `Dataset` (handling and documenting survey data sets) by Emmanuel Rousseaux

Documentation

- The success of TraMineR is largely due to the documentation.
- Web page `http://mephisto.unige.ch/traminer`
    - News (new release, ...)
    - Preview
    - Documentation:
        - User's guide (about 120 pages)
        - Tutorials
        - Web page (html) of the Reference manual
        - Papers by the TraMineR team
        - Publications by TraMineR users
    - Information about forthcoming training courses



TraMineR mailing-list

If you have questions about using TraMineR and/or encounter problems, please write to the TraMineR mailing-list

Online help

You can see here a short preview of what TraMineR can do for you. Just have a look to get the flavour of TraMineR's main features and of how easy it is to put them at work.
Reference manual [html], [pdf]. See also the TraMineR page on the CRAN.

TraMineR User's Guide

The **User's guide of TraMineR** (pdf, ~3.6MB) describes the features and usage of TraMineR by means of many examples from the social sciences. It may also serve as an introduction to discrete sequential data analysis.

Gabadinho, A., G. Ritschard, M. Studer and N. S. Müller, *Mining sequence data in R with the TraMineR package: A user's guide* University of Geneva, 2009. (http://mephisto.unige.ch/traminer)

Citing TraMineR

Thank you for citing the article below when presenting analyses realized with the help of TraMineR.

Gabadinho, A., Ritschard, G., Müller, N.S. & Studer, M. (2011), Analyzing and visualizing state sequences in R with TraMineR, *Journal of Statistical Software*. Vol. 40(4), pp. 1-37.

Tutorials and trainings

On our training page, you may find training materials from past course, workshops and tutorials.

Publications

Number of matching entries: 19/19.

2011

Gabadinho, A., Ritschard, G., Studer, M. & Müller, N.S. (2011), *"Extracting and Rendering Representative Sequences"*, In Fred, A., Dietz, J.L.G., Liu, K. & Filipe, J. (eds) Knowledge Discovery, Knowledge Engineering and Knowledge Management. Series: Communications in Computer and Information Science (CCIS). Volume 128, pp. 94-106. Springer-Verlag.
[Abstract] [BibTeX] [DOI] [Preprint (pdf)]

Gabadinho, A., Ritschard, G., Müller, N.S. & Studer, M. (2011), *"Analyzing and visualizing state sequences in R with TraMineR"*, Journal of Statistical Software. Vol. 40(4), pp. 1-37.

R-forge page

- TraMineR page on R-forge (`https://r-forge.r-project.org/projects/traminer/`)
- where you
  - find the development version
  - can post bug reports,
- Can join the discussion list (but broken search!)

Where asking for help?

- Best place for help is StackExchange
- There are `traminer` tags on
    - StackOverflow (SO)
      http://stackoverflow.com/questions/tagged/traminer
      for TraMineR R-code related questions
    - CrossValidated (CV)
      http://stats.stackexchange.com/questions/tagged/traminer
      for questions regarding statistical interpretation and methodological issues

Thank you!

References I

Abbott, A. (1997). Optimize. http://home.uchicago.edu/˜aabbott/om.html.

Aisenbrey, S. and A. E. Fasang (2010). New life for old ideas: The “second wave” of sequence analysis bringing the “course” back into the life course. *Sociological Methods and Research 38*(3), 430–462.

Beck, N. and J. N. Katz (1995). What to do (and not to do) with time-series cross-section data. *American Political Science Review 89*, 634–647.

Biemann, T. (2011). A transition-oriented approach to optimal matching. *Sociological Methodology 41*(1), 195–221.

Billari, F. C. (2001). The analysis of early life courses: Complex description of the transition to adulthood. *Journal of Population Research 18*(2), 119–142.

Brzinsky-Fay, C., U. Kohler, and M. Luniak (2006). Sequence analysis with Stata. *The Stata Journal 6*(4), 435–460.

Deville, J.-C. and G. Saporta (1983). Correspondence analysis with an extension towards nominal time series. *Journal of Econometrics 22*, 169–189.

Elzinga, C. H. (2003). Sequence similarity: A non-aligning technique. *Sociological Methods and Research 31*, 214–231.

References II

Elzinga, C. H. (2007). CHESA 2.1 User manual. User guide, Dept of Social Science Research Methods, Vrije Universiteit, Amsterdam.

Elzinga, C. H. and A. C. Liefbroer (2007). De-standardization of family-life trajectories of young adults: A cross-national comparison using sequence analysis. *European Journal of Population 23*, 225–250.

Gabadinho, A., G. Ritschard, N. S. Müller, and M. Studer (2011). Analyzing and visualizing state sequences in R with TraMineR. *Journal of Statistical Software 40*(4), 1–37.

Gabadinho, A., G. Ritschard, M. Studer, and N. S. Müller (2009). Mining sequence data in R with the TraMineR package: A user's guide. Technical report, Department of Econometrics and Laboratory of Demography, University of Geneva, Geneva.

Gabadinho, A., G. Ritschard, M. Studer, et N. S. Müller (2010). Indice de complexité pour le tri et la comparaison de séquences catégorielles. *Revue des nouvelles technologies de l'information RNTI E-19*, 61–66.

References III

Gabadinho, A., G. Ritschard, M. Studer, et N. S. Müller (2011). Extracting and rendering representative sequences. In A. Fred, J. L. G. Dietz, K. Liu, et J. Filipe (Eds.), *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, Volume 128 of *Communications in Computer and Information Science (CCIS)*, pp. 94–106. Springer-Verlag.

Grelet, Y. (2002). Des typologies de parcours : Méthodes et usages. Notes de travail Génération 92, Céreq, Paris.

Halpin, B. (2010). Optimal matching analysis and life-course data : The importance of duration. *Sociological Methods and Research 38*(3), 365–388.

Hollister, M. (2009). Is Optimal Matching Suboptimal? *Sociological Methods Research 38*(2), 235–264.

Liefbroer, A. C. and C. H. Elzinga (2012). Intergenerational transmission of behavioural patterns: How similar are parents' and children's demographic trajectories? *Advances in Life Course Research 17*, 1–10.

McVicar, D. and M. Anyadike-Danes (2002). Predicting successful and unsuccessful transitions from school to work using sequence methods. *Journal of the Royal Statistical Society A 165*(2), 317–334.

References IV

Oh, S.-J. and J.-Y. Kim (2004). A hierarchical clustering algorithm for categorical sequence data. *Information Processing Letters 91*(3), 135–140.

Perroux, O. et M. Oris (2005). Présentation de la base de données de la population de Genève de 1816 à 1843. Séminaire statistique sciences sociales, Université de Genève.

Ritschard, G., A. Gabadinho, N. S. Müller, and M. Studer (2008). Mining event histories: A social science perspective. *International Journal of Data Mining, Modelling and Management 1*(1), 68–90.

Rohwer, G. and U. Pötter (2002). TDA user's manual. Software, Ruhr-Universität Bochum, Fakultät für Sozialwissenschaften, Bochum.

Singer, J. D. and J. B. Willett (2003). *Applied longitudinal data analysis: Modeling change and event occurrence*. Oxford: Oxford University Press.

Studer, M. (2013). Weightedcluster library manual: A practical guide to creating typologies of trajectories in the social sciences with R. LIVES Working Papers 24, NCCR LIVES, Switzerland.

References V

Studer, M., G. Ritschard, A. Gabadinho, et N. S. Müller (2009). Analyse de dissimilarités par arbre d'induction. *Revue des nouvelles technologies de l'information RNTI E-15*, 7–18.

Studer, M., G. Ritschard, A. Gabadinho, et N. S. Müller (2010). Discrepancy analysis of complex objects using dissimilarities. In F. Guillet, G. Ritschard, D. A. Zighed, et H. Briand (Eds.), *Advances in Knowledge Discovery and Management*, Volume 292 of *Studies in Computational Intelligence*, pp. 3–19. Berlin : Springer.

Studer, M., G. Ritschard, A. Gabadinho, et N. S. Müller (2011). Discrepancy analysis of state sequences. *Sociological Methods and Research 40*(3), 471–510.

Wanner, P. et E. Delaporte (2001). Reconstitution de trajectoires de vie à partir des données de l'état civil (BEVNAT). une étude de faisabilité. Rapport de recherche, Forum Suisse des Migrations.

Widmer, E. and G. Ritschard (2009). The de-standardization of the life course: Are men and women equal? *Advances in Life Course Research 14*(1-2), 28–39.

References VI

Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning 42*(1/2), 31–60.