

# Sequence Analysis with TraMineR

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>

Summer School in Longitudinal and Life Course Research,  
Oxford, 2nd-6th September 2013

# Outline

- 1 TraMineR, What is it?
- 2 Basics of sequence analysis with TraMineR
- 3 More about TraMineR

## TraMineR

- **T**rajectory **M**iner 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 an unwanted state?
    - How does the trajectory complexity evolve across birth cohorts?
  - How is the life trajectory related to sex, social origin and other cultural factors?

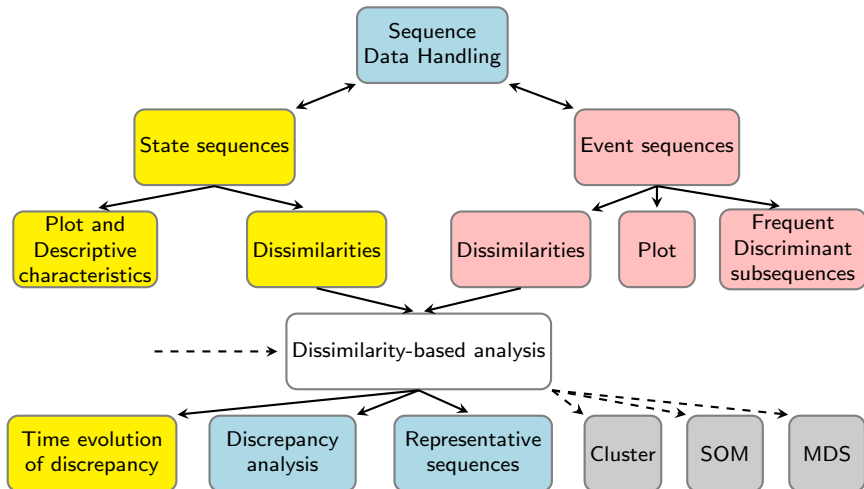
# What TraMineR offers to answer those questions

- 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



# 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)



# TraMineR: Where and why in R?

- Package for the free open source R statistical environment
  - R and TraMineR freely available from the **CRAN** (Comprehensive R Archive Network) <http://cran.r-project.org>
- TraMineR runs in R, it can straightforwardly be combined with other R commands and libraries. For example:
  - dissimilarities obtained with TraMineR can be inputted to already optimized processes for clustering, MDS, self-organizing maps, ...
  - TraMineR 's plots can be used to render clustering results;
  - complexity indexes can be used as dependent or explanatory variables in linear and non-linear regression, ...

# Installing TraMineR

- Stable version from the **CRAN**
  - Check that you have the latest version of R (upgrade if necessary)
  - Start R and run following command from the console

```
install.packages("TraMineR", dependencies = TRUE)
```

- On Linux, you may need to first install additional components.
- Development version from **R-Forge**
  - Command

```
source("http://mephisto.unige.ch/traminer/install-devel.R")
```

- also installs **TraMineRextras**, **WeightedCluster**, dependencies and a few other useful packages

## 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** (XX) 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 other ways of organizing sequences data (SPS or XT, SPELL, Person-Period, ...) and TraMineR supports most of them.

# Creating the state 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 <- seqstat1(mvad[, 17:86]))

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

- Create the 'state sequence' object

```
## mvad.lab <- seqstat1(mvad[,17:86])
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, labels = mvad.lab,
                  states = mvad.shortlab, weights = mvad$weight, xtstep = 6)
```

# Main sequence object attributes and seqdef arguments

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

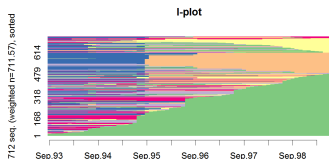
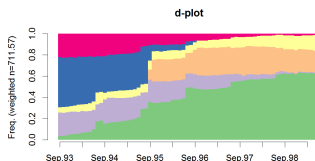
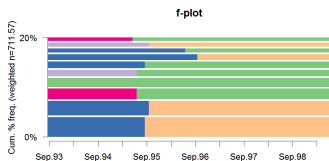


# Rendering sequences

```

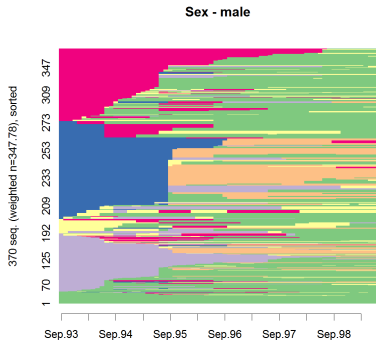
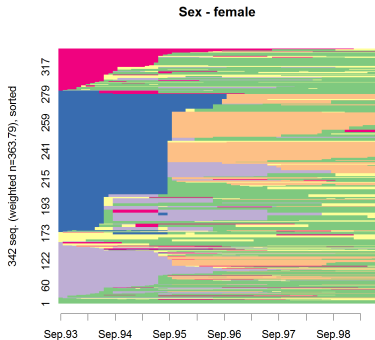
par(mfrow = c(2, 2))
seqfplot(mvad.seq, withlegend = FALSE, title = "f-plot", border = NA)
seqdplot(mvad.seq, withlegend = FALSE, title = "d-plot", border = NA)
seqlplot(mvad.seq, withlegend = FALSE, title = "I-plot", sortv = "from.end")
seqlegend(mvad.seq, position = "bottomright", fontsize = 1.2)

```



# Rendering sequences by group (sex)

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



employment	higher education	school
further education	joblessness	training

# Characterizing set of sequences

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

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

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

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

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

# Transition rates

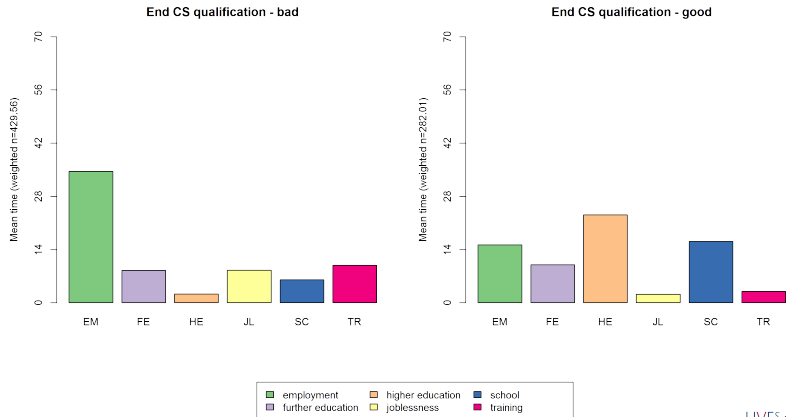
```
round(trate <- seqrate(mvad.seq), 3)
```

##		[-> EM]	[-> FE]	[-> HE]	[-> JL]	[-> SC]	[-> TR]
##	[EM ->]	0.986	0.002	0.003	0.007	0.000	0.002
##	[FE ->]	0.027	0.950	0.007	0.011	0.001	0.003
##	[HE ->]	0.010	0.000	0.988	0.001	0.000	0.001
##	[JL ->]	0.037	0.012	0.002	0.938	0.001	0.010
##	[SC ->]	0.012	0.008	0.019	0.007	0.950	0.004
##	[TR ->]	0.037	0.004	0.000	0.015	0.001	0.944

# Mean time in each state

by qualification gained at end of compulsory school

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



# Sequence of transversal distributions

For bad qualification at end of compulsory school, 9 months

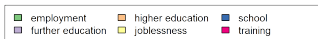
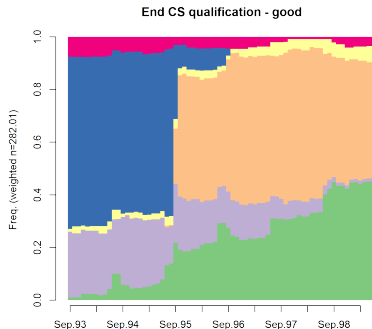
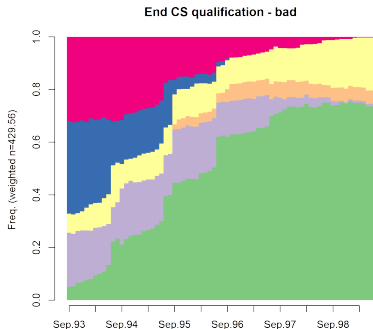
```
seqstatd(mvad.seq[mvad$gcse5eq == "bad", 6:15])
```

```
##      [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 transversal distributions (chronogram)

by qualification gained at end of compulsory school

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

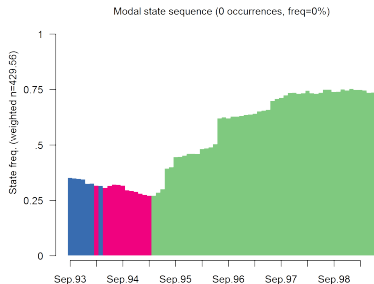


# Sequence of modal states

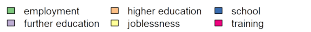
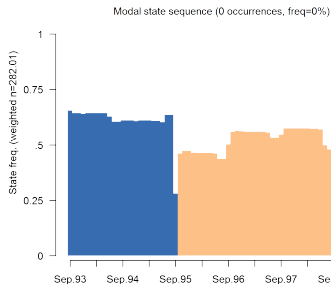
by qualification gained at end of compulsory school

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

End CS qualification - bad



End CS qualification - good

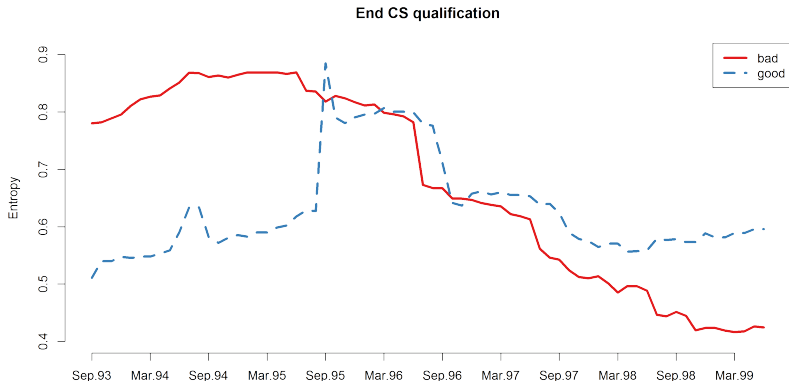




# Transversal entropies

## Time evolution of the transversal 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

`seqstatd()`

time in each state (longitudinal distribution)

`seqient()`

Longitudinal entropy

`seqST()`

Turbulence (Elzinga and Liefbroer, 2007)

`seqici()`

Complexity index (Gabadinho et al., 2011)

# Distinct successive states and their durations

- SPS format

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

- Distinct successive states(DSS)

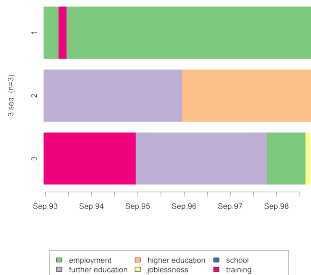
```
seqdss(mvad.seq)[1:3, ]
```

```
##      Sequence
## 1 EM-TR-EM
## 2 FE-HE
## 3 TR-FE-EM-JL
```

- Duration in successive states

```
seqdur(mvad.seq)[1:3, 1:5]
```

```
##      DUR1 DUR2 DUR3 DUR4 DUR5
## 1      4      2    64    NA    NA
## 2     36     34    NA    NA    NA
## 3     24     34    10     2    NA
```



# Complexity of the sequences

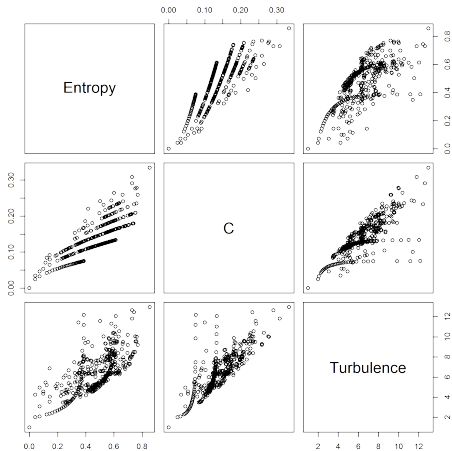
- To evaluate the complexity of a sequence we may consider
- **Longitudinal entropy**
  - does not account for the sequencing of the states  
(AABB and ABAB have same entropy)
- **Turbulence** (Elzinga and Liefbroer, 2007)
  - composite measure based on
    - the number of sub-sequences of the DSS sequence
    - 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

# Computing the sequence complexity measures

```
mvad.ient <- seqient(mvad.seq)
mvad.cplx <- seqici(mvad.seq)
mvad.turb <- seqST(mvad.seq)
ctab <- data.frame(mvad.ient, mvad.cplx, mvad.turb)
```

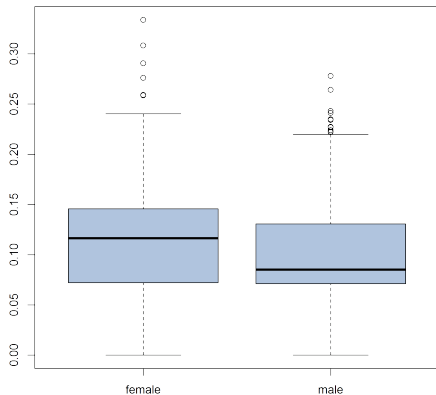
# Comparing the measures

```
plot(ctab)
```



# Distribution of complexity by sex

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



# Analyzing how complexity is related to covariates

## Regressing complexity on covariates

```
lm.ici <- lm(mvad.cplx ~ male + funemp + gcse5eq, data = mvad)
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.109	0.004	28.01	0.000
male	-0.013	0.004	-3.04	0.002
father unemployed	0.007	0.006	1.24	0.216
good ECS grade	0.010	0.005	2.20	0.028



# Pairwise dissimilarities between sequences

- Distance between sequences
  - Different metrics (LCP, LCS, OM, HAM, DHD)
- Once we have pairwise dissimilarities, we can
  - Partition a set of sequences into homogeneous **clusters**
  - Identify **representative sequences** (medoid, densest neighborhood)
  - Self-organizing maps (SOM) of sequences (Massoni et al., 2009)
  - **MDS** scatterplot representation of sequences
  - Measure the **discrepancy between sequences**
  - **Discrepancy analysis** of a set of sequences (ANOVA)
  - Grow **regression trees** for explaining the sequence discrepancy

# Summary of available distances

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

# 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]
```

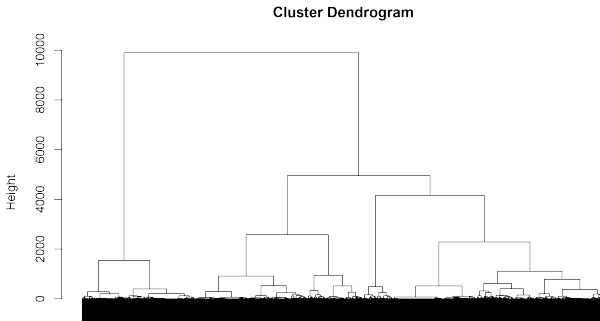
```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0   72   60   63   72   33
## [2,]   72    0   86  135   11  104
## [3,]   60   86    0   71   97   49
## [4,]   63  135   71    0  135   32
```

# 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 accept weights!):
    - `agnes()`: agglomerative nesting (average, UPGMA WPGMA, ward, beta-flexible, ...)
    - `diana()`: divisive partitioning
- For PAM and other direct partitioning methods
  - Packages: `cluster`, `fastclust`, `flashClust`, ...
  - `WeightedCluster` (Studer, 2013)

# Example: Hierarchical clustering (Ward)

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



```
as.dist(mvad.dist)  
hclust(*, "ward")
```

# 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)
```

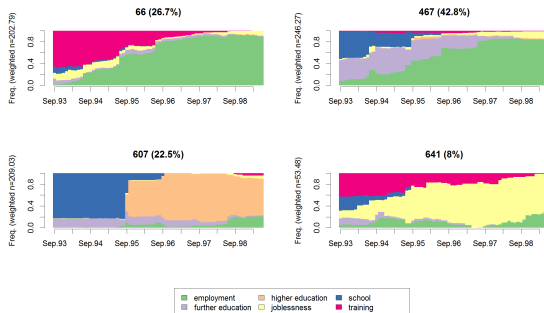
- Cluster membership is in `pam.mvad$clustering`

```
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)
```



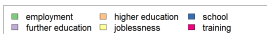
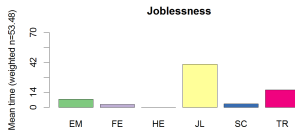
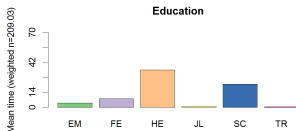
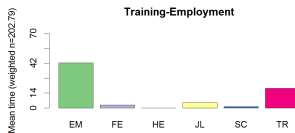
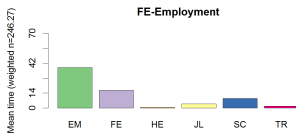
- Rearranging cluster order and defining labels

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



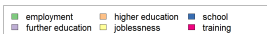
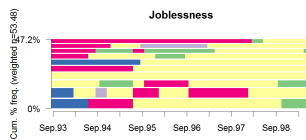
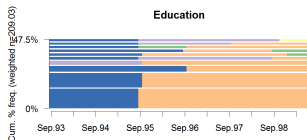
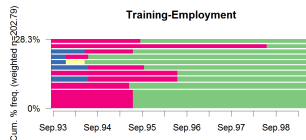
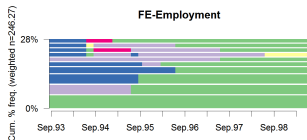
# Mean time in each state

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



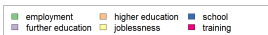
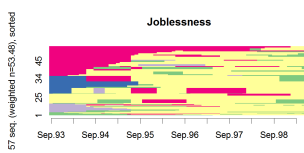
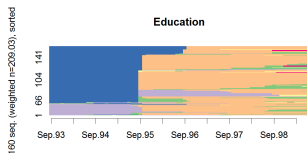
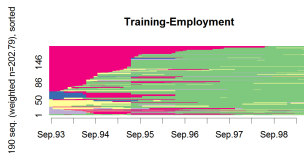
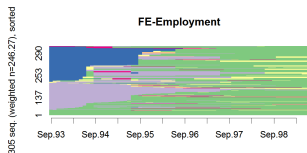
# Most frequent sequences

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



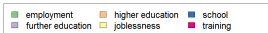
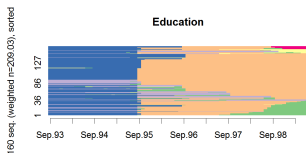
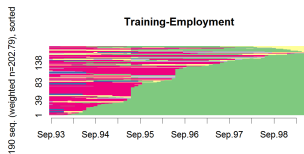
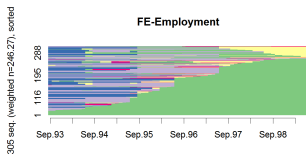
# Individual sequences (sorted by states from start)

```
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")
```



## 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
```



# Tree structured discrepancy analysis

- Objective: Find the most important predictors and their interactions.
- Iteratively segment the cases using values of covariates (predictors)
- Such that groups be as homogenous as possible.
- At each step, we select the covariate and split with highest  $R^2$ .
- Significance of split is assessed through a permutation  $F$  test.
- Growing stops when the selected split is not significant.

# Growing the tree

```
dt <- seqtree(mvad.seq ~ male + Grammar + funemp + gcse5eq + fmpr + livboth,
  weighted = FALSE, data = mvad, diss = mvad.dist, R = 5000)
print(dt, gap = 3)
```

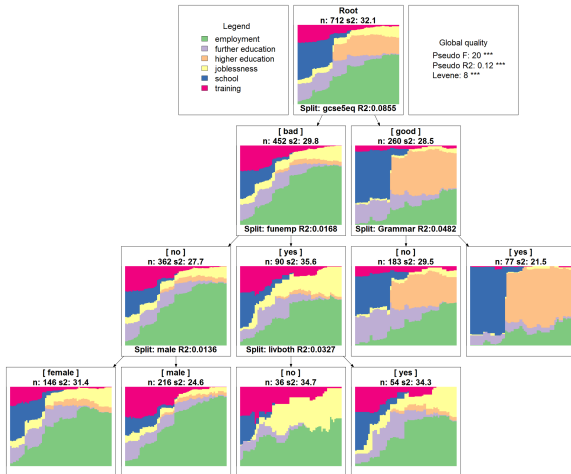
```
## Dissimilarity tree:
## Parameters: minSize=35.6, maxdepth=5, R=5000, pval=0.01
## Formula: mvad.seq ~ male + Grammar + funemp + gcse5eq + fmpr + livboth
## Global R2: 0.12
##
## Fitted tree:
##
## |-- Root (n: 712 disc: 32)
##   |--> gcse5eq 0.086
##     |-- [ bad ] (n: 452 disc: 30)
##       |--> funemp 0.017
##         |-- [ no ] (n: 362 disc: 28)
##           |--> male 0.014
##             |-- [ female ] (n: 146 disc: 31)[(FE,2)-(EM,68)] *
##             |-- [ male ] (n: 216 disc: 25)[(EM,70)] *
##             |-- [ yes ] (n: 90 disc: 36)
##               |--> livboth 0.033
##                 |-- [ no ] (n: 36 disc: 35)[(EM,70)] *
##                 |-- [ yes ] (n: 54 disc: 34)[(EM,70)] *
##             |-- [ good ] (n: 260 disc: 29)
##               |--> Grammar 0.048
##                 |-- [ no ] (n: 183 disc: 30)[(FE,22)-(EM,48)] *
##                 |-- [ yes ] (n: 77 disc: 21)[(SC,25)-(HE,45)] *
```

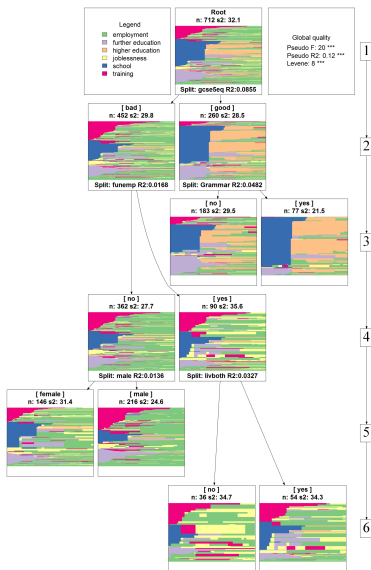
# Graphical tree

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

```
R> 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`

# 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

## Sequence analysis in R

[ home ] [ doc ] [ training ] [ preview ] [ who uses it ] [ history ] [ install ] [ help & contact ]

### 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

QuickSearch:   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.

### Documentation

- Mailing-list
- Online help
- User's Guide
- Citing TraMineR
- Trainings
- Publications
- Links

## 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.
- 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.
- 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.

## References II

- 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.
- Massoni, S., M. Olteanu, et P. Rousset (2009). Career-path analysis using optimal matching and self-organizing maps. In *Advances in Self-Organizing Maps : 7th International Workshop, WSOM 2009, St. Augustine, FL, USA, June 8-10, 2009*, Volume 5629 of *Lecture Notes in Computer Science*, pp. 154–162. Berlin : Springer.
- 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 III

- 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.
- 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.
- 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.



## References IV

- Studer, M., G. Ritschard, A. Gabadinho, et N. S. Müller (2011). Discrepancy analysis of state sequences. *Sociological Methods and Research* 40(3), 471–510.
- 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.
- Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning* 42(1/2), 31–60.