Sequence analysis for social scientists

Brendan Halpin, Dept of Sociology, University of Limerick

Oslo June 16-18 2015

Outline

- What is sequence analysis?
- Why it can be worth doing, and how it complements existing approaches
- How to do it, and how to think about it
- Practical, hands-on focus, using (inter alia) my SADI add-on for Stata (Halpin, 2014a)

Sequence Analysis

- What is sequence analysis?
  - Large, growing and ramifying research area
  - From Abbott and Hrycak (1990) to the 2015 edition of *Sociological Methodology*
  - See Halpin (2013) for an annotated bibliography
- Focus on lifecourse trajectories as *sequences*, as wholes
- Usually proceed by defining distances between pairs of sequences, classify, etc

Why do Sequence Analysis?

- Why would we want to do it
  - Holistic vs analytic?
  - Exploratory vs hypothesis testing?
  - Descriptive, visualisation
- Complexity of longitudinal processes hard to capture
- How should we think about $d \to D$?
- Complementary alternative to stochastic techniques which model data generation process

Sequences are messy

- Lifecourse sequences are epiphenomena of more fundamental underlying processes
- The processes are potentially complex: difficult to predict distribution of sequences
- Other techniques (hazard rate models, models of late outcome using history, models of the pattern of transition rates) give a powerful but partial view
- SA clearly allows us visualise complex data; possibly allows us observe features that will otherwise be missed

Potentially complex processes

- The generating processes are complex:
    - individuals bring different characteristics from the beginning
    - history matters, including via duration dependence (individuals accumulate characteristics)
    - time matters:
        - calendar time (e.g. economic cycle), state distribution may change dramatically
        - developmental time (maturation)
        - processes in other lifecourse domains
- Too many parameters to model, hard to visualise distribution of life courses, also the possibility of *emergent* features
    - Clear exploratory advantages
    - possibility of detecting things that might not be detected otherwise

Timing, sequence, quantum

- Different things can be interesting
  - Timing: when things happen
  - Sequence: in what order do things happen
  - Quantum: how much time is spent in different states (Billari et al., 2006)
- Many applications in longitudinal social science: annotated bibliography in Halpin (2013)

Non-holistic approaches

- Numerous non-holistic approaches exist
- Typically they will discard some aspect of the information in the data, and focus powerfully on another
- For instance, focus on
  - cumulated duration in states (how much but not when)
  - transition patterns between states (period-to-period but not overall)
  - time-to-event of leaving spell (spells, perhaps pooled, but lose sight of individual career).

Cumulative duration

- For instance, summarise trajectories in terms of cumulative time in each state
- Typically use as a predictor (e.g., proportion of time unemployed predicting later ill-health)
- Or as an outcome: variables measured earlier (e.g., school performance) predicting proportion of time unemployed.

Transition rate models

- Model rates of period-to-period change: e.g., monthly movement between labour market statuses
- Model origin–destination patterns: e.g., transition between class at entry to labour market, and class at age 35
- Markov models
- Very useful, good overview, can be descriptive or stochastic: tables make categorical data digestible
- Disadvantage: the focus on the t-1/t or t_0/t_T pattern means a loss of individual continuity
- Some potential to model longer Markov chains (Gabadinho, 2014)

Hazard-rate modelling

- Hazard-rate modelling is one of the dominant statistical alternative
- Either in terms of survival tables and curves (essentially descriptive)
- Or full stochastic models of the determinants of the hazard rate (Cox and/or parametric)
- Example: what characteristics speed up (or slow down) exit from unemployment?
- Very nice conceptual model of the temporal process
- Can test hypotheses
- Disadvantage: spell orientation, lack of whole-trajectory overview

Latent class analysis

- Latent class growth curve models
  - Where theory allows a developmental model of a quantitative outcome
  - Account for the structure of repeated measurement of individuals
  - Not so suitable for categorical variables
- Latent class models can be applied to careers
  - However, difficult to properly incorporate the longitudinality
  - Examples: Lovaglio and Mezzanzanica (2013); Barban and Billari (2012)

Holistic approaches

- Holistic approaches by definition treat whole trajectories as units
- Classification of sequences is a typical goal
- Usually achieved by defining inter-sequence similarity and cluster analysis
- But other aspects of similarity may be interesting
    - Variation of similarity by grouping variable (cohort, social class)
    - Dyad similarity (couples' time use, mother–daughter fertility etc)
    - Distance to pre-defined ideal types (empirical or theoretical)

Defining similarity

- Defining similarity the key challenge: must be
  - efficient
  - coherent, and
  - sociologically meaningful
- We will consider a number of methods to do this
  - Hamming distance and Optimal Matching distance (today)
  - Dynamic Hamming, time-warping measures and combinatorial subsequence measures (later)

Hamming distance and Optimal Matching

- The simplest way to compare sequences is element-wise
- Given a rule for $d(a, b)$, project it onto $D(A, B)$ as $D(A, B) = \sum_i d(A_i, B_i)$
- Requires sequence of equal length
- Hamming distance: recognises match or similarity at same time
- Simple but important case of mapping $d(a, b) \rightarrow D(A, B)$

Hamming distance example

Input four short sequences

```
input s1 s2 s3 s4 s5
1 2 3 2 3
2 3 2 3 1
4 2 3 2 3
1 1 1 1 1
end

// Define the state differences
matrix scost = (0,1,2,3 \ ///
                1,0,1,2 \ ///
                2,1,0,1 \ ///
                3,2,1,0 )

hamming s1-s5, subs(scost) pwd(ham)
```

Hamming distance example

Input four short sequences

```
input s1 s2 s3 s4 s5
1 2 3 2 3
2 3 2 3 1
4 2 3 2 3
1 1 1 1 1
end

// Define the state differences
matrix scost = (0,1,2,3 \ ///
                1,0,1,2 \ ///
                2,1,0,1 \ ///
                3,2,1,0 )

hamming s1-s5, subs(scost) pwd(ham)
```

Resulting distances

```
. matrix list ham

symmetric ham[4,4]
     c1   c2   c3   c4
r1    0
r2  1.2    0
r3   .6  1.4    0
r4  1.2  1.2  1.8    0
```

Optimal Matching

- Hamming recognises similarity at the same time
- If sequences have similarity that is out of alignment this will not be recognised
- OM defines similarity like Hamming, but with insertion and deletion to allow sequences to align
- I.e., it cuts bits out in order to slide other parts along to match
- Insertion/deletion also enables comparison of sequences of different lengths

OM example

OMA call

```
. oma s1-s5, subs(scost) indel(1.5) ///
             pwd(oma) length(5)
```

OM example

OMA call

```
. oma s1-s5, subs(scost) indel(1.5) ///
             pwd(oma) length(5)
```

Resulting distances

- OM distances

```
symmetric oma[4,4]
      c1    c2    c3    c4
r1     0
r2    .6     0
r3    .6    .6     0
r4   1.2   1.2   1.8     0
```

- Hamming distances

```
symmetric ham[4,4]
      c1    c2    c3    c4
r1     0
r2   1.2     0
r3    .6   1.4     0
r4   1.2   1.2   1.8     0
```

OM vs Hamming

- For most pairs the OM and Hamming distance is the same
- For the pairs (1,2) and (2,3), OM distance is less because "alignment" allows a better match
- 1 vs 2

Seq 1	1	2	3	2	3	-
Seq 2	-	2	3	2	3	1
Cost	i	0	0	0	0	i

- 2 vs 3

Seq 2	-	2	3	2	3	1
Seq 3	4	2	3	2	3	-
Cost	i	0	0	0	0	i

A more general example

To convert ABCD into CDAAB the following set of operations gives the cheapest path:

Operation	Intermediate state	Cost
Sequence 2	ABCD	= 0
		=
		=
		=
		=
		=
		=
Sequence 1	CDAAB	=

A more general example

To convert `ABCD` into `CDAAB` the following set of operations gives the cheapest path:

Operation	Intermediate state	Cost
Sequence 2	`ABCD`	$= 0$
insert `C`	`CABCD`	$+2 = 2$
		$=$
		$=$
		$=$
		$=$
		$=$
Sequence 1	`CDAAB`	$=$

A more general example

To convert `ABCD` into `CDAAB` the following set of operations gives the cheapest path:

Operation	Intermediate state	Cost
Sequence 2	`ABCD`	$= 0$
insert `C`	`CABCD`	$+2 = 2$
insert `D`	`CDABCD`	$+2 = 4$
		$=$
		$=$
		$=$
		$=$
Sequence 1	`CDAAB`	$=$

A more general example

To convert ABCD into CDAAB the following set of operations gives the cheapest path:

Operation	Intermediate state	Cost
Sequence 2	ABCD	$= 0$
insert C	CABCD	$+2 = 2$
insert D	CDABCD	$+2 = 4$
const A = A	CDABCD	$+0 = 4$
		$=$
		$=$
		$=$
Sequence 1	CDAAB	$=$

A more general example

To convert ABCD into CDAAB the following set of operations gives the cheapest path:

Operation	Intermediate state	Cost
Sequence 2	ABCD	= 0
insert C	CABCD	+2 = 2
insert D	CDABCD	+2 = 4
const A = A	CDABCD	+0 = 4
subs B→A	CDAACD	+1 = 5
		=
		=
Sequence 1	CDAAB	=

A more general example

To convert ABCD into CDAAB the following set of operations gives the cheapest path:

Operation	Intermediate state	Cost
Sequence 2	ABCD	$= 0$
insert C	CABCD	$+2 = 2$
insert D	CDABCD	$+2 = 4$
const A = A	CDABCD	$+0 = 4$
subs B→A	CDAACD	$+1 = 5$
subs C→B	CDAABD	$+1 = 6$
		$=$
Sequence 1	CDAAB	$=$

A more general example

To convert ABCD into CDAAB the following set of operations gives the cheapest path:

Operation	Intermediate state	Cost
Sequence 2	ABCD	= 0
insert C	CABCD	+2 = 2
insert D	CDABCD	+2 = 4
const A = A	CDABCD	+0 = 4
subs B→A	CDAACD	+1 = 5
subs C→B	CDAABD	+1 = 6
delete D	CDAAB-	+2 = 8
Sequence 1	CDAAB	= 8

Programming OM

- OM distance is defined as the cheapest set of "elementary operations" that edit one sequence into another
- Determining the cheapest set of “elementary operations” is potentially complex – a large population of candidates
- However, it can be stated as a recursive problem and programmed very efficiently
- Understanding how it is programmed can help understand the principle of OM

OM: Recursive problem

$$\Delta_{OM}(A^p, B^q) =$$

$$min \begin{cases} \Delta_{OM}(A^{p-1}, B^q) & + indel \\ \Delta_{OM}(A^{p-1}, B^{q-1}) & + \delta(a_p, b_q) \\ \Delta_{OM}(A^p, B^{q-1}) & + indel \end{cases}$$

(Δ represents distance between sequences, and δ differences within the state space)

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$

s_2

		A	B	C	D
	C	2	1	0	1
s_1	D	3	2	1	0
	A	0	1	2	3
	A	0	1	2	3
	B	1	0	1	2

0	2	4	6	8
2				
4				
6				
8				
10				

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$
$= min(0+2, 2+2, 2+2) = 2$

s_2

		A	B	C	D
	C	2	1	0	1
s_1	D	3	2	1	0
	A	0	1	2	3
	A	0	1	2	3
	B	1	0	1	2

0	2	4	6	8
2				
4				
6				
8				
10				

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$

s_2

		A	B	C	D
	C	2	1	0	1
s_1	D	3	2	1	0
	A	0	1	2	3
	A	0	1	2	3
	B	1	0	1	2

0	2	4	6	8
2	2			
4				
6				
8				
10				

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$

$= min(2 + 1, 2 + 2, 4 + 2) = 3$

$$\begin{array}{cc|c|c|c|c|}
 & & \multicolumn{4}{c}{s_2} \\
 & & A & B & C & D \\
\hline
 & C & 2 & 1 & 0 & 1 \\
\hline
s_1 & D & 3 & 2 & 1 & 0 \\
\hline
 & A & 0 & 1 & 2 & 3 \\
\hline
 & A & 0 & 1 & 2 & 3 \\
\hline
 & B & 1 & 0 & 1 & 2 \\
\hline
\end{array}$$

$$\begin{array}{c|c|c|c|c|}
0 & 2 & 4 & 6 & 8 \\
\hline
2 & 2 & & & \\
\hline
4 & & & & \\
\hline
6 & & & & \\
\hline
8 & & & & \\
\hline
10 & & & & \\
\hline
\end{array}$$

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$

		s_2			
		A	B	C	D
	C	2	1	0	1
s_1	D	3	2	1	0
	A	0	1	2	3
	A	0	1	2	3
	B	1	0	1	2

0	2	4	6	8
2	2	3		
4				
6				
8				
10				

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$

$$= min(4 + 0, 3 + 2, 6 + 2) = 4$$

s_2

		A	B	C	D
	C	2	1	0	1
s_1	D	3	2	1	0
	A	0	1	2	3
	A	0	1	2	3
	B	1	0	1	2

0	2	4	6	8
2	2	3		
4				
6				
8				
10				

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$

			s_2		
		A	B	C	D
	C	2	1	0	1
s_1	D	3	2	1	0
	A	0	1	2	3
	A	0	1	2	3
	B	1	0	1	2

0	2	4	6	8
2	2	3	4	
4				
6				
8				
10				

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$

$$= min(6+1, 4+2, 8+2) = 6$$

s_2

		A	B	C	D
	C	2	1	0	1
s_1	D	3	2	1	0
	A	0	1	2	3
	A	0	1	2	3
	B	1	0	1	2

0	2	4	6	8
2	2	3	4	
4				
6				
8				
10				

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$

			s_2		
		A	B	C	D
	C	2	1	0	1
s_1	D	3	2	1	0
	A	0	1	2	3
	A	0	1	2	3
	B	1	0	1	2

0	2	4	6	8
2	2	3	4	6
4				
6				
8				
10				

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$

		s_2			
		A	B	C	D
	C	2	1	0	1
s_1	D	3	2	1	0
	A	0	1	2	3
	A	0	1	2	3
	B	1	0	1	2

0	2	4	6	8
2	2	3	4	6
4	4	4	4	4
6				
8				
10				

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$

		s_2			
		A	B	C	D
	C	2	1	0	1
s_1	D	3	2	1	0
	A	0	1	2	3
	A	0	1	2	3
	B	1	0	1	2

0	2	4	6	8
2	2	3	4	6
4	4	4	4	4
6	4	5	6	6
8				
10				

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$

s_2

		A	B	C	D
	C	2	1	0	1
s_1	D	3	2	1	0
	A	0	1	2	3
	A	0	1	2	3
	B	1	0	1	2

0	2	4	6	8
2	2	3	4	6
4	4	4	4	4
6	4	5	6	6
8	6	5	7	8
10				

Implementing the recursive algorithm

Cell value: $min(c_{i-1,j-1} + \omega_{i,j},\ c_{i,j-1} + \iota,\ c_{i-1,j} + \iota)$

s_2

		A	B	C	D
	C	2	1	0	1
s_1	D	3	2	1	0
	A	0	1	2	3
	A	0	1	2	3
	B	1	0	1	2

0	2	4	6	8
2	2	3	4	6
4	4	4	4	4
6	4	5	6	6
8	6	5	7	8
10	8	6	6	8

Two example data sets

- We will be primarily using two data sets as examples
  - MVAD: McVicar/Anyadike-Danes data on the school-to-work transition in Northern Ireland (72 months, 6 states)
  - BSSEQ: 6 years of labour market history of women who have a birth at end of year 2 (72 months, 4 states)

Initial step: looking at life course data

- It's harder to get an overview of lifecourse that cross-sectional data
- However, a number of numeric and graphical techniques are available

Numeric summaries

We can summarise lifecourse data in terms of:

- Cumulative duration
- Number of spells
- Patterns of transition rates
    - month by month
    - start by finish
- Durations to event (time to first job, first marriage, first child)

Useful to break down these measures by covariates, and model them

Cumulative duration

```
use mvad
cumuldur state*, cd(cd) nstates(6)
reshape long cd, i(id) j(durtype)
label values durtype state
table male durtype, c(mean cd) format(%5.2f)
table grammar durtype, c(mean cd) format(%5.2f)
```

```
----------------------------------------------------
          |                 durtype
     male |     E      F      H      S      T      U
----------+-----------------------------------------
        0 | 29.24  12.73  10.12   7.30   5.55   7.06
        1 | 34.96  10.75   6.81   5.00   9.12   5.36
----------------------------------------------------

----------------------------------------------------
          |                 durtype
  grammar |     E      F      H      S      T      U
----------+-----------------------------------------
        0 | 34.25  12.42   6.07   4.44   8.09   6.74
        1 | 23.02   8.47  18.93  13.62   4.32   3.64
----------------------------------------------------
```

Number of spells

```
. nspells state*, gen(nsp)
. tab nsp grammar, col nofreq

           |        grammar
       nsp |         0          1 |     Total
-----------+----------------------+----------
         1 |      6.17       4.65 |      5.90
         2 |     20.24      24.81 |     21.07
         3 |     30.70      33.33 |     31.18
         4 |     19.21      19.38 |     19.24
         5 |     12.52       6.98 |     11.52
         6 |      4.12       6.20 |      4.49
         7 |      3.95       1.55 |      3.51
         8 |      1.37       2.33 |      1.54
         9 |      1.03       0.78 |      0.98
        10 |      0.34       0.00 |      0.28
        11 |      0.34       0.00 |      0.28
-----------+----------------------+----------
     Total |    100.00     100.00 |    100.00
```

Transition rates

```
use mvad

reshape long state, i(id) j(t)

by id: gen last = state[_n-1] if _n>1

label values last state

tab last state, row nofreq
```

Transition rates

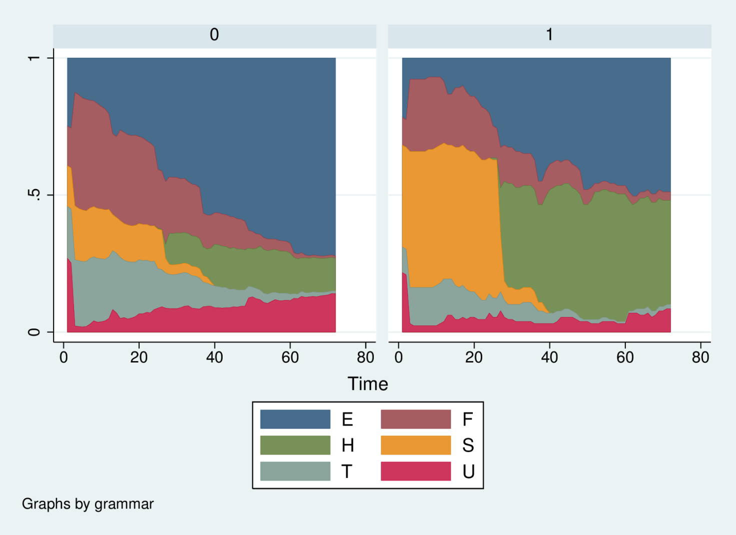
```
           |                                state
      last |         E          F          H          S          T          U |     Total
-----------+------------------------------------------------------------------+----------
         E |    22,039        115         56         39         58        146 |    22,453 
           |     98.16       0.51       0.25       0.17       0.26       0.65 |    100.00 
-----------+------------------------------------------------------------------+----------
         F |       227      7,927         54          8         33         73 |     8,322 
           |      2.73      95.25       0.65       0.10       0.40       0.88 |    100.00 
-----------+------------------------------------------------------------------+----------
         H |        60          1      5,787          0          3         11 |     5,862 
           |      1.02       0.02      98.72       0.00       0.05       0.19 |    100.00 
-----------+------------------------------------------------------------------+----------
         S |        59         50         74      4,120         19         23 |     4,345 
           |      1.36       1.15       1.70      94.82       0.44       0.53 |    100.00 
-----------+------------------------------------------------------------------+----------
         T |       197         21          0          4      4,973         69 |     5,264 
           |      3.74       0.40       0.00       0.08      94.47       1.31 |    100.00 
-----------+------------------------------------------------------------------+----------
         U |       182        120          9         39         64      3,892 |     4,306 
           |      4.23       2.79       0.21       0.91       1.49      90.39 |    100.00 
-----------+------------------------------------------------------------------+----------
     Total |    22,764      8,234      5,980      4,210      5,150      4,214 |    50,552 
           |     45.03      16.29      11.83       8.33      10.19       8.34 |    100.00 
```

Graphs

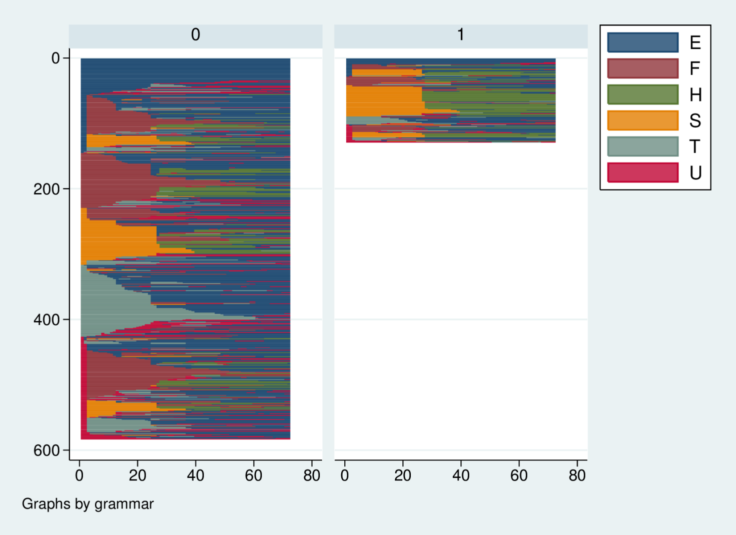
Graphs give us an even better overview. Consider

- Chronograms
- Survival plots
- Index plots
- Transition rate time-series

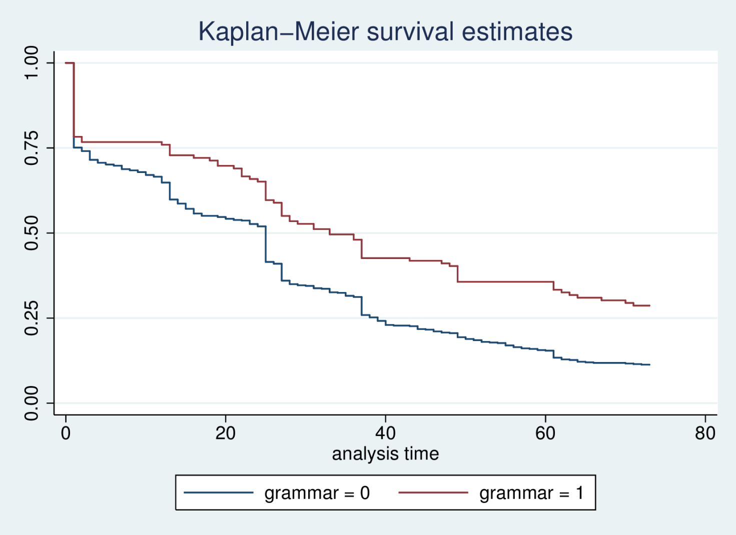
Chronograms



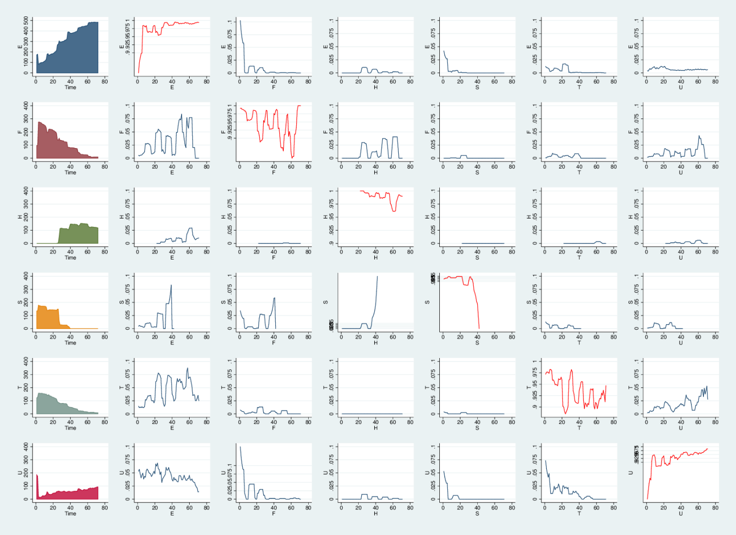
Index plots


0
1
0
200
400
600
0
20
40
60
80
0
20
40
60
80
E
F
H
S
T
U
Graphs by grammar

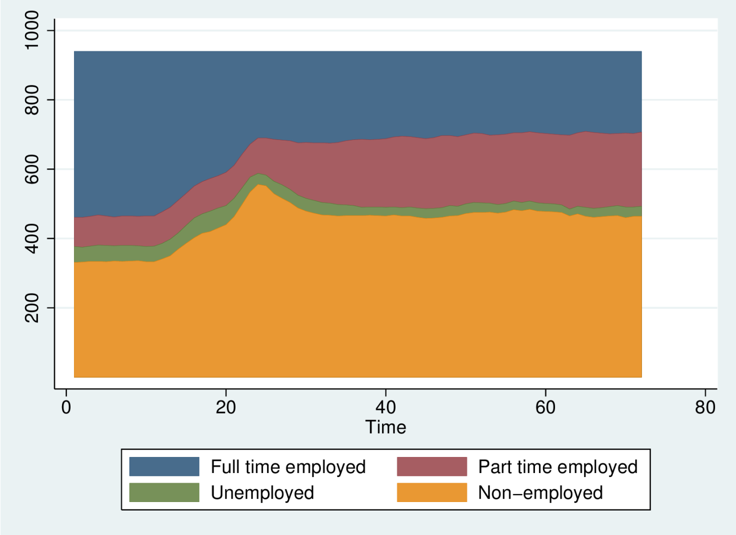
Survival plots: time to first job



Transition rate time-series



Chronogram, mothers' labour market history (BS)



OM on BS data

```
use bsseq
matrix scost = (0,1,2,3 \ ///
                1,0,1,2 \ ///
                2,1,0,1 \ ///
                3,2,1,0 )
oma state*, subs(scost) indel(1.5) pwd(oma) len(72)
matlist oma[1..5,1..5]
```

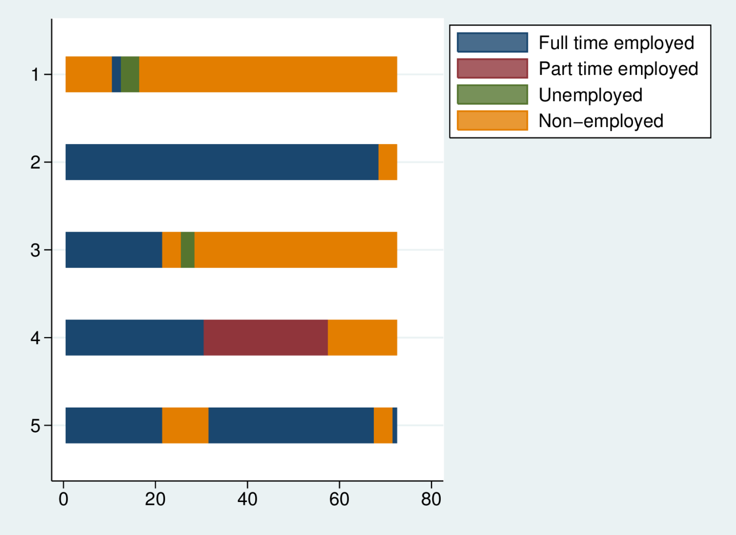
OM output

```
. oma state*, subs(scost) indel(1.5) pwd(oma) len(72)
Normalising distances with respect to length
(0 observations deleted)
417 unique observations
nrefs:     0

. matlist oma[1..5,1..5]

             |        c1         c2         c3         c4         c5
-------------+-------------------------------------------------------
          r1 |         0
          r2 |  2.694444          0
          r3 |  .7777778   1.916667          0
          r4 |  1.861111   .8333333   1.083333          0
          r5 |  2.277778   .4583333   1.541667   .8333333          0
```

Hamming for comparison

```
. hamming state*, subs(scost) pwd(ham)

. corrsqm ham oma
VECH correlation between ham and oma: 0.9946


. matlist ham[1..5,1..5]

             |        c1         c2         c3         c4         c5
-------------+------------------------------------------------------
          r1 |         0
          r2 |  2.694444          0
          r3 |  .7777778   1.916667          0
          r4 |  1.861111   .8333333   1.083333          0
          r5 |  2.277778         .5   1.583333   1.222222          0
```

First five sequences



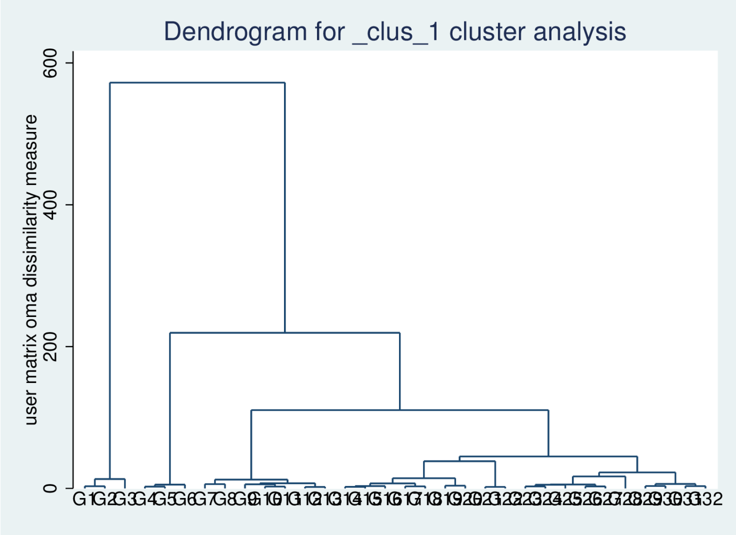
What to do with distances?

- Pairwise distance matrices are an intermediate point
- One useful thing: create a data-driven classification
- Use cluster analysis, typically using Ward's linkage
- Number of clusters is a matter for thought, 8 is convenient for exposition

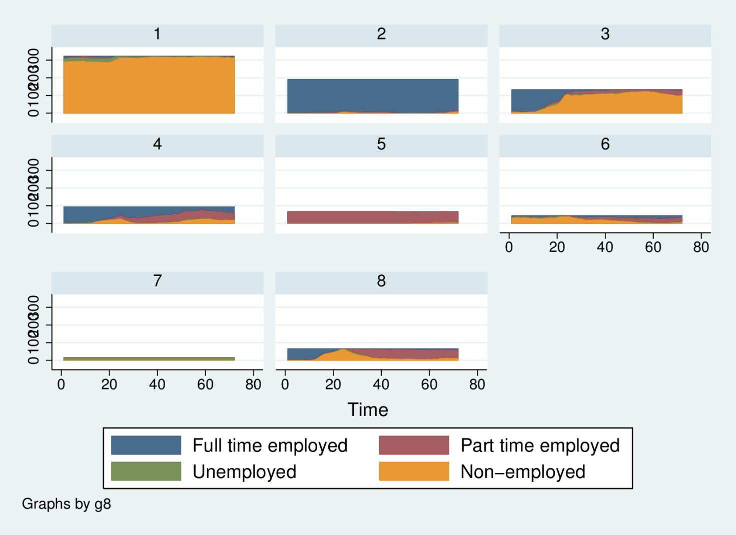
Clustering OM

```
clustermat wards oma, add
cluster generate g8=groups(8)
cluster dendrogram, cutnumber(32)
chronogram state*, by(g8)
```

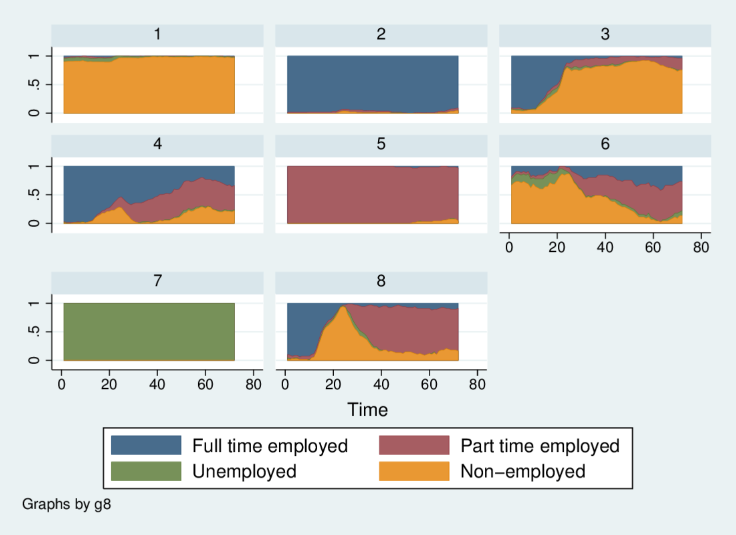
Dendrogram


Dendrogram for _clus_1 cluster analysis
user matrix oma dissimilarity measure
0
200
400
600
G1G2G3G4G5G6G7G8G9G10G11G12G13G14G15G16G17G18G19G20G21G22G23G24G25G26G27G28G29G30G31G32

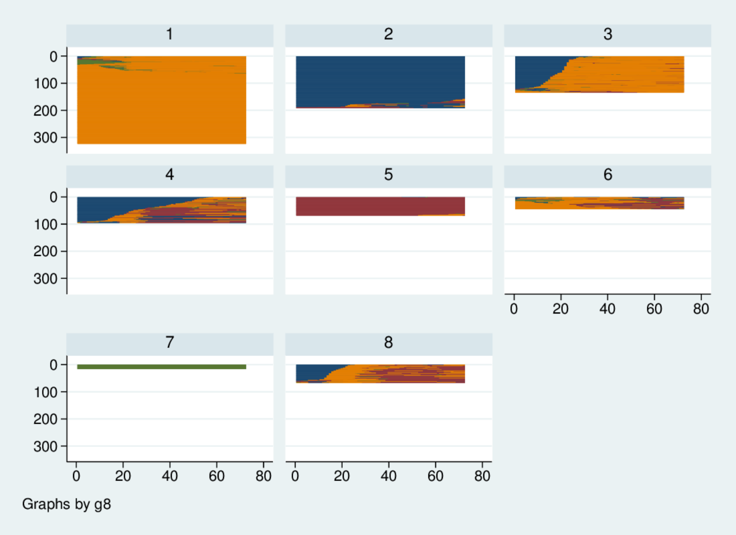
Chronogram by cluster



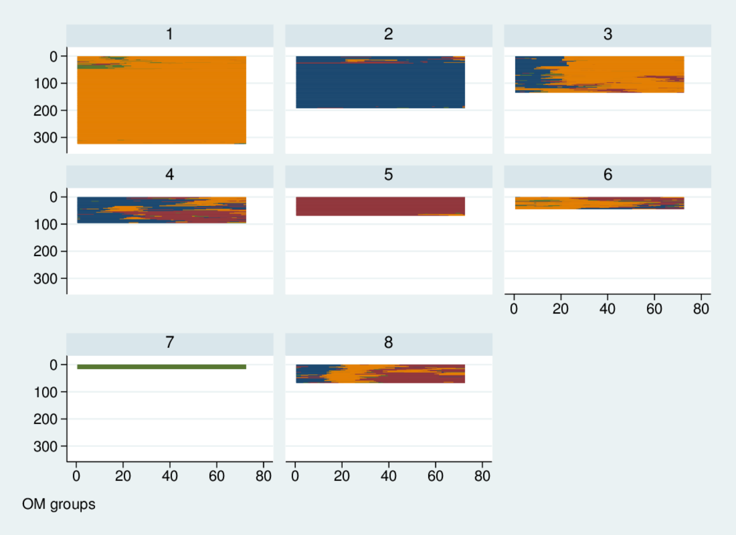
Chronogram, proportional



Indexplot



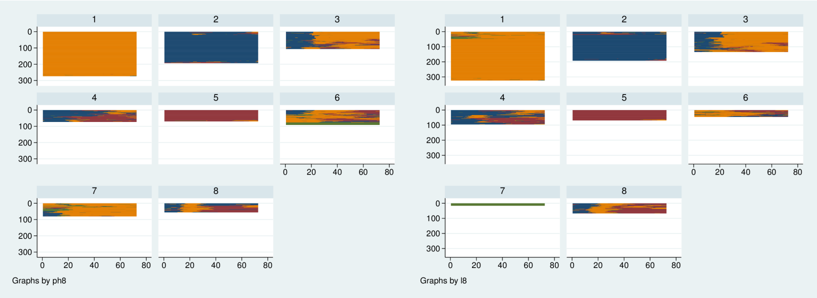
Indexplot in dendrogram order



Details

```
clustermat wards oma, add
cluster generate g8 = groups(8)
cluster generate g999 = groups(800), ties(fewer)

chronogram state*, by(g8)
chronogram state*, by(g8) prop

reshape long state, i(pid) j(t)
sqset state pid t
sqindexplot, by(g8, legend(off))
sqindexplot, by(g8, legend(off)) order(g999)
```

Compare Hamming (L) and OM (R) solutions



ARI and permtab

	Hamming							
OM	1	2	3	4	5	6	7	8
1	273	0	1	0	0	1	48	0
2	0	192	0	0	0	0	0	0
3	0	0	85	0	1	16	32	0
4	0	0	10	69	0	0	0	16
5	0	0	0	0	68	0	0	0
6	0	1	0	0	0	44	0	0
7	0	0	0	0	0	16	0	0
8	0	0	10	4	0	14	0	39

- Kappa-max: 0.7791
- Adjusted Rand Index: 0.7818

Complexity of sequences

- Complexity of sequences is relevant: more complex means less likely to be similar (and perhaps, similarity is more interesting)
- How to measure? Number of spells is part of it
- Also distribution of time
- A single long spell is the simplest sequence
- Many spells in many different states is very complex

Shannon Entropy

- Information theory relates complexity to "entropy"
- More complex objects are harder to describe, cannot be compressed
- Shannon Entropy: $\epsilon = -\sum p_i \log_2 p_i$ where p_i is the proportion of months in state i
- Takes account of diversity of state but `ABABAB` counts as no more complex than `AAABBB`
- Perhaps add n-spells information: $\epsilon' = \epsilon \times \frac{m}{l}$ where m is number of spells and l is length

Example: entropy

```
entropy state*, gen(ent) cd(pcd) nstates(4)
nspells state*, gen(nsp)
gen ent2 = ent*nsp/72
table g8, c(mean ent mean ent2 mean nsp) format(%6.3f)
```

```
---------------------------------------------
       g8 |  mean(ent)  mean(ent2)   mean(nsp)
----------+----------------------------------
        1 |      0.150       0.008       1.536
        2 |      0.100       0.004       1.359
        3 |      1.143       0.061       3.560
        4 |      1.053       0.057       3.684
        5 |      0.074       0.003       1.235
        6 |      1.252       0.091       4.844
        7 |      0.000       0.000       1.000
        8 |      1.489       0.097       4.597
---------------------------------------------
```

Elzinga's turbulence

- In Elzinga (2010) a measure of complexity is proposed that is more appropriate for spell data
- It is based on duration weighted spells, and on subsequence counting
- It combines a measure based on the number of distince subsequences, with a measure of the variance of their durations
- It is (only) available in TraMineR
- However, in practice the simpler Shannon entropy correlates highly with it

Regular expressions

- If sequences are represented as text, text-processing tools such as "regular expressions" can be used to sort between them
- Refer to lab notes for more details

```
stripe state*, gen(seqst)
list seqst in 1/5,clean
count if regexm(seqst,"^A+$")
count if regexm(seqst,"^AAAAAA+.*DDDDDD.*AAAAAA.*$")
count if regexm(seqst,"AB.*AB")
```

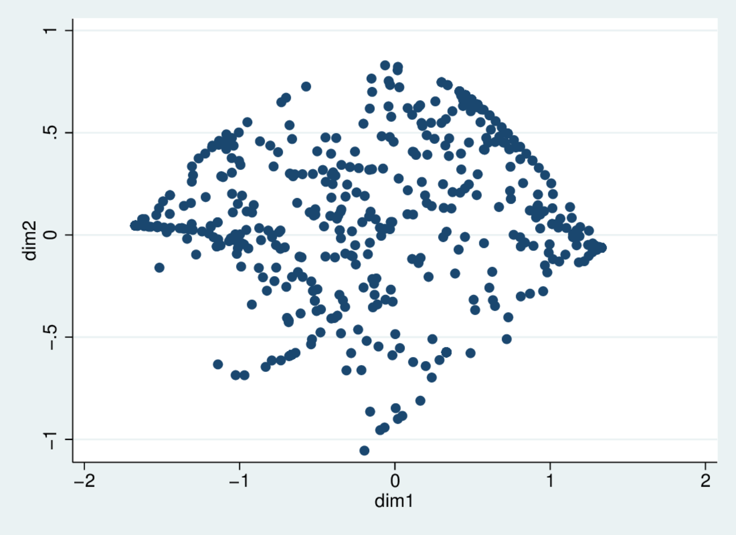
Multi-dimensional scaling

- The other "obvious" thing to do with pairwise distances is multi-dimensional scaling
- The network of distances implies a coherent space: can we re-construct it?
- Preferably with dimensions much less than number of sequences!
- Standard MDS uses principal component analysis

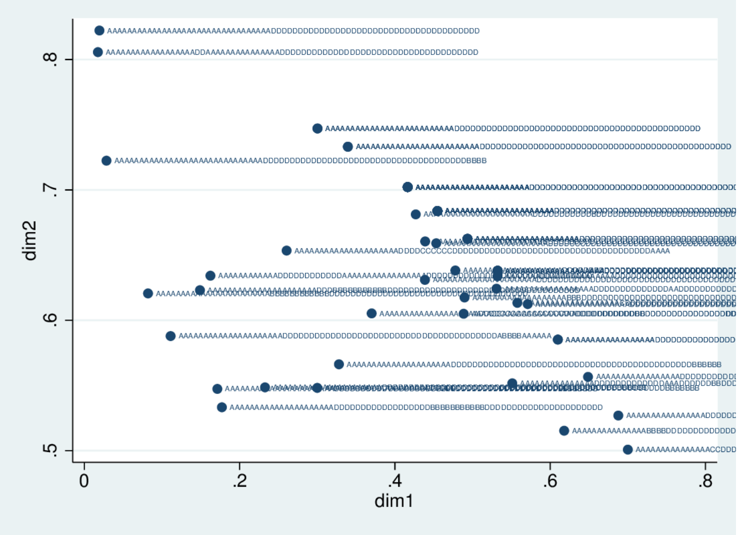
Example

```
. mdsmat oma, dim(3)
(row names of (dis)similarity matrix differ from column names; row names used)

Classical metric multidimensional scaling
    dissimilarity matrix: oma

                                            Number of obs          =       940
    Eigenvalues > 0       =        188      Mardia fit measure 1 =    0.7556
    Retained dimensions   =          3      Mardia fit measure 2 =    0.9932

    ---------------------------------------------------------------------------
                 |                     abs(eigenvalue)          (eigenvalue)^2
      Dimension  |  Eigenvalue      Percent    Cumul.        Percent    Cumul.
    -------------+-------------------------------------------------------------
              1  |   1205.3971        67.73     67.73          98.57     98.57
              2  |   95.282325         5.35     73.08           0.62     99.19
              3  |   44.082404         2.48     75.56           0.13     99.32
    -------------+-------------------------------------------------------------
              4  |   28.932307         1.63     77.19           0.06     99.38
              5  |   23.350698         1.31     78.50           0.04     99.41
              6  |   12.040492         0.68     79.17           0.01     99.42
              7  |   10.398137         0.58     79.76           0.01     99.43
              8  |   8.8446418         0.50     80.26           0.01     99.44
              9  |   6.3672493         0.36     80.61           0.00     99.44
             10  |   6.1013343         0.34     80.96           0.00     99.44
    ---------------------------------------------------------------------------
```

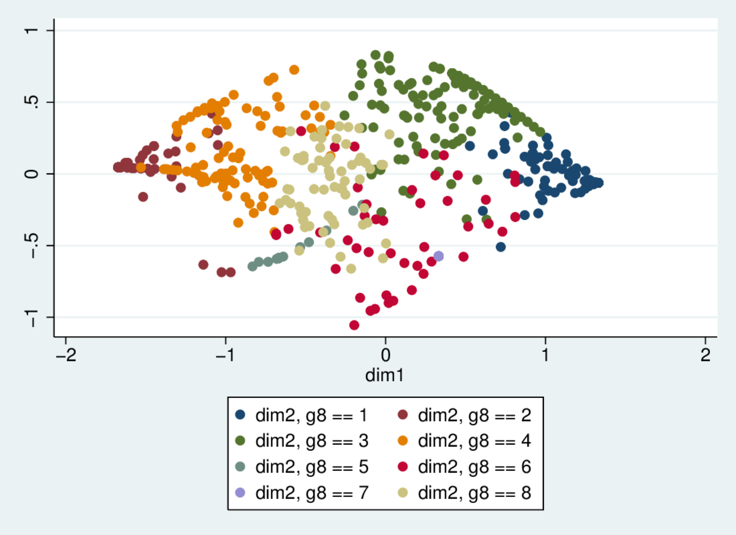
Scatterplot



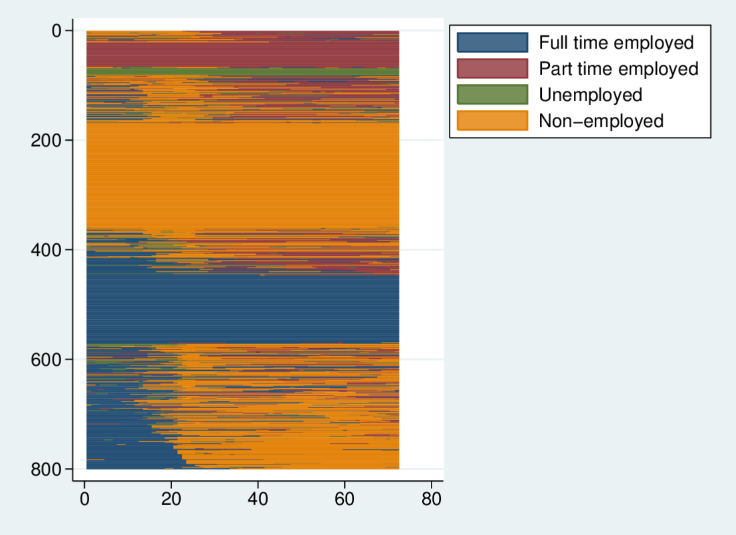
Scatterplot



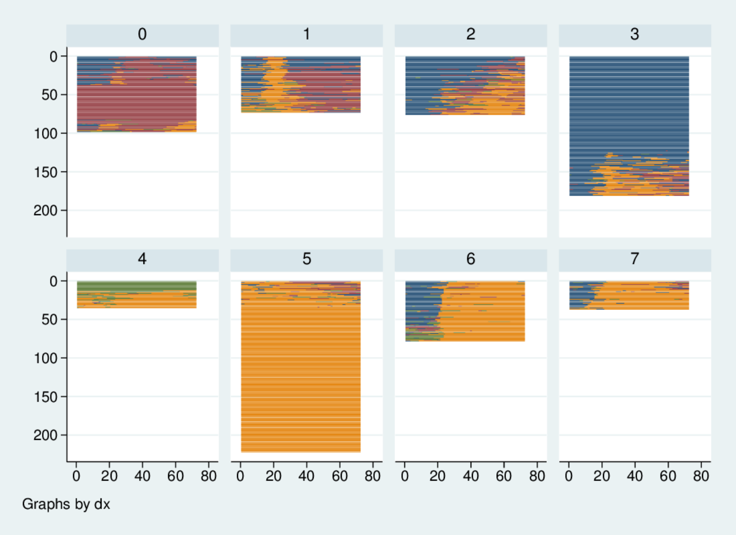
Scatterplot by cluster solution



Avoid clustering: Indexplot ordered by 1st MDS dimension



Partitioning by MDS



Are substitution costs a problem?

- Repeated claims in the literature:
  - that sociologists don't know how to set substitution costs,
  - that we can't match the effectiveness of molecular biology
- Yes, our analytical goals are often much less well defined than those of the biologists
- No, substitution costs are not an intractable problem

Mapping states to sequences

- The essence of SA is mapping a view of a state space onto a view of a trajectory space: $d(s) \rightarrow D(S)$
- We start with *knowledge* or a *view* of how states relate to each other (what states are like each other, what states are dissimilar)
- With a suitable algorithm we map this perspective onto trajectories through the state space: what trajectories are more or less similar
- The nature of the algorithm determines
  - Whether the mapping makes sense
  - Exactly how the structure of the state space affects the structure of the trajectory space

OMA coherent?

- Can we expect OMA to provide a coherent $d(s) \rightarrow D(S)$ mapping?
- Elementary operations are intuitively appealing:
  1. $D(\mathtt{ABC}, \mathtt{ADC}) = f(d(\mathtt{B}, \mathtt{D}))$
  2. $D(\mathtt{ABCD}, \mathtt{ABD}) = f(\mathit{indel})$
  3. minimising concatenation of these two operations to link any pair of trajectories
- If 3 is reasonable, 1 and 2 determine how state space affects trajectory space

Thinking about state spaces and distances

- Costs can be thought of as distances between states
- If state space is $\mathbb{R}^n$, distance is intuitive
- If state space is categorical, how define distance?
  - State space as efficient summary of clustered distribution in $\mathbb{R}^n$: distances are between cluster centroids
  - State space can be mapped onto specific set of quantitative dimensions; each state located at the vector of its mean values; Euclidean or other distances between vectors
  - States can be located relative to each other on theoretical grounds

Transitions and substitutions

- Transition rates frequently proposed as basis for substitution costs
- Critics of OMA complain of substitution operations implying impossible transitions (e.g., Wu)
- Even proponents of OMA are sometimes concerned about "impossible" transitions (e.g., Pollock)
- But substitutions are not transitions, {not even a little bit!}
  - substitutions happen across sequences, $D(\mathtt{ABC}, \mathtt{ADC}) = f(d(\mathtt{B}, \mathtt{D}))$ (similarity of states)
  - transitions happen within sequences (movement between state)

Informative transition rates

- No logical connection between substitutions and transition rates
- but under certain circumstances transition rates can inform us about state distances
- If state space is a partitioning of an unknown $\mathbb{R}^n$, movement is random (unstructured), and the probability of a move is inversely related to its length, then
- distance between states will vary inversely with the transition rates
- However, these conditions usually not met

Deceptive transiton rates

- Example: using voting intentions as a way of defining inter party distances
- UK: relatively high Con–LibDem two-way flows; ditto Lab–LibDem
- But Con–Lab transitions much lower: implies a potentially incoherent space (non-metric, more below)
    - $d(\mathrm{Con}, \mathrm{Lab}) > d(\mathrm{Con}, \mathrm{LibDem}) + d(\mathrm{LibDem}, \mathrm{Lab})$
- Procedure confuses party state space and voter characteristics
- Voter polarisation/loyalty is trajectory information, not state information
- Another type of problem: irrelevant distinctions can cause similar states to have low transition rates

Take “space” seriously

- Very useful to think in spatial terms
  1. State space as efficient summary of clustered distribution in $\mathbb{R}^n$
  2. State space mapped onto specific set of quantitative dimensions
  3. State space defined on theoretical grounds
- For 1 and 2, explicitly multidimensional, in case 2 dimensions are explicit
- For 1 and 3, we can attempt to recover the implicit dimensions

Looking at state spaces

- Two very simple state spaces:
    - Single dimension, equally spaced:

0	1	2	3
1	0	1	2
2	1	0	1
3	2	1	0

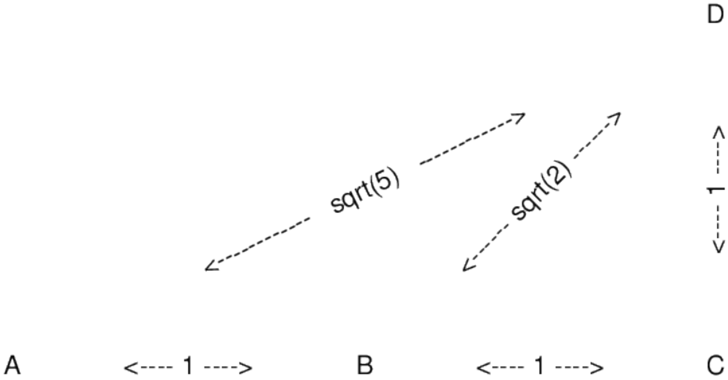
- All states equidistant – $n - 1$ dimensions

0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0

More dimensions

- E.g., 2D picture of inter-party distances: location on left–right scale, plus on pro-/anti-EU scale
- Distances are Euclidean or other metric (e.g., L1)
  - Euclidean: $\sqrt{\sum_i (r_i - s_i)^2}$
  - L1 (city block): $\sum_i |r_i - s_i|$
- Generalises easily to many dimensions
- Problem: how to weight different dimensions?
  - Scale by standard deviation? Substantive importance?

2-D example



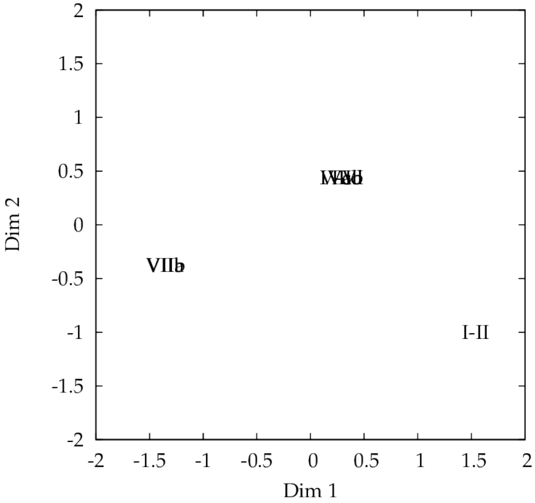
Spatial structure of theoretical spaces

- We can analyse “theoretically-informed” or *ad hoc* state spaces spatially
- Principle components analysis of substitution matrix
- Examples: Halpin and Chan (1998) McVicar and Anyadike-Danes (2002):

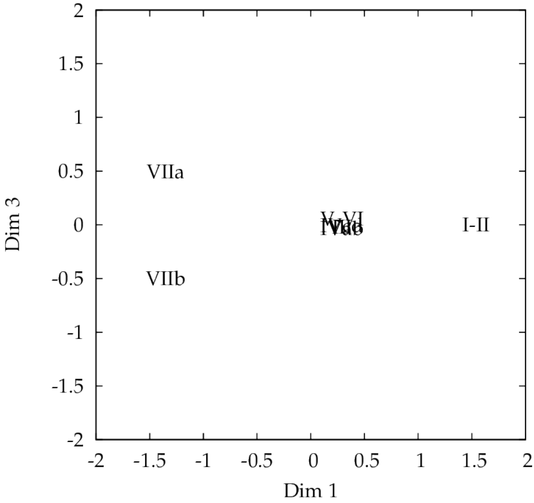
I–II	0	2	2	2	2	3	3
III	2	0	1	1	1	2	2
IV*ab*	2	1	0	1	1	2	2
IV*cd*	2	1	1	0	1	2	2
V–VI	2	1	1	1	0	2	2
VII*a*	3	2	2	2	2	0	1
VII*b*	3	2	2	2	2	1	0

E	0	1	1	2	1	3
F	1	0	1	2	1	3
H	1	1	0	2	1	2
S	2	2	2	0	1	1
T	1	1	1	1	0	2
U	3	3	2	1	2	0

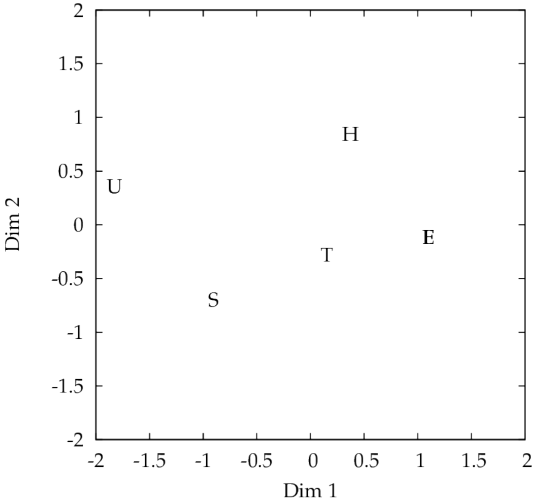
H&C, 1st two PCA dimensions



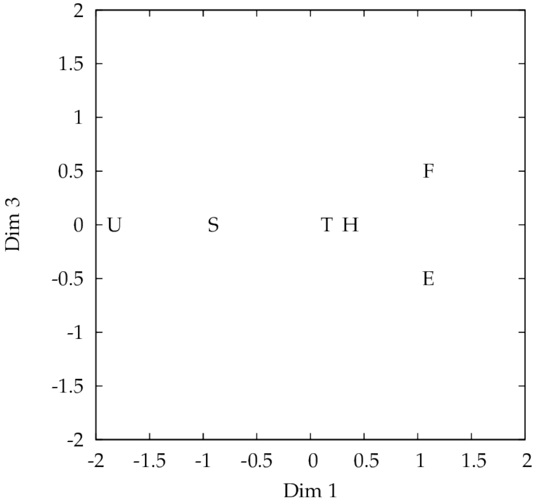
H&C, dimensions 1 & 3



MVAD, 1st two dimensions



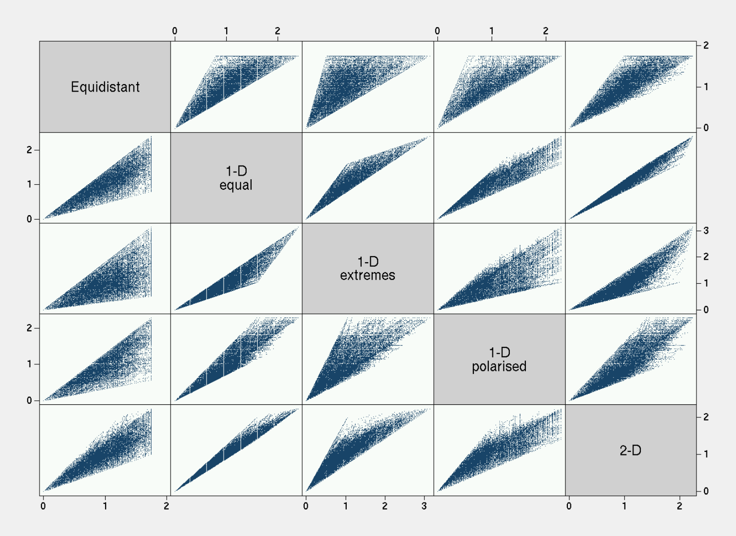
MVAD, dimensions 1 & 3



Structure passes through

- State space structure passes through to trajectory space structure
    - Distances between states clearly affect distances between trajectories containing high proportions of those states
        - If $d(\texttt{"A"},\texttt{"B"}) << d(\texttt{"A"},\texttt{"C"})$ then $D(\texttt{"..AAAA.."},\texttt{"..BBB.."})$ will tend to be less than $D(\texttt{"..AAAA.."},\texttt{"..CCC.."})$
    - Differential distances promote alignment: `AADDAAA` and `AAADDAA` are more likely to be aligned to match the `DD` if $d(\texttt{"A"},\texttt{"D"})$ is large
    - If the state distances are non-metric, the trajectory distances may also be non-metric (at least between trajectories consisting of near 100% one state)
    - Unidimensional states spaces will tend to be reflected strongly in 1st principle component of trajectory space

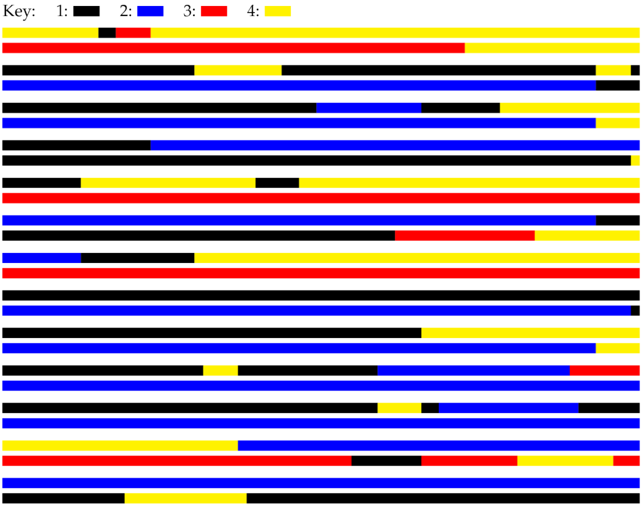
Comparing effects



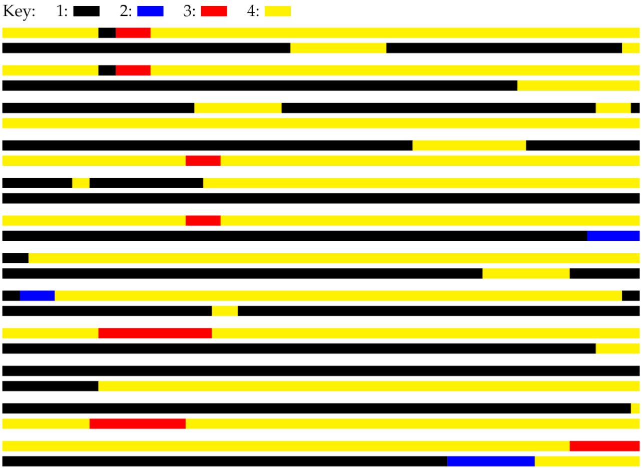
Correlations

Equidistant	1.00				
1-D equal	0.85	1.00			
1-D extremes	0.66	0.93	1.00		
1-D polarised	0.83	0.94	0.81	1.00	
2-D	0.87	0.98	0.91	0.90	1.00

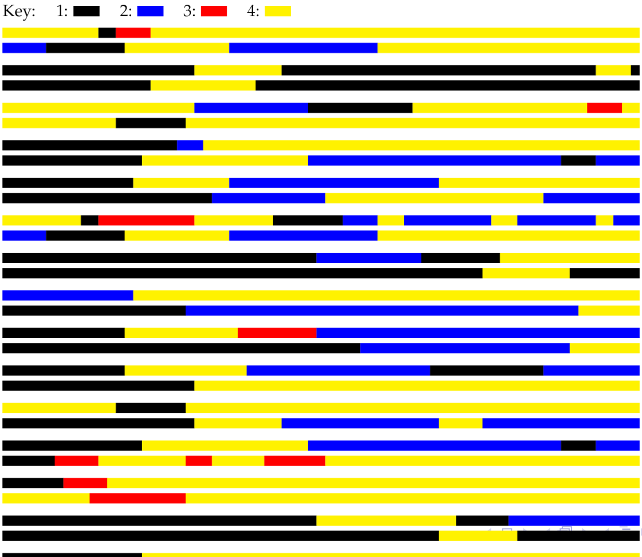
Equidistant relatively greater than 1-D



Equidistant relatively less than 1-D



Equidistant close to 1-D



Designing state spaces

- Be explicit about state spaces and what distances mean
- Think spatially
  - Choose high or low dimensions, but have your reasons
- Simplify state space as far as possible
  - Drop irrelevant distinctions
  - Drop longitudinal information: let the sequence encode the temporal information, make state space cross-sectional

Dropping temporal information

- e.g., Simplify marital status:

	Living alone	Living with partner
Legally married	Separated	Married
Not legally married	Single, never married, post-cohabitation, divorced	Cohabiting

- The sequence will distinguish adequately between the various “single” states
- Parity sequences: Women’s annual fertility history
  - in parity terms: 000112333344444
  - in birth event terms: 000101100010000

Costing OM: a tractable problem

- Substitution costs make a big difference
  - but largely understandable in operation
  - and an asset – more meaningful state space, more meaningful trajectory space
- Think spatially! Use data and geometric models
- Simplify
- Let the sequence do the temporal work

SA and further analysis

- With pairwise distances or a cluster solution we can move on to conventional analysis:
  - Explain the clusters: who goes where?
  - Predict from the clusters: do they have consequences for the future?
- Approaches: tabular, ANOVA, regression, logit
- Using clusters, MDS dimensions or other summaries of the distances

Explaining cluster membership, MVAD data

```
. tab g8 funemp, chi

          |        funemp
       g8 |         0          1 |     Total
----------+----------------------+----------
        1 |     13.28      11.97 |     13.06
        2 |     22.52      24.79 |     22.89
        3 |      9.41       5.13 |      8.71
        4 |     20.84      18.80 |     20.51
        5 |      8.24      17.09 |      9.69
        6 |      3.03      10.26 |      4.21
        7 |      6.89       5.13 |      6.60
        8 |     15.80       6.84 |     14.33
----------+----------------------+----------
    Total |    100.00     100.00 |    100.00

     Pearson chi2(7) =  28.5978   Pr = 0.000
```

```
. tab g8 gcse5eq, chi

           |        gcse5eq
        g8 |         0          1 |     Total
-----------+----------------------+----------
         1 |     17.26       5.77 |     13.06
         2 |     29.87      10.77 |     22.89
         3 |      2.21      20.00 |      8.71
         4 |     20.80      20.00 |     20.51
         5 |     13.05       3.85 |      9.69
         6 |      5.75       1.54 |      4.21
         7 |      6.64       6.54 |      6.60
         8 |      4.42      31.54 |     14.33
-----------+----------------------+----------
     Total |       452        260 |       712

     Pearson chi2(7) = 209.0925   Pr = 0.000
```

Association between covariates and clustering

- Where we have outcome variables, we may want to see how well they are predicted by the cluster solution
- Here one question is whether the cluster solution has additional explanatory power over and above simple summaries such as cumulated duration
- Nested model test (pretend, for the example, that `grammar` is an outcome)

```
cumuldur state*, cd(cd) nstates(6)
logit grammar cd1-cd5
est store base
logit grammar cd1-cd5 i.g8
lrtest base
```

Beating cumulated duration

```
Logistic regression                               Number of obs   =        712
                                                  LR chi2(12)     =     107.71
                                                  Prob > chi2     =     0.0000
Log likelihood = -283.04946                       Pseudo R2       =     0.1598

------------------------------------------------------------------------------
     grammar |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         cd1 |   .0404702   .0259219     1.56   0.118    -.0103358    .0912761
         cd2 |   .0064551   .0278439     0.23   0.817    -.0481178    .0610281
         cd3 |   .0527723   .0262769     2.01   0.045     .0012706     .104274
         cd4 |   .0036833   .0259473     0.14   0.887    -.0471725    .0545391
         cd5 |   .0260562   .0278449     0.94   0.349    -.0285188    .0806312
             |
          g8 |
          2  |    .803025    .562242     1.43   0.153    -.2989491    1.904999
          3  |   1.263318   .9776174     1.29   0.196    -.6527766    3.179413
          4  |   1.752938   .6169286     2.84   0.004     .5437803    2.962096
          5  |   .9323015   .8809664     1.06   0.290    -.7943608    2.658964
          6  |   2.599953   1.522719     1.71   0.088    -.3845203    5.584427
          7  |   2.348554    .815007     2.88   0.004     .7511697    3.945939
          8  |   3.368678   1.034953     3.25   0.001     1.340208    5.397148
             |
       _cons |   -5.30223   1.884739    -2.81   0.005    -8.996251   -1.608209
------------------------------------------------------------------------------

Likelihood-ratio test                                 LR chi2(7)  =      21.03
(Assumption: base nested in .)                        Prob > chi2 =     0.0037
```

MDS and modelling

- It may make sense to model with the MDS dimensions

```
mdsmat oma, dim(3)
matrix dim=e(Y)
svmat dim
logit grammar cd1-cd5 dim1-dim3
lrtest base
```

MDS dimensions and model

```
Logistic regression                               Number of obs   =        712
                                                  LR chi2(8)      =      95.84
                                                  Prob > chi2     =     0.0000
Log likelihood = -288.98292                       Pseudo R2       =     0.1422

------------------------------------------------------------------------------
     grammar |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         cd1 |   .0361684     .10541     0.34   0.732    -.1704313    .2427681
         cd2 |   .0529717   .1157715     0.46   0.647    -.1739364    .2798797
         cd3 |   .0727049    .094217     0.77   0.440     -.111957    .2573669
         cd4 |   .0083839   .0486104     0.17   0.863    -.0868908    .1036585
         cd5 |   .0172631   .0781496     0.22   0.825    -.1359074    .1704336
        dim1 |  -.9561052   2.407083    -0.40   0.691    -5.673902    3.761691
        dim2 |   1.942324   .7237847     2.68   0.007     .5237325    3.360916
        dim3 |   1.408145   1.796422     0.78   0.433    -2.112777    4.929068
       _cons |  -4.346376   6.378463    -0.68   0.496    -16.84793    8.155182
------------------------------------------------------------------------------

. lrtest base

Likelihood-ratio test                                 LR chi2(3)  =      9.16
(Assumption: base nested in .)                        Prob > chi2 =    0.0272
```

MDS correlated?

```
. corr cd* dim*
(obs=712)

             |      cd1      cd2      cd3      cd4      cd5      cd6
-------------+------------------------------------------------------
         cd1 |   1.0000
         cd2 |  -0.3075   1.0000
         cd3 |  -0.6320   0.0022   1.0000
         cd4 |  -0.4384  -0.2480   0.5044   1.0000
         cd5 |  -0.0393  -0.2969  -0.3062  -0.2696   1.0000
         cd6 |  -0.2772  -0.1232  -0.2111  -0.1194   0.0408   1.0000
        dim1 |   0.7224   0.2218  -0.3431  -0.4694  -0.0406  -0.7369
        dim2 |  -0.0326  -0.3829   0.3578   0.7098  -0.0525  -0.4964
        dim3 |   0.5810  -0.6630  -0.6685  -0.1359   0.3294   0.3453
```

Studer et al's discrepancy

- Studer et al. (2011) propose a method for treating distances matrices analogously to SS in regression and ANOVA
- The average distance to the centre of the whole matrix is the analogue of total sum of squares
- With a grouping variable, the distance to the centre for each groups is the residual sum of squares
- This allows a pseudo-R^2 and a pseudo-F test
- Permutation is used to approximate the sampling distribution of pseudo-F

Discrepancy and MVAD

```
use mvad

matrix md = (0, 1, 1, 2, 1, 3\ ///
             1, 0, 1, 2, 1, 3\ ///
             1, 1, 0, 2, 1, 2\ ///
             2, 2, 2, 0, 1, 1\ ///
             1, 1, 1, 1, 0, 2\ ///
             3, 3, 2, 1, 2, 0)
matrix rownames md = E F H S T U
matrix colnames md = E F H S T U

set matsize 1000
oma state*, subs(md) indel(1.5) pwd(oma) length(72)
discrepancy funemp, dist(oma) idvar(id) niter(1000) dcg(d2c)
```

Discrepancy results

```
. discrepancy funemp, dist(oma) idvar(id) niter(100) dcg(d2c)

Discrepancy based R2 and F, 100 permutations for p-value

             | pseudo R2   pseudo F    p-value
-------------+---------------------------------
      funemp |   .007956   5.694094        .17

----------------------------------------------------------
   funemp |     N(d2c)    min(d2c)   mean(d2c)    max(d2c)
----------+-----------------------------------------------
        0 |        595    .2215114     .463736    1.919831
        1 |        117    .2757618    .5502117    1.518995
----------------------------------------------------------
```

Alternatives to OM and Hamming

- OMA is the dominant but not the only approach
- It receives justified and unjustified criticism in terms of its fit to lifecourse data
- One axis of critique relates to costs: Dynamic Hamming sidesteps this
- Another relates to whether token strings are:
  - a good way to represent life-course processes (continuous time, discrete state space, infrequent transitions)
  - and whether operations on token-strings match sociological difference

Alternatives

- Hollister's LOM and my OMv attempt to fix OM by paying attention to the local context of operations (but fail: non-metric)
- TWED "warps time" and has more sensitivity to spell order
- Lesnard's Dynamic Hamming estimates substitution costs from the data and does no alignment
- Elzinga's duration-weighted combinatorial measures pay strict attention to spell order and duration
- See Halpin (2014b) for a discussion
- See Studer and Ritschard (2014) for a comprehensive review of distance measures

An aside: Metric spaces

- To treat a dissimilarity as a distance, it must be compatible with a "metric space"
- Everyday 3D Euclidean space is metric, but we can relax many of the characteristics of Euclidean space and still think in spatial terms, using e.g., cluster analysis and MDS
- Four conditions are required
  - $d(x,x) = 0$; identity
  - $d(x,y) \geq 0$; non-negativity
  - $d(x,y) = d(y,x)$; symmetry
  - $d(x,y) \leq d(x,z) + d(z,y)$; the "triangle inequality"
- LOM and OMv do not satisfy the triangle inequality

Hollister's Localised OM

- Hollister argues that OM's elementary operations need to take into account the context: the adjacent states, at least
- Inserting a B between two Bs is cheaper than between an A and a C
- Operates very like OM, with substitution costs, but a modified approch to indels
- To insert element k between elements i and j the indel cost is:

$$\iota = \alpha \frac{\delta_{i,k} + \delta_{j,k}}{2} + \beta$$

where α and β are chosen by the analyst

LOM non-metric

Hollister's measure violates the triangle inequality for the following trio:

- BBBBAB, CCCACC and BBBACC

For a substitution cost of 1, α 0.5 and β 0.5 (i.e., $\iota = 0.5\frac{\delta_{i,k}+\delta_{j,k}}{2} + 0.5$), the direct distance between sequences 1 and 2 is 6 units. However, the indirect distance passing through sequence 3 is 5.5 (2.5 plus 3):

	Distance		
	LOM	OM	
Pair	$\delta = 1, \alpha = \beta = 0.5$	$\iota = 1.0$	$\iota = 0.75$
1, 2	6	6	5.5
1, 3	2.5	3	2.5
2, 3	3	3	3

Halpin's duration-adjusted OMv

- My approach had a very similar motivation: operations should be weighted less in big spells, more in short ones
- Scale indel and substitution costs according to the square-root of the spell length
- Also non-metric: sequences with long spells are closer to all other spells, without affecting distances between other spells

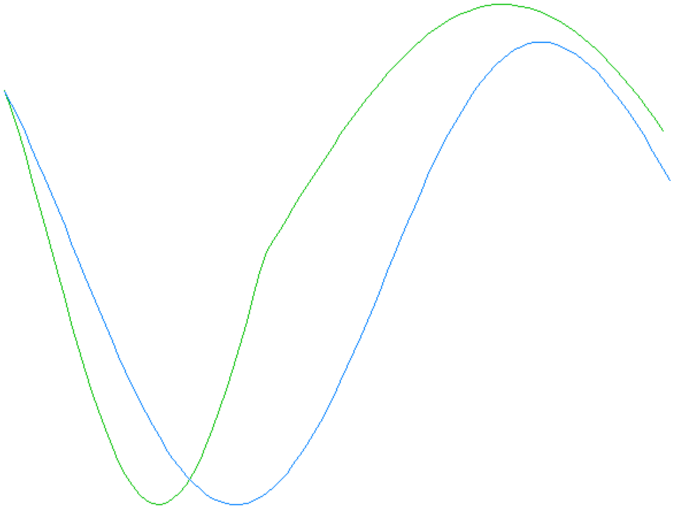
Warping time

- What of time-warping?
- Abbott and Hrycak (1990) use the term to suggest non-linear time scales
- OMv "warps time" by weighting it differently in different spells
- In turn informed by Sankoff and Kruskal (1983), *Time Warps, String Edits and Macromolecules*
- But time-warping refer to a specific set of algorithms

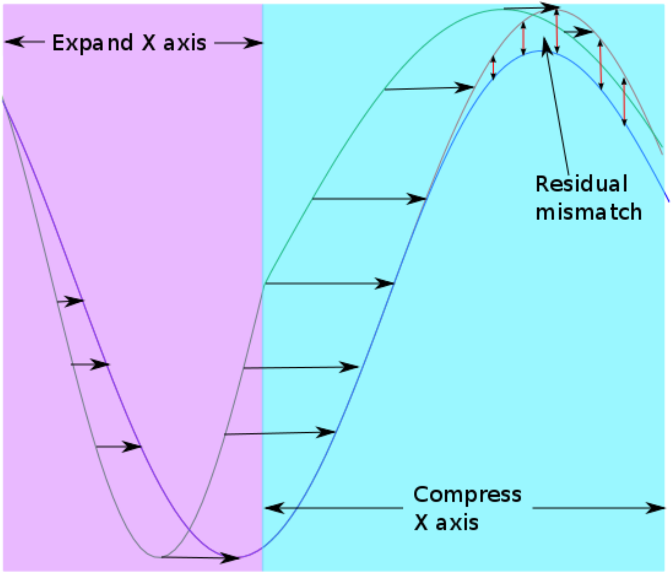
Time warping algorithms

- Formally, time warping is a family of algorithms that do “continuous time-series to time-series correction” while OM *et al* do “string to string correction” (Marteau, 2007)
- Focus on comparing pairs of continuous-time high-dimensional time-series in $\mathbb{R}^n$
- Operates by locally compressing or expanding the time scale of one trajectory to minimise the distance to the other
- Distance is usually Euclidean in $\mathbb{R}^n$ or other simple distance

TWED: Matching 1D series



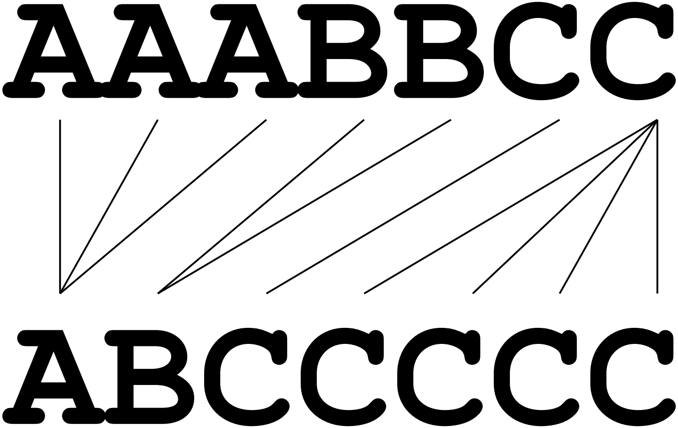
TWED: Compress and expand



TW algorithms

- TW used widely: was used for speech recognition, signature verification, other machine learning tasks
- Typically used to match a high-dimensional time-series to a “dictionary” of standard elements
- Conceptually it is a continuous time approach but implementations must be discrete – sampling or periodic summaries:
  - e.g., sound sampled at 41 kHz
  - rainfall summarised daily
  - employment history reported monthly
- Kruskal and Liberman (1983) show that the continuous time logic can be faithfully implemented with discretised series

Discrete time-warping



AAABBCC

ABCCCCC

TW with stiffness penalty: TWED

- Violation of the triangle inequality is due to TW usually having no cost to expansion or compression, only to the residual point-by-point distance
- Marteau (2007, 2008) proposes a TW algorithm that has a "stiffness" penalty
- Satisfies the triangle inequality
- Can be programmed very similarly to OM (recursive algorithm)
- Stiffness penalty like but not like *indel* cost – squeezing/stretching, not inserting/deleting
- Point-to-point distance just like substitution

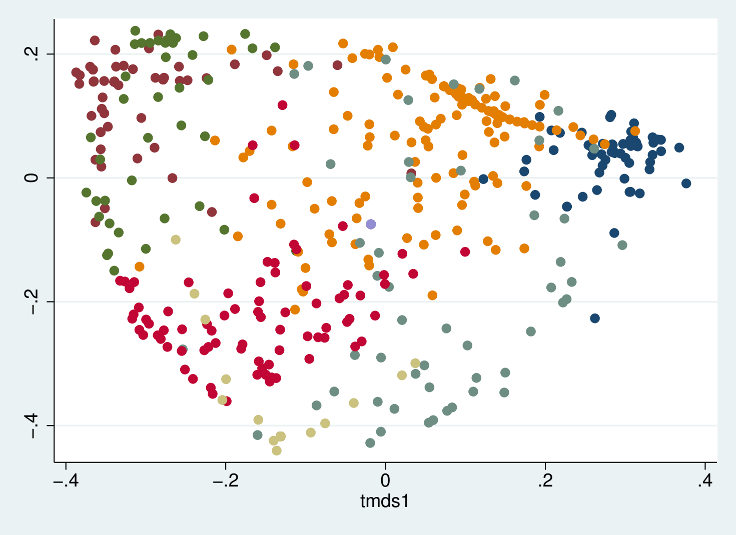
TWED: Recursive algorithm

TW distance, $\delta(A^p, B^q) =$

$$min \begin{cases} \delta(A^{p-1}, B^q) & + d_{LP}(a_p, a_{p-1}) + \gamma d_{LP}(t_{a_p}, t_{a_{p-1}}) + \lambda \\ \delta(A^{p-1}, B^{q-1}) & + d_{LP}(a_p, b_q) \quad + \gamma d_{LP}(t_{a_p}, t_{b_q}) \\ \delta(A^p, B^{q-1}) & + d_{LP}(b_q, b_{q-1}) + \gamma d_{LP}(t_{b_q}, t_{b_{q-1}}) + \lambda \end{cases}$$

(Marteau, 2007)

MDS/Cluster with TWED



TWED attractive

- TWED has a completely different "narrative" from OM: warping time rather than editing token strings
- Nonetheless, gives results that are not radically different
- More noticeable differences for more complex sequences
- For high values of λ and γ, tends to yield Hamming distance
- For very low values of λ and γ, closer (but still not that close) to X/t
- Distribution in sequence space more like OM than X/t

Dynamic Hamming

- Dynamic Hamming takes a completely different slant: no alignment
- Similarity at the same time only, where similarity is defined by time-dependent transition patterns
  - While changes are common differences matter less
  - While change is rare, differences are more marked
- Naturally appropriate for "clock" time, e.g., daily, weekly, annual patterns
- Less obviously appropriate for "developmental" time, where a common feature is people taking the same route at different speeds
- Lesnard (2006); Lesnard and de Saint Pol (2009); Lesnard (2010), implemented by him (`seqcomp`), in Traminer and `SADI`

Combinatorial approaches

- Combinatorial methods are a completely different approach to sequence comparison
- Proposed by Elzinga (2003, 2005)
- Compare sequences in terms of common “subsequences” rather than string-edits

Counting sequences

- The sequence `ABC` has as subsequences:
  - the null (empty) string
  - `A`, `B` and `C`
  - `AB`, `AC` and `BC`
  - and `ABC` itself
- A sequence of length l has 2^l subsequences
- If elements are repeated not all subsequences are distinct

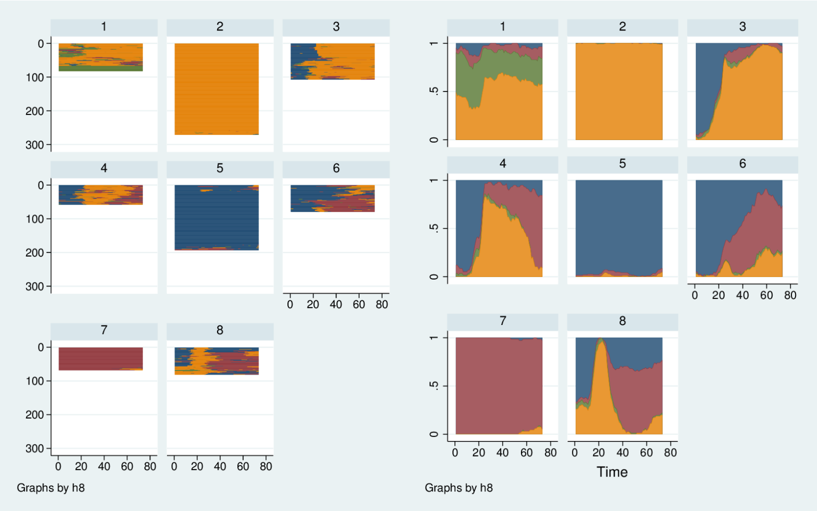
Combinatorial measures

- Elzinga has proposed a number of measures that count subsequences
  - Longest common subsequence
  - Number of common subsequences
  - Number of matching subsequences
- A completely different logic, combinatorial rather than string-editing: "the same states in the same order"
- One particularly attractive approach: number of matching spell-subsequences weighted by duration (I refer to it as "X/t")

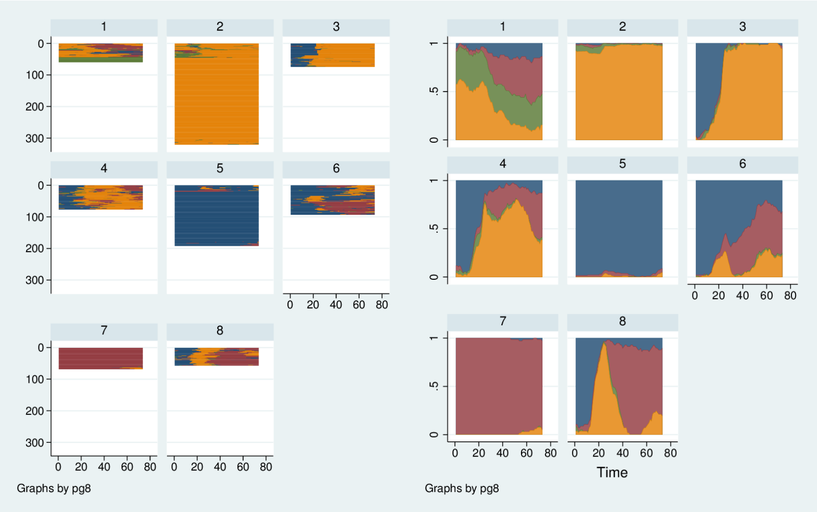
Code to run all the measures

```
use bsseq

set matsize 1000
matrix sm = (0,1,2,3\1,0,1,2\2,1,0,1\3,2,1,0)
matrix fl = (0,1,1,1\1,0,1,1\1,1,0,1\1,1,1,0)

hamming    state1-state72, subs(sm) pwd(ham)
oma        state1-state72, subs(sm) indel(1.5) pwd(om) len(72)
twed       state1-state72, subs(sm) nu(0.5) lambda(0.5) pwd(twd) len(72)

hamming    state1-state72, subs(fl) pwd(haf)
oma        state1-state72, subs(fl) indel(0.5) pwd(of) len(72)
twed       state1-state72, subs(fl) nu(0.5) lambda(0.5) pwd(twf) len(72)
dynhamming state1-state72, pwd(dyn)

preserve
combinprep, state(state) length(l) nspells(nsp) idvar(pid)
combinadd state1-l‘r(maxspells)’, pws(xtd) nsp(nsp) nstates(‘r(nels)’) rtype(d)
restore
```

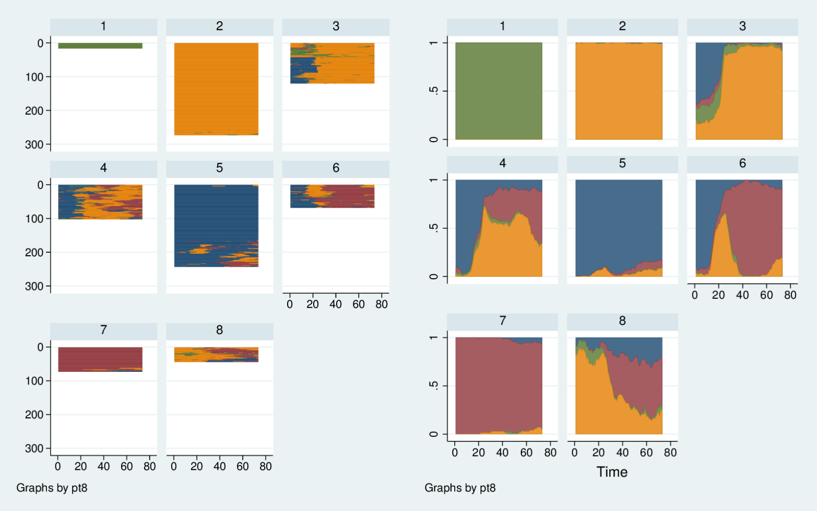
Hamming, linear matrix



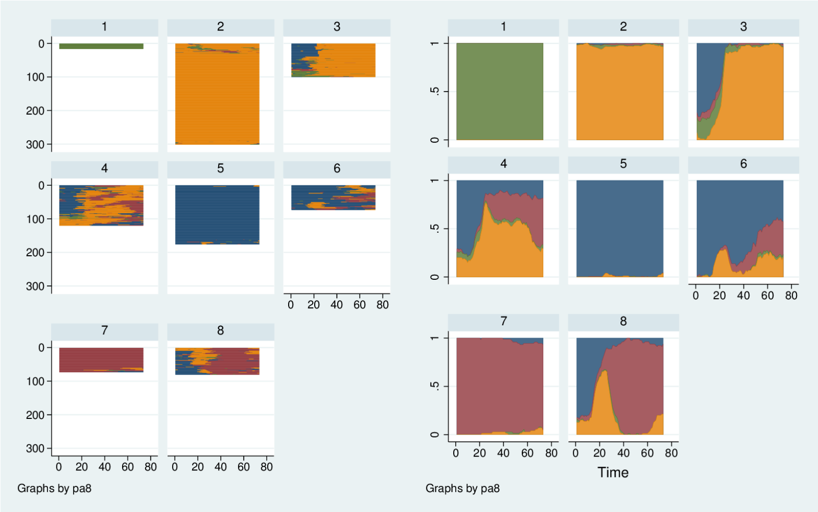
OM, linear matrix



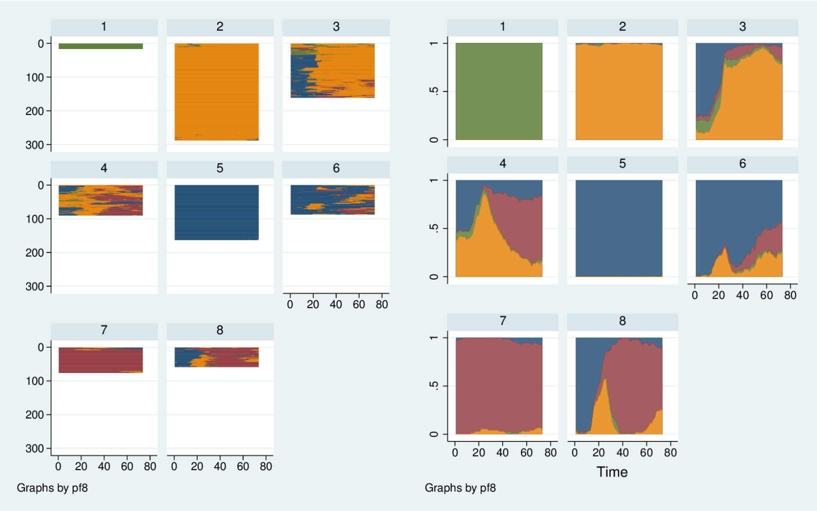
TWED, linear matrix



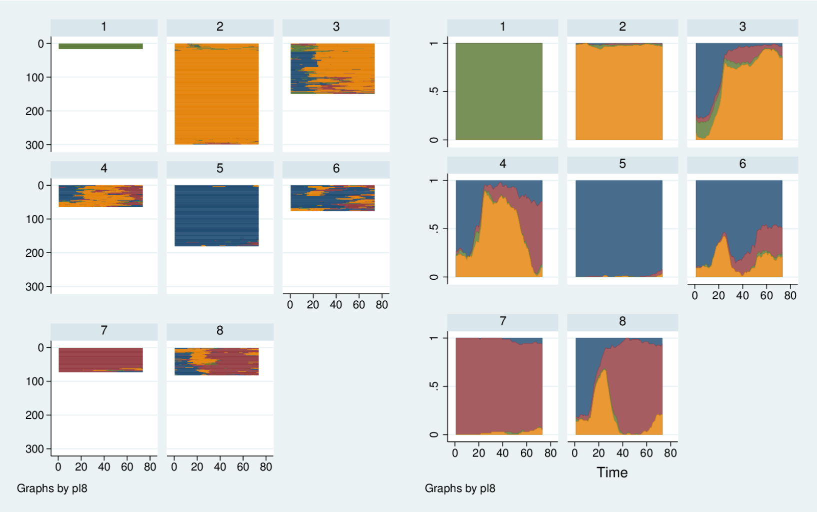
Hamming, flat matrix



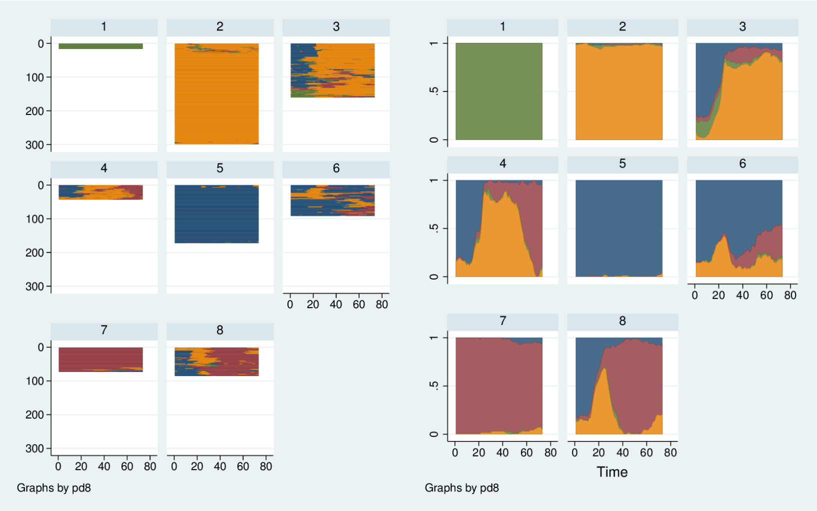
OM, flat matrix



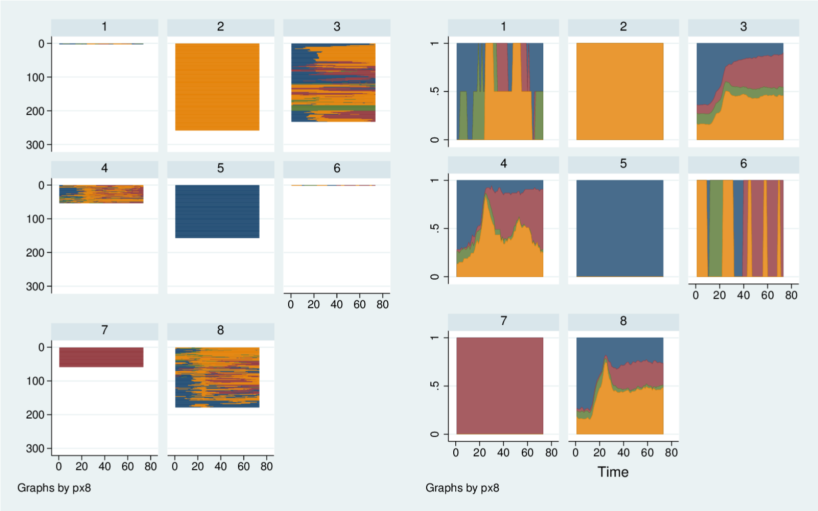
TWED, flat matrix



Dynamic Hammming



X/t


1
2
3
4
5
6
7
8
0
100
200
300
0
20
40
60
80
Graphs by px8
1
2
3
4
5
6
7
8
1
.5
0
0
20
40
60
80
Time
Graphs by px8

Multiple domains

- Lifecourse analysis recognises the interrelatedness of domains
- Somewhat hard to handle in many approaches: a potential strength of SA?
- In practice, not very well developed; most research on single domains
- Some work (Dijkstra and Taris (1995), Pollock (2007), Gauthier et al. (2010))

Combined distance versus combining distances

- How to proceed?
- Conduct parallel analyses and combine results?
- Combine domains into a single variable?
- The former is easy but will be less sensitive to the synchronisation of domains
- The latter involves a large state space and problem in defining distances
- However, better sensitivity to cross-domain features makes it attractive

Combine by cross-tabulation

- The simplest approach is to create a new state space that is the cross-tabulation of the two (or more) domains
- This yields a large number of states, one for each combination
- How then to determine costs?

Determining costs

- Simplest strategy is to sum across the domains
- In short, $d^{AB}_{ik,jl} = d^{A}_{i,j} + d^{B}_{k,l}$
- There may be justification for imposing other patterns, for instance,
  - imposing a ceiling
  - changing d^A for certain values in domain B
  - weighting the domains differentially
- Note that with two different substitution matrices it can be difficult to weight equally
  - equalise by max substitution cost?
  - equalise by average substitution cost?
  - equalise by average substitution cost weighted by occurrence in the data?

Implementation

- We take a simple case (four parity levels and five employment statuses)
- First step is to create the interaction or crosstabulation of the states

```
// Reshape long to work on all months simultaneously
reshape long parx emp, i(pid) j(month)

// Create a variable that is the interaction of the two
gen cross = emp+(parx-1)*5

// Verify the state interaction variable
tab cross
table parx emp, c(mean cross)

// Back to wide, fix the variable order
reshape wide parx emp cross, i(pid) j(month)
order pid parx* emp* cross*
```

Create the substitution cost matrix

- We have two substitution cost matrices, 4x4 and 5x5:

```
matrix spar = (0,1,2,3\ ///
               1,0,1,2\ ///
               2,1,0,1\ ///
               3,2,1,0)
```

```
matrix semp = (0,1,2,3,3\ ///
               1,0,1,2,2\ ///
               2,1,0,1,1\ ///
               3,2,1,0,1\ ///
               3,2,1,1,0)
```

- Both have a max of 3, otherwise perhaps divide each by its max

Combine into 20x20

```
// Use Mata to combine the two matrices
mata:
spar = st_matrix("spar")
semp = st_matrix("semp")

// each element becomes a 5x5 block
sparx = spar # J(1,5,1) # J(5,1,1)

// replicate the 5x5 matrix 4x4 times
sempx = semp
for (i=2; i<=4; i++) {
  sempx = sempx,semp
}
sempxy = sempx
for (i=2; i<=4; i++) {
  sempxy = sempxy\sempx
}

// The combined matrix is the element-wise sum; return it from Mata to Stata
st_matrix("mcsa", sempxy :+ sparx)
end
```

The combined matrix

```
symmetric mcsa[20,20]
      c1  c2  c3  c4  c5  c6  c7  c8  c9c10c11c12c13c14c15c16c17c18c19c20
 r1    0
 r2    1   0
 r3    2   1   0
 r4    3   2   1   0
 r5    3   2   1   1   0
 r6    1   2   3   4   4   0
 r7    2   1   2   3   3   1   0
 r8    3   2   1   2   2   2   1   0
 r9    4   3   2   1   2   3   2   1   0
r10    4   3   2   2   1   3   2   1   1   0
r11    2   3   4   5   5   1   2   3   4   4   0
r12    3   2   3   4   4   2   1   2   3   3   1   0
r13    4   3   2   3   3   3   2   1   2   2   2   1   0
r14    5   4   3   2   3   4   3   2   1   2   3   2   1   0
r15    5   4   3   3   2   4   3   2   2   1   3   2   1   1   0
r16    3   4   5   6   6   2   3   4   5   5   1   2   3   4   4   0
r17    4   3   4   5   5   3   2   3   4   4   2   1   2   3   3   1   0
r18    5   4   3   4   4   4   3   2   3   3   3   2   1   2   2   2   1   0
r19    6   5   4   3   4   5   4   3   2   3   4   3   2   1   2   3   2   1   0
r20    6   5   4   4   3   5   4   3   3   2   4   3   2   2   1   3   2   1   1   0
```

Dyadic SA

- SA typically uses all-pair-wise distances, or distance to special cases
- Dyadic SA is also useful: distance between a specific pair
  - Couple time-diaries
  - Couple labour market histories
  - Mother–daughter fertility histories, etc.

Research questions

- Allows testing hypotheses about dyadic similarity
  - Are couples' time-use patterns or life-course histories aligned
  - Are fertility patterns inherited?
  - Under what conditions are dyadic distances smaller or larger?
  - How do couples arrange joint lifecourses?

Similarity and difference

- Couples may coordinate their lives under very different gender constraints
- Fertility patterns may be similar within the constraints of different cohort patterns of fertility
- The relationship between sequences may not be one of replication
    - some daughters may completely reject their mother's fertility pattern

Literature

- Off-scheduling (Lesnard, 2008) Dyadic in concept but actually creates combined sequences
- Robette et al. (2015): Mother–daughter labour market careers
- Fasang and Raab (2014): Intergenerational fertility; notes that focus on similarity ignores heterogeneity
- Raab et al. (2014): Jun 13 2015 15:18:18 Sibling dyads, fertility

Practical issues

- We can calculate dyadic distances with standard software
- For efficiency it might better to just calculate dyads' distances
- But the cost of calculating all pairs is relatively small, and offers an advantage:
    - Compare dyadic distances with distances to all others

Strategy: Begin with dyad-ordered data

	Dyad	1	1	2	2	3	3	4	4
Type		M	D	M	D	M	D	M	D
M	1	11	12	13	14	15	16	17	18
D	1	21	22	23	24	25	26	27	28
M	2	31	32	33	34	35	36	37	38
D	2	41	42	43	44	45	46	47	48
M	3	51	52	53	54	55	56	57	58
D	3	61	62	63	64	65	66	67	68
M	4	71	72	73	74	75	76	77	78
D	4	81	82	83	84	85	86	87	88

Sort by types

	Dyad	1	2	3	4	1	2	3	4
Type		D	D	D	D	M	M	M	M
D	1	22	24	26	28	21	23	25	27
D	2	42	44	46	48	41	43	45	47
D	3	62	64	66	68	61	63	65	67
D	4	82	84	86	88	81	83	85	87
M	1	12	14	16	18	11	13	15	17
M	2	32	34	36	38	31	33	35	37
M	3	52	54	56	58	51	53	55	57
M	4	72	74	76	78	71	73	75	77

Submatrices

- Two submatrices, with distances from each mother to each daughter (and transpose)
- Distance from mother to her own daughter on diagonal (and transpose)
- Use distance from mother to all daughters to assess whether distance to own daughter is unusual

Submatrices

	Pair	1	2	3	4
Type		M	M	M	M
D	1	21	23	25	27
D	2	41	43	45	47
D	3	61	63	65	67
D	4	81	83	85	87

	Pair	1	2	3	4
Type		D	D	D	D
M	1	12	14	16	18
M	2	32	34	36	38
M	3	52	54	56	58
M	4	72	74	76	78

Extract diagonals and other information

- The main info is on the diagonals: the dyad distances (repeated across the two submatrices since distance is symmetric)
- Other summaries are also interesting
  - mean distance of each daughter to all mothers (and vice versa)
  - variance, standard deviation of this distance
  - z-score of dyad distance relative to all distances
  - rank of dyad distance compared with all distances

Abbott, A. and Hrycak, A. (1990). Measuring resemblance in sequence data: An optimal matching analysis of musicians' careers. *American Journal of Sociology*, 96(1):144–85.

Barban, N. and Billari, F. (2012). Classifying life course trajectories: A comparison of latent class and sequence analysis. *Journal of the Royal Statistical Society Series C*, 61(5):765–784.

Billari, F. C., Fürnkranz, J., and Prskawetz, A. (2006). Timing, sequencing and quantum of life course events: A machine learning approach. *European Journal of Population*, 22:37–65.

Dijkstra, W. and Taris, T. (1995). Measuring the agreement between sequences. *Sociological Methods and Research*, 24(2):214–231.

Elzinga, C. H. (2003). Sequence similarity: A non-aligning technique. *Sociological Methods and Research*, 32(1):3–29.

Elzinga, C. H. (2005). Combinatorial representations of token sequences. *Journal of Classification*, 22(1):87–118.

Elzinga, C. H. (2010). Complexity of categorical time series. *Sociological Methods and Research*, 38(3):463–481.

Fasang, A. and Raab, M. (2014). Beyond transmission: Intergenerational patterns of family formation among middle-class american families. *Demography*, 51(5):1703–1728.

Gabadinho, A. (2014). Package 'pst'. probabilistic suffix trees and variable length Markov chains. Technical report, CRAN.

Gauthier, J.-A., Widmer, E. D., Bucher, P., and Notredame, C. (2010). Multichannel sequence analysis applied to social science data. *Sociological Methodology*, 40(1):1–38.

Halpin, B. (2013). Sequence analysis. In Baxter, J., editor, *Oxford Bibliographies in Sociology*. Oxford University Press, New York.

Halpin, B. (2014a). SADI: Sequence analysis tools for Stata. Working Paper WP2014-03, Dept of Sociology, University of Limerick, Ireland.

Halpin, B. (2014b). Three narratives of sequence analysis. In Blanchard, P., Bühlmann, F., and Gauthier, J.-A., editors, *Advances in Sequence Analysis: Theory, Method, Applications*. Springer, Berlin.

Halpin, B. and Chan, T. W. (1998). Class careers as sequences: An optimal matching analysis of work-life histories. *European Sociological Review*, 14(2).

Kruskal, J. B. and Liberman, M. (1983). The symmetric time-warping problem. In Sankoff and Kruskal (1983), pages 125–161.

Lesnard, L. (2006). Optimal matching and social sciences. Document du travail du Centre de Recherche en Économie et Statistique 2006-01, Institut Nationale de la Statistique et des Études Économiques, Paris.

Lesnard, L. (2008). Off-scheduling within dual-earner couples: An unequal and negative externality for family time. *American Journal of Sociology*, 114(2):447–90.

Lesnard, L. (2010). Setting cost in optimal matching to uncover contemporaneous socio-temporal patterns. *Sociological Methods and Research*, 38(3):389–419.

Lesnard, L. and de Saint Pol, T. (2009). Patterns of workweek schedules in France. *Social Indicators Research*, 93:171–176.

Lovaglio, P. G. and Mezzanzanica, M. (2013). Classification of longitudinal career paths. *Quality and Quantity*, 47(2):989–1008.

Marteau, P.-F. (2007). Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching. *ArXiv Computer Science e-prints*.

Marteau, P.-F. (2008). Time Warp Edit Distance. *ArXiv e-prints*.

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

Pollock, G. (2007). Holistic trajectories: A study of combined employment, housing and family careers by using multiple-sequence analysis. *Journal of the Royal Statistical Society: Series A*, 170(1):167–183.

Raab, M., Fasang, A. E., Karhula, A., and Erola, J. (2014). Sibling similarity in family formation. *Demography*, 51(6):2127–2154.

Robette, N., Bry, X., and Éva Lelièvre (2015). A "global interdependence" approach to multidimensional sequence analysis. *Sociological Methodology*, Online advance copy.

Sankoff, D. and Kruskal, J. B., editors (1983). *Time Warps, String Edits and Macromolecules*. Addison-Wesley, Reading, MA.

Studer, M. and Ritschard, G. (2014). A comparative review of sequence dissimilarity measures. Working Paper 2014-33, LIVES, Geneva.

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