University of Limerick
Department of Sociology Working Paper Series

Working Paper WP2012-02
July 2012

Brendan Halpin

Department of Sociology, University of Limerick

Sequence Analysis of Life-course Data: A Comparison of Distance Measures

Available at `http://www.ul.ie/sociology/pubs/wp2012-02.pdf`

Sequence analysis of life-course data: a comparison of distance measures

Brendan Halpin
Dept of Sociology
University of Limerick
brendan.halpin@ul.ie

July 2012

Contents

1 The importance of measure in sequence analysis

Longitudinal data relating to the lifecourse, in a wide variety of domains, is increasingly common. A range of different approaches are used to analyse such data, including hazard-rate modelling, panel models, Markov chain models, latent growth-curve models, and sequence analysis. By sequence analysis is meant the holistic treatment of lifecourse trajectories by calculating similarities or distances between pairs of trajectories, viewed as whole units. Sequence analysis of lifecourse trajectories is a niche technique, of real (if bounded) value, particularly but not only for exploratory and descriptive purposes, and is discussed in a growing literature.

In this literature, one of the dominant streams focuses on "optimal matching analysis" (e.g., Kruskal, 1983; Abbott and Forrest, 1986; Abbott and Hrycak, 1990; Abbott, 1995; Chan, 1995; Halpin and Chan, 1998; Abbott and Tsay, 2000; Scherer, 2001; McVicar and Anyadike-Danes, 2002; Pollock, 2007; Müller et al., 2008; Piccarreta and Lior, 2010), using the optimal matching algorithm (OM) to define distances between sequences. OM has been the object of a range of criticisms, some more and some less deserved, bearing on the extent to which it can be expected to detect sociologically meaningful patterns (e.g.,

Wu, 2000; Levine, 2000; McVicar and Anyadike-Danes, 2010; Pollock and Roberts, 2012). While a range of alternative distance measures has been proposed (e.g., Elzinga, 2003; Hollister, 2009; Halpin, 2010; Lesnard, 2010; Elzinga and Wang, 2012), the use of OM (followed often by cluster analysis or multi-dimensional scaling) is predominant.

The purpose of this paper is to compare OM with a range of competing definitions of inter-sequence distance. How do the alternative distance measures operate? How do they differ in the patterns they expose? How do they relate to sociologically meaningful differences between sequences?

The paper examines seven competing measures:

- Hamming distance
- Optimal Matching
- Hollister's localised OM
- Halpin's duration-weighted OMv
- Lesnard's dynamic Hamming distance
- Marteau's time-warp edit distance
- Elzinga's duration weighted combinatorial similarity index

The performance of the measures is compared in a number of ways:

- by simulation, testing their ability to detect differences in the processes generation the sequences, under a number of scenarios
- using real data sets, testing the ability to differentiate between real data and data simulated from the time-dependent transition structure of the data
- using real data sets, examining the association between the cluster solutions and non-sequence covariates
- by examination of the correlation structure between measures using real and simulated data, to understand how and why they differ in comparing sequences.

1.1 Describing the distance measures

The standard measures: Hamming and OM

The simplest way of mapping information about differences in a state space to differences between trajectories in the state space, is the Hamming distance. For two sequences of equal length (perhaps truncating the longer), the distance between the sequences is the sum of the period-by-period distance between the states. If the time things happen is particularly important, this is the appropriate and uncontroversial measure to use. However, if similarity is defined not as "the same or similar things happening at the same time" but as "the same or similar things happening at the same or similar time", Hamming distance can fail to observe strong similarity that is slightly out of phase. For instance, the intuitively evident similarity between `ABCD` and `DABC` will be invisible to Hamming. The Optimal Matching Algorithm improves on the Hamming distance by allowing "alignment", that is, sliding parts on one sequence along the other if this improves matching. This is done in terms of "edits", insertions and deletions to shift parts of sequences. One

way of looking at OM is to say that the aligned sequences are compared in Hamming distance terms, with an additional penalty based on how much editing was required. OM is "optimal" because the dynamic programming algorithm that calculates the distance is guaranteed to identify the shortest distance between sequence pairs, and to do so very efficiently.

OM is clearly an improvement on Hamming distance, but we must ask how important this is for life-course data. Where spells are much longer than the time-units represented by the tokens in the sequences, misalignment will matter less. Thus, the Hamming distance recognises that `AAABBBCCCD` and `DAAABBBCCC` are quite similar (though OM will see them as even more similar). In practice, as we will see, there is a very high correlation between Hamming and OM distances for life course data: we must ask if the added conceptual complexity of OM has any real benefit for analysis.

A second question to raise is whether token-sequence focused measures like Hamming and OM are appropriate for life course data, where transitions are rare. In other contexts it can be advantageous to consider life-course trajectories as sequences of spells, with durations, bounded by transition events. Hamming and OM treat each token in the context of the whole sequence, but without reference to its local context or the spell within which it is embedded. Four more distance measures are considered, each of which tries in different ways to improve on OM in this respect.

Modified OM: Hollister's local OM and Halpin's duration adjusted OM

"Localised OM" (Hollister, 2009) and "duration-adjusted OM" (Halpin, 2010) (hereafter LOM and OMv) are variants on OM that were developed independently but have substantial conceptual similarity. Both focus on the fact that OM's "elementary operations" of insertion, deletion and substitution take no account of the local context in which they operate (apart from that imposed by working through the joint structure of the sequences). Both approaches object to the fact that, for instance, the cost of deleting the second element, `B`, from both `ABA` and `BBB`, will be the same, despite the fact that the operation on the former sequence makes for a much bigger sociological change than that on the latter. In the former case it completely alters the spell structure of the sequence, while in the latter is simply shortens a spell.

LOM deals with this by adapting the OM algorithm to take account of the values of adjacent tokens in costing the elementary operations. 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.[1]

OMv's solution to this issue is to weight operations on a token less, the longer is the spell in which it is located:

$$\iota = \frac{\iota_1}{\sqrt{l}}$$

where ι_1 is the indelcost for a spell of length 1, and l is the spell length. In OMv, this logic is also applied to substitutions, on the grounds that substitutions can be considered as sequential deletion–insertion operations (with a "discounted" cost).

While the specifics of the motivation and implementation (in both cases, relatively simple adaptations of the OM dynamic programming algorithm) differ, the underlying

[1] To insert at an end of a sequence, $\iota = \alpha\delta_{i,k} + \beta$.

concern and (potentially) the consequences for which sequence pairs are seen as more or less similar, are closely related. In particular, operations that change spell length will be relatively cheaper than those that change spell structure.

However, neither measure generates metric distances. This casts a shadow on their use in further processing such as cluster analysis using conventional methods, but more importantly puts in question their claim to represent a true distance in the trajectory space. They violate the triangle inequality, which demands that for points A, B and C, the A-C distance must be less than or equal to the sum of the A-B and B-C distances. Non-metric measures arise, for instance, in situations where A may be similar to B due to shared characteristics, and B similar to C due to shared characteristics that are not shared with A, such that A and C are very dissimilar. Non-metric measures are effective in determining whether two objects are similar or not, but not necessarily in placing all objects in a coherent mutual space. However, because it is not clear how much of a problem this is in practice, both measures are retained in the subsequent analysis.

LOM and the triangle inequality

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

1. `BBBBAB`
2. `CCCACC`
3. `BBBACC`

If the substitution cost between non-identical states is 1, and the insertion cost is half the "adjacent substitution cost" plus 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

Hollister's measure deviates from OM (with indelcost of 1.0) only in one respect: the 1–3 distance is lower, because we can change s3 into s1 by inserting a B between Bs, at a reduced insertion cost of 0.5*subs(B,B)+0.5 = 0.5 and carrying out two substitutions. The reduced indelcost doesn't come into play for the other two comparisons because a substitution-only route is cheaper. For OM with an indelcost of 1.0, the s1–s3 distance is 3. If we reduce the indelcost to 0.75, the s1–s3 distance falls to 2.5, as for LOM (the reduced indelcost is greater than that effective under LOM, but it applies throughout the sequence and is triggered more than once). However, since this reduced indelcost applies to all sequences, this results in the s1–s2 distance falling also. LOM's primary advantage, the fact that its elementary operations take local context into account, turns out to be a problem: two sequences can be closer to each other because of their joint characteristics, while their distances to other sequences (where the combination of characteristics does not apply) are not reduced, resulting in violations of the triangle inequality.

OMv is also non-metric

The same problem applies to OMv: sequences with longer spells have lower indelcosts, with the result that their distance to any other sequence is systematically lower. If s2 is such a long-spell sequence, the combined distance from s1 through s2 to s3 will be smaller than under OM, even if s2 and s3 are not long-spell sequences. If s2 and s3 have many short spells, their direct distance can exceed the indirect distance, thus violating the triangle inequality.

Moreover, having uneven spell length can exacerbate this. Consider `ABCCCCCCC` and `AAABBBCCC`: the average spell length is the same but the indelcost is $\frac{\iota}{\sqrt{7}}$ for seven of the former sequence's nine elements, but $\frac{\iota}{\sqrt{3}}$ for all nine of the latter's.

For example, the distance between `BBBBAB` and `CCAAAC` is 3. However, going through `BBBBBB` the distance is $0.41 + 2.45 = 2.86$. While it is normal that the distance between `BBBBAB` and `BBBBBB` should be low, the fact that `BBBBBB` consists of a single spell means that its distance even from a spell with no shared elements such as `CCAAAC` is reduced. In fact, it is substantively undesirable that the distance between `BBBBAB` and `CCAAAC` (which are very different but share an `A` spell in the middle) is greater than that between `BBBBBB` and `CCAAAC`.[2]

Time-warping

Marteau proposes a modified time-warping distance measure which he calls the time-warp edit distance (TWED) (Marteau, 2008, 2007). This is quite similar to OM in its operation, as it uses a substitution matrix, and has operations analagous to substitution, insertion and deletion (though the latter two are better thought of as compression and expansion, or even better as compress-A and compress-B). It has a stiffness paramater ν and a gap-penalty, λ.

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). That is, conceptually time-warping uses continuous time, but it can be shown to work well in discrete time (Kruskal and Liberman, 1983). Marteau shows that there is a low bound to the discrepancy caused by such discretisation for this measure. While TWED can accommodate any sort of state space, and is usually described in terms of $\mathbb{R}^n$, a space composed of many real dimensions, where distances between points can be calculated in Euclidean or other terms, there is no difficulty in mapping to a discrete state space where distances between states can be given in a table (or "substitution matrix"). TWED is designed to accommodate irregular time-sampling, but is a little simpler to program when we have fixed time steps, as is the case considered here, and as is typically the case with lifecourse data.. TWED differs from other time-warping measures in that it generates metric distances between sequences.

It differs strongly from OM in that the operations (i) consider consecutive pairs of tokens in all three operations (ii) has a stiffness parameter that cumulates each time a comparison is made where time is realigned, and (iii) doesn't edit the content or order of the sequence (insert or delete) but aligns by altering the time dimension

The compress operations are costed at $d(s_{i-1}, s_i) + \nu + \lambda$. That is, compressing at point i depends on the similarity of s_i to s_{i-1}, plus the stiffness parameter (ν) and the

[2] Attempts have been made to remedy this by scaling distances according to spell length, such that sequences with long spells have their distances increased. So far this has been without success. As for LOM, the OMv variation applies only for certain pairs of sequences and is not global.

Table 1: Three example sequences, with their duration-weighted subsequences

S_1 = (A/10, B/4, C/6)	S_2 = (A/10, B/7, C/3)	S_3 = (B/9, A/5, B/6)
ABC / 20	ABC / 20	BAB / 20
AB / 14	AB / 17	BA / 14
AC / 16	AC / 13	BB / 15
BC / 10	BC / 10	AB / 11
A / 10	A / 10	B / 9 (+ 6)
B / 4	B / 7	A / 5
C / 6	C / 3	(B / 6)
SXX: $\sum t_{1i}^2 = 1104$	$\sum t_{2i}^2 = 1116$	$\sum t_{2i}^2 = 1192$

gap penalty λ.[3] This operation is considered as time-warping, stretching or compressing, depending on which sequence is being considered. Matching is TWED's equivalent of substitution: when we stop deleting or warping time, we consider the difference between the now-aligned tokens as a matching cost (with exactly the same effect as substitution). However, the comparison is between consecutive pairs of tokens, and has a stiffness penalty of $2\nu(|i-j|)$, i.e., twice the stiffness parameter times the time dislocation between the two sequences.

TWED offers an alternative to OM that is very similar in terms of implementation, but quite different in its motivation. By virtue of its stretching and compressing operations, and its attention to successive pairs of tokens, it is likely to respect the spell structure of the trajectory better than OM. In this respect, and since it generates metric distances, it may well achieve what LOM and OMv attempted.

Duration weighted subsequence counting: Elzinga's X/t

Elzinga (2005, 2003, 2006) proposes a set of measures for comparing sequences based on enumerating common subsequences (where, for instance, `AC` is a subsequence of `ABC`)[4]. Within this framework, time can be taken account of in two broad ways. We can repeat elements according to the number of time-units a spell lasts (as is done in all the other techniques used here; we can refer to these as calendar sequences, with one element per time unit) or we can record sequences as lists of spells which have both a state and a duration (referred to as spell sequences, or X/t for short). He discusses a number of ways of generating a similarity measure: calculating the longest common prefix, the longest common subsequence, the number of distinct common subsequences, and the count of common subsequences, *inter alia*. He also discusses a number of ways of incorporating the duration data. The implementation of his X/t approach used here counts all common subsequences between pairs of spell sequences, weighting by the cumulative duration of the subsequence (in Elzinga (2006) he proposes weighting by the sum of the product of the durations of the subsequences; this approach is computationally a little simpler to implement).

Table 1 shows three example sequences with their subsequences. All three have the same number of elements, and thus the same number of subsequences. However, since S_3

[3] More generally, ν should be multiplied by the time difference between $t_{\lfloor i-1 \rfloor}$ and t_i, but that is always 1 in the sort of data we are using.

[4] More recent work, as described in Elzinga and Wang (2012), seems to generalise this approach in a number of respects, not least of which is the introduction of variable distance between states.

Table 2: Enumerating common tuples, S_1 and S_2

S_1	s_3	Product of duration	
ABC / 20	ABC / 20	20×20	400
AB / 14	AB / 17	14×17	238
AC / 16	AC / 13	16×13	208
BC / 10	BC / 10	10×10	100
A / 10	A / 10	10×10	100
B / 4	B / 7	4×7	28
C / 6	C / 3	6×3	18
SXY_{12}	$= \sum t_{1i}t_{2i}$		1092

Table 3: Enumerating common tuples, S_1 and S_3

S_1	S_2	Product of duration	
ABC / 20	—	—	0
—	BAB / 20	—	0
AB / 14	AB / 11	11×14	154
AC / 16	—	—	0
—	BA / 14	—	0
BC / 10	—	—	0
—	BB / 15	—	0
A / 10	A / 5	10×5	50
B / 4	B / 9	4×9	36
B / 4	B / 6	4×6	24
C / 6	—	—	0
SXY_{13}	$= \sum t_{1i}t_{3i}$		264

has a repeated element, it has a smaller number of *distinct* subsequences (the subsequence `B` appears twice, with a total duration of 15). The SXX measure calculated in the final row is the sum of the square of the cumulated duration in each distinct subsequence (so `B` in S_3 yields 15×15 rather than $9 \times 9 + 6 \times 6$).

The distance measure is defined as:

$$d_{X/t} = \sqrt{SXX + SYY - 2 \times SXY}$$

where SXY is the sum of the product of the cumulated duration of each subsequence shared between sequences X and Y. SXX and SYY represent the same measure for X compared with X and Y with Y, respectively, that is, the sum of the square of the cumulated duration of each subsequence. An alternative would be to weight shared subsequences according to the sum of the product of the time in each state – this will yield greater differences between sequences with similar spell order but different durations. For instance, in the example above, S1 and S3 share an `ABC` subsequence, and this is weighted at $20^2 = 400$, the same as the subsequences' contributions to SXX and SYY, rather than $10 \times 10 + 4 \times 7 + 6 \times 3 = 146$ (compared to 152 to SXX and 158 to SYY). However, since the differences in the state-specific durations will feature in the shorter subsequences (`AB`, `AC`, `BC`, `A`, `B` and `C`) this does not compromise the measures' ability to distinguish between similar sequences. Indeed, it is possible to argue that the other approach is deficient in

Table 4: Calculating the distances from the sums of products of duration

	SXY				Distance		
	S_1	S_2	S_3		S_1	S_2	S_3
S_1	1104	1092	264	S_1	0	6.0	42.0
S_2	1092	1116	342	S_2	6.0	0	40.3
S_3	264	342	1192	S_3	42.0	40.3	0

multiply counting the differences. The primary reason for using the subsequence cumulated duration is computational convenience: it requires storing a single datum per subsequence, rather than a vector as long as the number of elements. Where sequences are long, the number of subsequences can be extremely large.

Elzinga has proposed an efficient algorithm for enumerating subsequences common to a pair of sequences. My approach is slightly different: I enumerate all subsequences of all sequences in a first pass, and then do a pairwise identification of common subsequences for all pairs of sequences. Since the enumeration of subsequences happens only once per sequence, this is also efficient (at least in processing terms: it requires ample memory). For the sorts of sequences considered in this paper (that is to say, with a maximum number of spells rarely above 10 or 12), the X/t measure can be calculated quite quickly, but since the processing time and memory requirements of the enumeration pass are approximately $O(2^l)$[5] there are sharp limits as the number of spells rises.

Elzinga has implemented many of his proposed measures in his own software, CHESA. A number are also implemented in the R package for sequence analysis, TraMineR (Gabadinho et al., 2011), and this X/t implementation will shortly be available in SADI.

Dynamic Hamming

The final measure is Lesnard's dynamic Hamming distance (Lesnard, 2006; Lesnard and de Saint Pol, 2009; Lesnard, 2010). This starts from the insight that transition rates can be used to calculate inter-state distances, but goes further in recognising that a significant part of the longitudinal structure of sequences lies in the changing transition matrix over time. It therefore uses transition-derived inter-state distances in an element-wise (Hamming-style) sequence comparison, where the distances change over time. It has a big advantage over strict Hamming in that the state-distances evolve dynamically. In particular, when transitions are more common, states are more similar, which means that differences in the timing of a transition are discounted where transitions are common and accentuated where they are rare. Correspondingly, because it does not allow alignment in the OM fashion, it respects the time dimension more: it identifies similarity at the same time, while being less sensitive to difference at busy times. It is most appropriate where there is a clear "calendar", such as a 24-hour day, seven-day week, or seasons of a year – where everyone experiences the same "clock" though they may respond to it differently. Where time is less constrained, or is "developmental" (where people go through the same or similar sequences of states, but at different speeds) it is less appropriate. Its greatest use has been in analysis of time-diary data.

Lesnard has written software to implement dynamic Hamming, available for Stata and SAS at http://laurent.lesnard.free.fr/. It is also available in the R package for se-

[5] That is, are roughly proportional to 2^l, where l is the number of spells.

quence analysis, TraMineR (Gabadinho et al., 2011), and will shortly be available in SADI.

2 Sequence analysis of simulated data

This section compares the performance of the measures with simulated data, under a number of scenarios. In each scenario there are two processes generating the sequences, which differ in simple ways. We test the measures in terms of how sensitive they are to the difference, when cluster analysis is carried out the resulting pairwise differences. The point is not necessarily to recover the distinction in the cluster analysis, but to see how the seven measures compare in their sensitivity to the six different contrasts.

2.1 Multiple simulation scenarios

The simulation regimes use a 3-state space over 40 time points. The six patterns are as follows:

- the subsets have different transition matrices
- the subsets have different baseline transition probabilities, but the same pattern
- one subset has a forced transition to a given state at a random time in mid-sequence
- one subset has a change in the baseline transition probability, at a random time in mid-sequence
- the two subsets have transition rates that are weighted averages of two regimes, where the weight oscillates over time at different rates
- again two different weighted averages of two transition regimes, but where the weight oscillates over time at the same rate but out of synch.

These patterns are intended to approximate the sorts of differences between lifecourse sequences that analysts might be interested in picking up.

2.2 The basic test

The simulation test consists of generating 1,000 40-unit-long sequences, and generating pairwise distance matrices for the seven distance measures. This is done for five transition-rate thresholds, generating sequences with lower and higher numbers of spells. Basic characteristics of the simulations are presented in Appendix Table 7. Cluster analyses are run on the distance matrices, generating solutions of 2, 4, 8, 16 and 32 groups. The association between the cluster grouping and the binary simulation type (each simulation scenario has two types of mechanism) is then subjected to a χ^2 test. While it is not thought that any distance measure should unambiguously recover the mechanism type (not least because any given sequence could in most cases be generated by either mechanism, though the likelihood will be higher under one than another), a higher χ^2 is evidence that cluster analysis based on this distance measure is better at responding to the input information.

The basic run of five threshold values by five cluster group sizes by six simulation regimes by seven distance measures is repeated 200 times, yielding 210,000 distinct results. Even with the relatively fast Stata plugins, this takes a substantial amount of time

(of the order of two days per simulation regime). Two hundred runs may be more than is strictly necessary, but it gives confidence that the results are consistent.

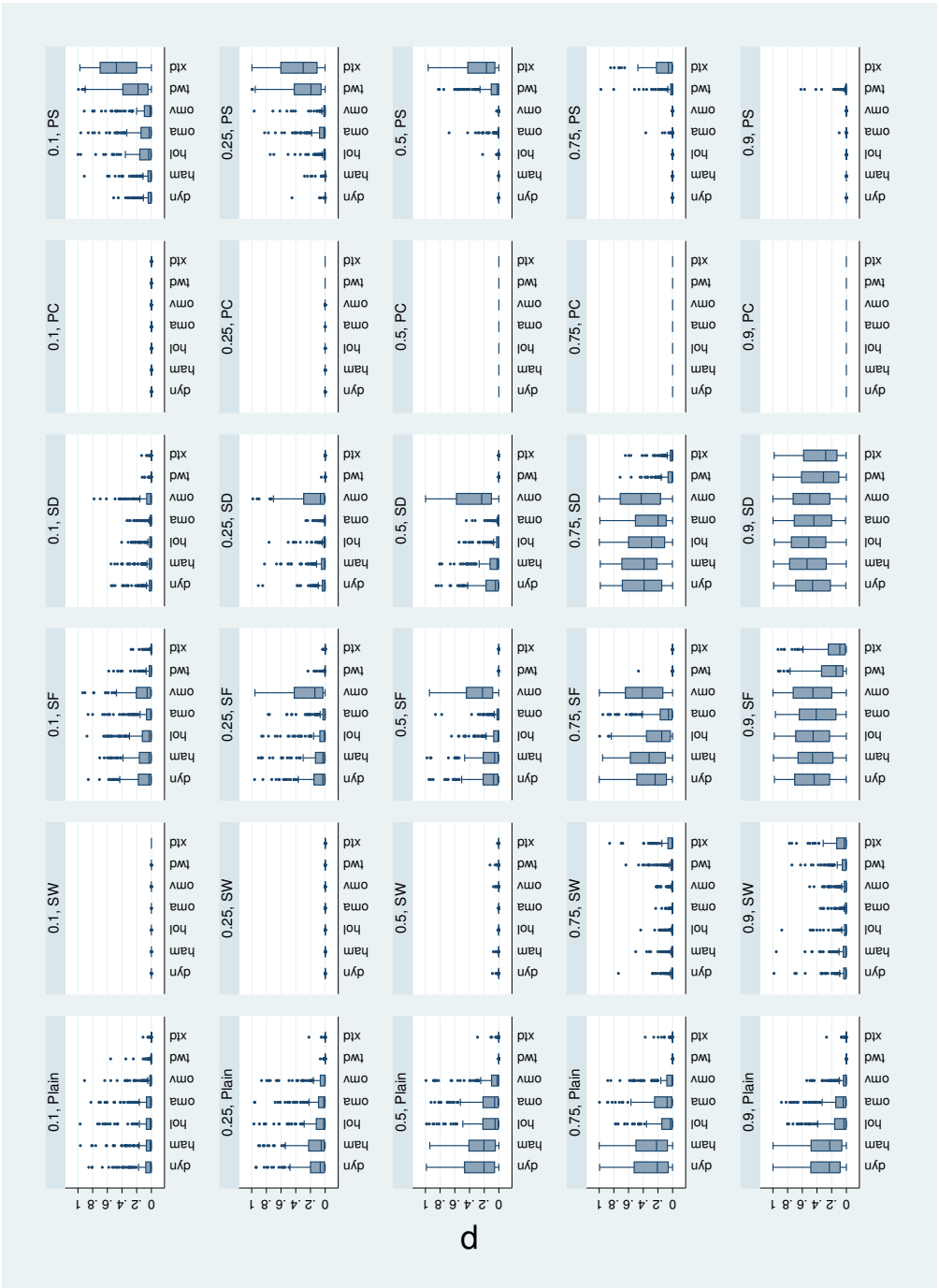

Figure 1: Right-tail χ^2 area, simulation results, 16 cluster solution, by simulation regime (columns) and transition parameter (rows)

We are effectively judging how well the distance measures can detect the "signal" that is the difference programmed into each simulation regime. As noted we do not expect to be able to reproduce the difference, but we want to see how much association there is between it and the cluster solutions. The distance measures are compared with each other, rather than against the task of recovering the difference. How they fare across the regimes, with different transition rates and different numbers of groups in the cluster solution, is informative. We judge this performance in terms of the χ^2 statistic (both its value and the area under the right tail; this latter is not to considered a p-value in the conventional sense but rather a metric for comparison). Figure 1 summarises the distribution of the area under the right tail, across five transition rates and six simulation regimes, for the 16-cluster solution (see Figure 2 to 6 for all five cluster solutions, and Figure 7 to 12 for the same information organised by simulation rather than cluster solution size).

Clearly, the number of groups in the cluster solution will have a big impact in the ability of the process to "detect" the signal. If the simulation regime creates two very distinct types of sequences, two groups may well pick up the difference well, but it is likely that most simulation regimes will create sequences that overlap substantially in characteristics. Insofar as that is true, the few-groups clustering may be responding to characteristics that are not diagnostic of the hidden difference (e.g., such as the average distribution of state), and in that case discrimination may not emerge until we consider more disaggregated groupings. It may also be the case that too-disaggregated solutions lose their discriminating power, as the clusters become small and idiosyncratic. In this respect Figure 1 represents a fairly high number of clusters (given we have a three-state space, 40 units and between approximately two and eight spells on average) – we see that for many runs, for most of the distance measures and simulation regimes, the association between the cluster solution and the hidden difference is large. Consider the "plain" simulation first (column 1). For the threshold values of 0.1 and 0.25 (corresponding to less than two and about three spells on average, respectively, typical of a lot of lifecourse sequence data – see Table 7 for details), most measures provide low right-tail values. However, it is clear that X/t, TWED and, to a lesser extent, OMv, are doing somewhat better than the others. This is also true of the higher transition rates too, but there the other measures perform less well, particularly the Hamming and Dynamic Hamming. The plain simulation regime is carefully set up to differ only in the transition pattern: average time spent in each state and average transition rate is the same, but while Type 1 has unpatterned transitions, Type 2 favours A to B over A to C, B to C over A, C to A over B. Where transition rates get high, sequences from of the same type will have no particular tendency to be in the same state at the same time, whereas when transitions are rare, the coincidence of having a transition from, say, state A, at approximately the same time, will make two sequences more similar. Thus the Hamming measures will have some success at low rates of transition but less at high. However, since the type difference here is strongly related to spell sequence (i.e., ABC versus ABA) the measures that pay attention to spell order have more discrimination.

The second simulation regime, the "switch" simulation, has a single base transition rate, but type 2 experience a forced transition to state 3 at an individual-specific point at random between period 15 and period 35. Thus is reflects an "event" that happens to a subset of cases, rather than persistent difference. For all measures, this is an easier distinction to detect at the 16-group cluster solution, more so at lower base transition rates. At higher transition rates TWED and X/t seem to do slightly worse, a pattern which is repeated if we look at cluster solutions with fewer groups: if anything, there OMA has the advantage, though it is not marked. Since the design feature of the regime is

being in a given state at a given (approximate) point, it makes sense that at low transition rates this is easier to detect than at high (where it is easier for sequences to exit the state). For the same reason one would expect that measures closer to Hamming will do well, with the ability to align (to detect similarity at approximately the same place) also quite important. X/t does relatively poorly here because it is less attached to location and more to sequence than other measures. While TWED is an aligning measure, however, with a cost to recognising similarity out of alignment, it seems to be closer to X/t in this case.

The "slow–fast" regime types differ in the rate of transition, but the pattern is the same and the rates are steady through time. Figure 1 shows that X/t, TWED and to a lesser extent OMA are the strongest here (all measures lose out at the highest transition rate, but this may be due to distinction between the faster transition regime and the slower declining as the base rate rises). The Hamming measures and the adapted OM measures do less well. The difference between the sequences should largely take the form of shorter spells and more frequent transitions in type 2 – X/t's focus on spells will be strongly affected by number of spells, TWED will also pick up common spell patterns.

The "slow-down" regime is similar. Here type 2 switches from accelerated transitions to a below-average rate at a random time in mid-sequence. Rather than a many-spells/few-spells contrast between types as in the previous regime, we have short-then-long versus a stable transition pattern. At 16 clusters, and low to medium base transition rates, all measures (except OMv) pick up this feature fairly well, but X/t and TWED do better (especially at the 0.75 transition threshold). Again, the defining feature of the regime being focused on spell patterns rather than states, the spell-oriented measures do well.

The "plain-cycle" and "plain-step" regimes are designed to test the dynamic Hamming measure, and have types with transition rates that change in different ways over time. "Plain-cycle" has two underlying transition matrices, and each type uses a time-dependent weighted average of them. Type 1 goes from 100% matrix A at t_0 to 100% matrix B at t_{20} and back to matrix A by the end. Type 2 goes through this cycle twice, at twice the speed. "Plain-step" has both types going through the cycle twice, but out of sync, as if in a time-diary data set one subsample is on an 8–4 cycle and the other 9–5.

The "plain-cycle" difference is very easily picked up by all measures, for all cluster solution sizes and at all base transition rates. This is probably due to a systematic difference in the distribution across the three states of the two types. While Figure 1 is not very informative, viewing the raw figures for the mean χ^2 suggests that X/t and TWED are systematically (but not very far) ahead of the other measures. X/t's bad showing in the 0.1-threshold, 2-cluster plot (Figure 2) reflects an odd distribution, with some very low χ^2 values but a high mean.)

The "plain-step" is clearly in the Dynamic Hamming spirit, and indeed the two Hamming measures do best here (but the dynamic version not clearly better than the basic one). X/t and TWED do distinctly badly however: because the two types are only a little out of sync, they do not tend to have different spell patterns.

Winners and losers

We can get an overall view of how the seven measures fare across the six simulation regimes in a number of ways. A simple one is to identify for each run, for each cluster solution, which measure does best and which worse. A more complicated way is to use a regression model to estimate simulation/measure specific averages. The results are fairly consistent. From Table 5 we see that for the "plain" regime, TWED is by far the most common "best" measure while the two Hamming measures are worst. For the "SW"

Table 5: Winners and losers: best and worst measures by simulation regime (percentages)

Simulation regime		Distance measure						
		dyn	ham	hol	oma	omv	twd	xtd
Plain	best	2.2	2.3	3.3	3.04	4.96	62.2	22.24
	worst	26.96	25.62	13.72	15.52	9.46	3.74	5.2
SW	best	9.76	11.32	14.5	21.1	11.02	16.72	15.64
	worst	12.34	12.5	8.4	6.46	15.6	20.5	24.3
SF	best	2.72	2.86	3.34	5.12	2.76	22.68	60.54
	worst	17.1	18.56	13.48	9.7	34.6	4.8	1.86
SD	best	4.12	4.02	3.52	4.82	3.74	31.54	48.26
	worst	15.1	17.92	13.58	10.18	35.22	4.82	3.3
PC	best	.9	.92	2.42	1.22	1.08	47.1	46.4
	worst	17.56	17.64	7.32	10.26	42.16	2.36	2.82
PS	best	34.26	43.96	5.58	4.26	6.8	2.86	2.38
	worst	2.48	2.56	5.1	11.64	5.2	36.72	36.34

regime, OMA most commonly does best, while X/t does worst most often (however, the honours are widely distributed for this regime). For the "SF" or slow–fast simulation, X/t is a clear winner, being the best measure in 60% of the runs, with TWED a distant second. For the "SD" or slow-down simulation, X/t wins nearly half the time, with TWED not too far behind. There is a similar pattern for the "PC" simulation, with TWED slightly ahead of X/t, and the other measures a long way behind (note that for this simulation, all the measures did absolutely well). The "PS" simulation, though structurally similar to "PC", yields the opposite result, with Hamming (and then dynamic Hamming) doing clearly best, with TWED and X/t doing worst.

While counting winners and losers gives a clear picture, it throws away some information. If we look at the effect of measure on the average χ^2 or on the average $\log \chi^2$, we retain much of this information. We can do this in the framework of a regression model, with terms for the interaction of cluster-solution size and transition threshold (as factors) and for the interaction between simulation regime and distance measure. In Stata terms we fit the following model:

```
xtreg chi i.sim##i.measure i.ngroups##i.threshold, i(id)
```

The `id` variable identifies cluster solutions within simulation runs. The `xtreg` random effects model thus generates properly conservative standard errors; however, the parameter estimates are the same (or almost so) as those from a conventional regression model. Table 6 summarises these models, by reporting the net effect of measure, separately for each simulation, setting the effect of TWED to zero (these effects simply combine the parameter estimates for measure and the measure/simulation interaction).

From the regression models we get almost exactly the same picture as from the winner/loser analysis:

Table 6: Modelling χ^2 and $\log \chi^2$ by simulation regime and distance measure: average differences to TWED by simulation regime

Simulation regime	Additive effect on χ^2 Distance measure						
	dyn	ham	hol	oma	omv	twd	xtd
Plain	-56.363	-56.327	-52.439	-53.250	-49.451	0.000	-34.124
SW	-0.329	0.867	2.474	2.571	-1.074	0.000	-1.350
SF	-15.803	-15.728	-14.179	-12.055	-17.840	0.000	10.362
SD	-18.856	-19.005	-17.079	-15.847	-21.533	0.000	4.245
PC	-88.214	-87.501	-59.669	-66.936	-96.961	0.000	17.779
PS	75.676	81.090	30.208	13.067	28.438	0.000	13.115
Simulation regime	Additive effect on $\log \chi^2$ Distance measure						
	dyn	ham	hol	oma	omv	twd	xtd
Plain	-1.525	-1.515	-1.312	-1.355	-1.168	0.000	-0.771
SW	0.163	0.190	0.287	0.496	0.237	0.000	-0.042
SF	-0.793	-0.818	-0.706	-0.548	-0.939	0.000	0.469
SD	-0.764	-0.764	-0.674	-0.562	-0.915	0.000	0.308
PC	-0.227	-0.228	-0.153	-0.163	-0.249	0.000	0.016
PS	1.474	1.544	0.771	0.365	0.767	0.000	-0.135

Simulation	Winner	Average χ^2	Average $\log \chi^2$
Plain	TWED	TWED	TWED
SW	OMA	OMA	OMA
SF	X/t	X/t	X/t
SD	X/t	X/t	X/t
PC	TWED	X/t	X/t
PS	Hamming	Hamming	Hamming

The only difference is for the "PC" simulation where in the regression models X/t does better than TWED, but we see that the difference is small.

Overall the exercise suggests that no one measure is dominant. Hamming, OMA, TWED and X/t all do well, with dynamic Hamming often close to Hamming. Hollister's LOM is generally in the middle, often not too far from OMA. Halpin's OMv is also often close to OMA but is also often the weakest measure.

A Appendix

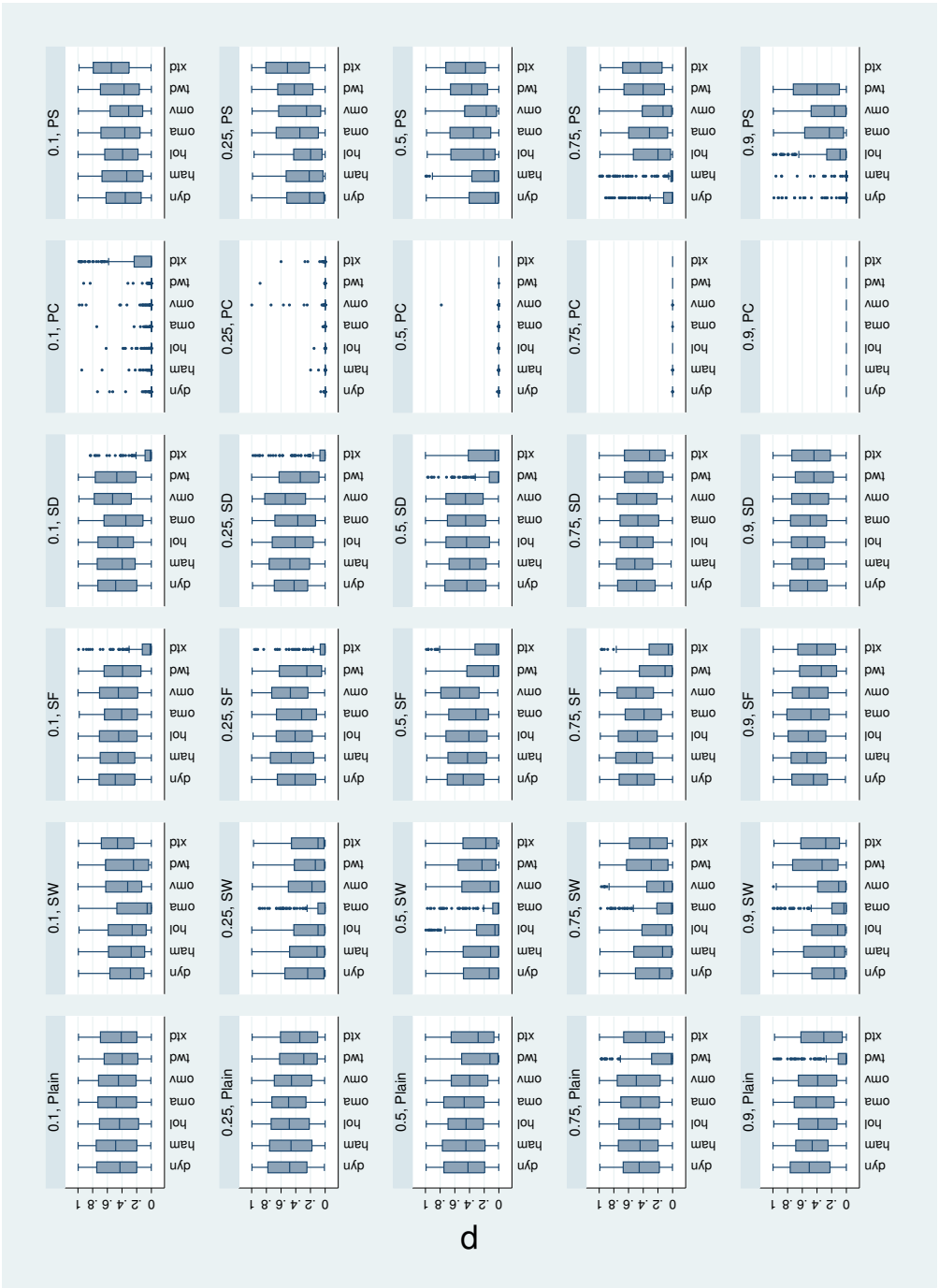

Figure 2: Right-tail χ^2 area, simulation results, 2 cluster solution, by simulation regime (columns) and transition parameter (rows)

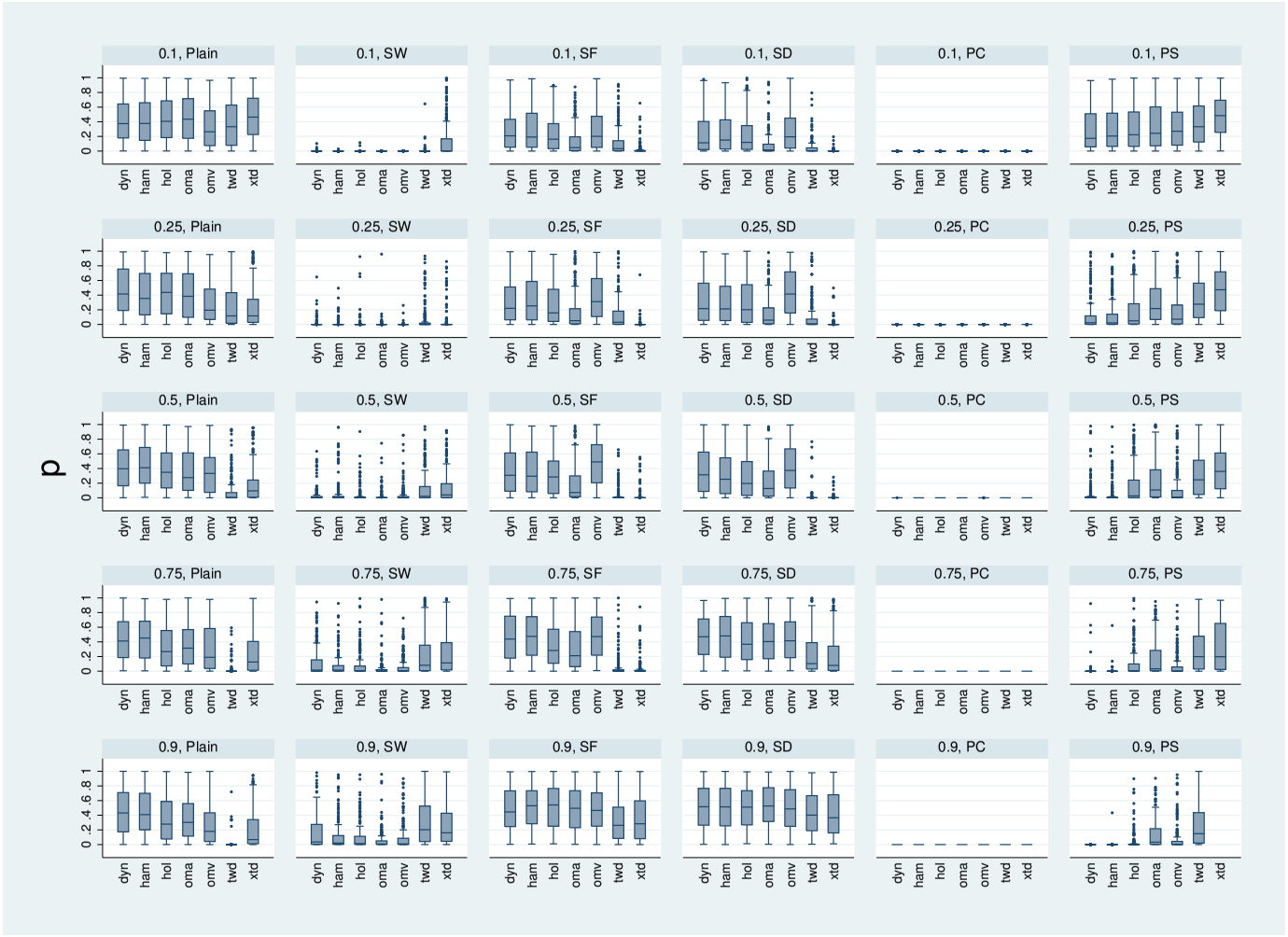

Figure 3: Right-tail χ^2 area, simulation results, 4 cluster solution, by simulation regime (columns) and transition parameter (rows)

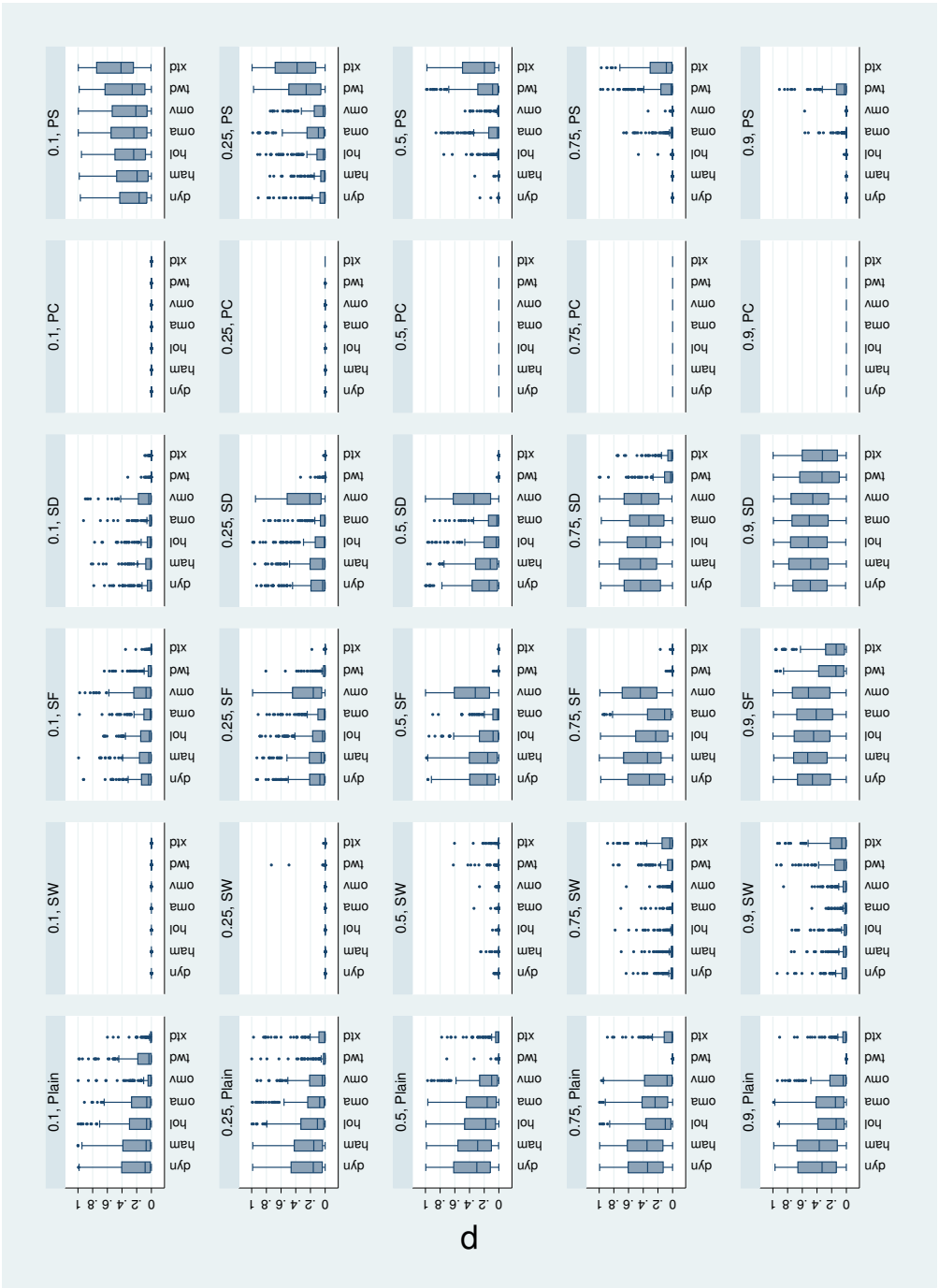

Figure 4: Right-tail χ^2 area, simulation results, 8 cluster solution, by simulation regime (columns) and transition parameter (rows)

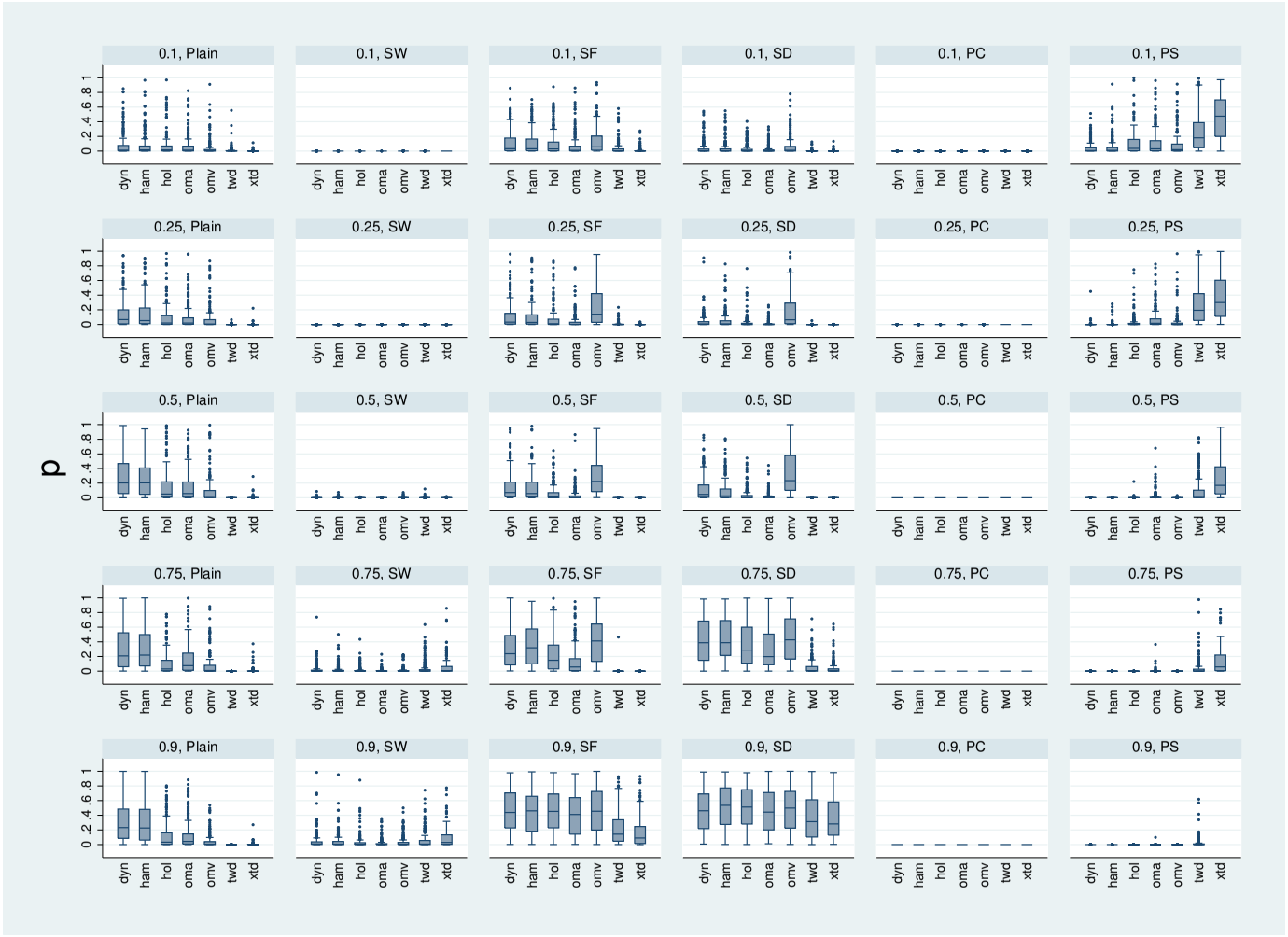

Figure 5: Right-tail χ^2 area, simulation results, 16 cluster solution, by simulation regime (columns) and transition parameter (rows)

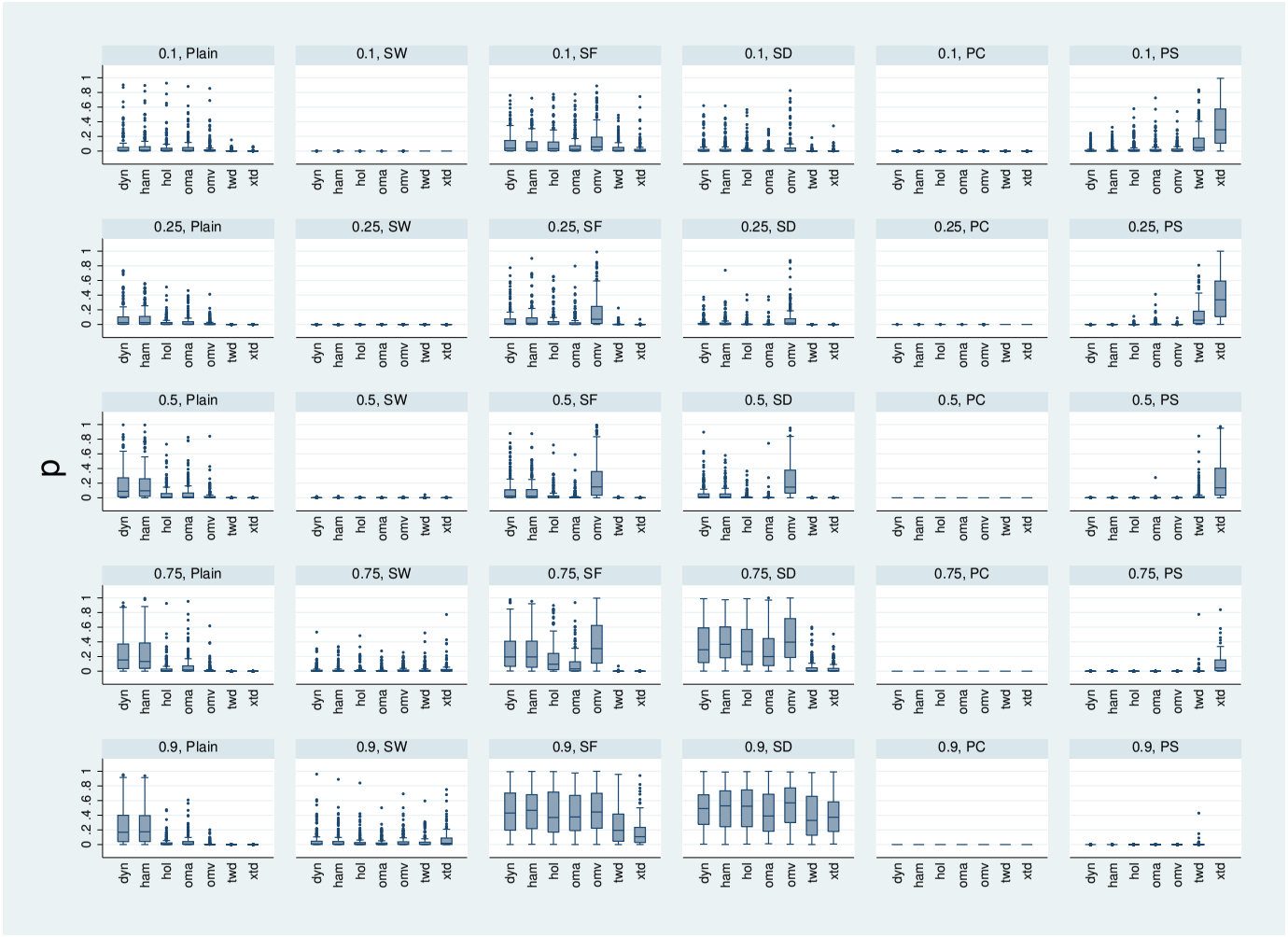

Figure 6: Right-tail χ^2 area, simulation results, 32 cluster solution, by simulation regime (columns) and transition parameter (rows)

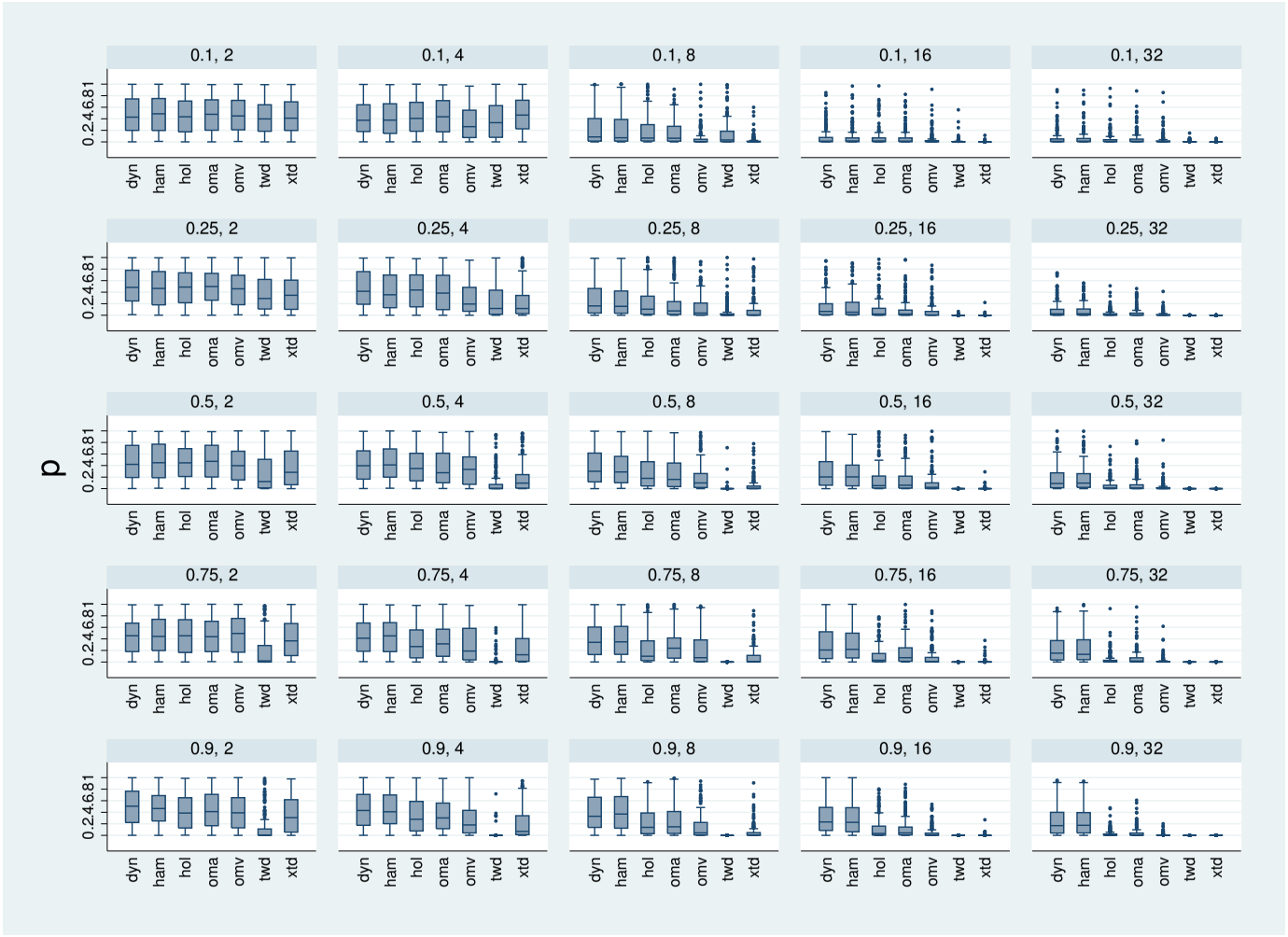

Figure 7: Right-tail χ^2 area, simulation results, "plain" simulation regime, by cluster solution size (columns) and transition parameter (rows)

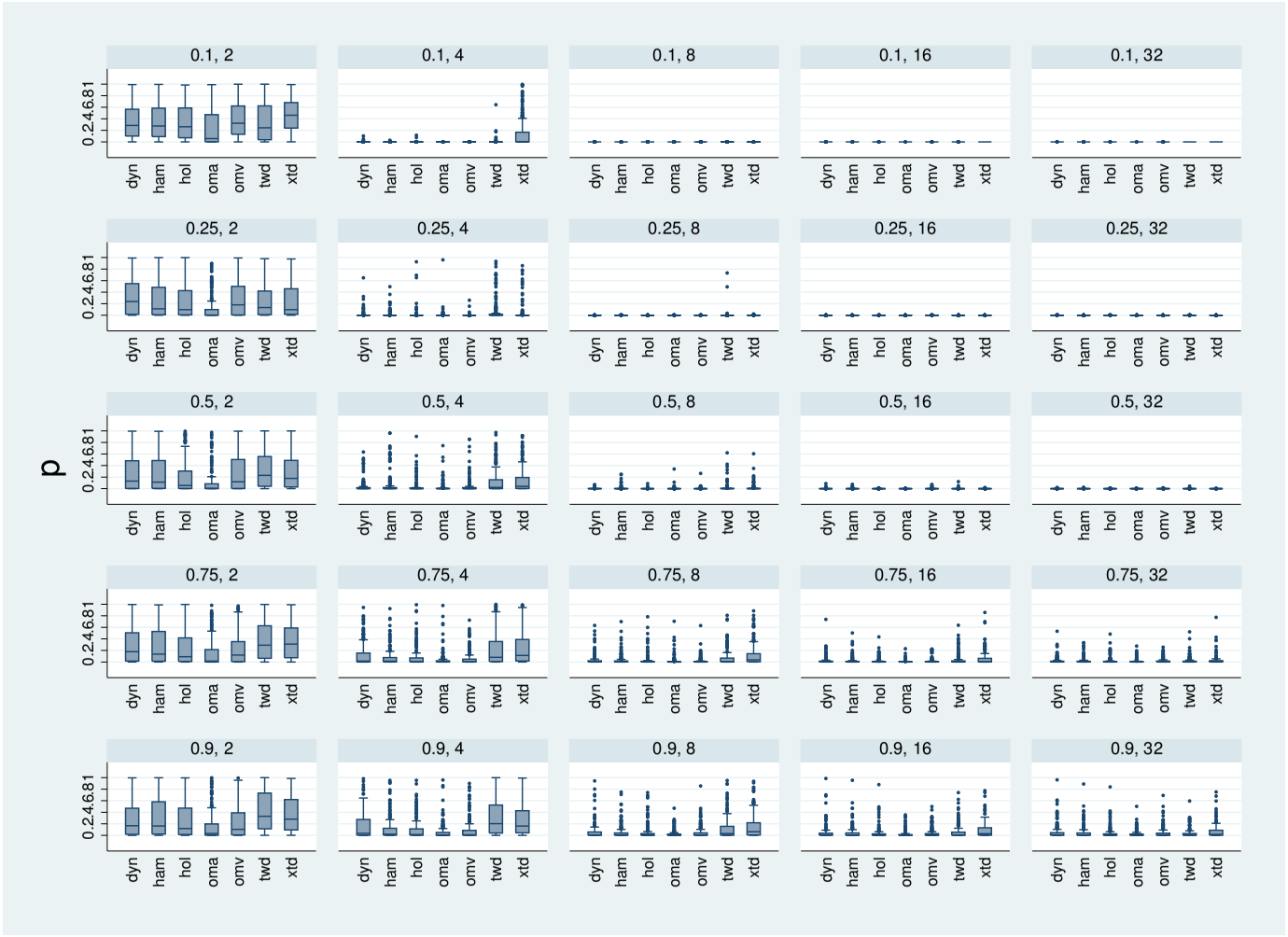

Figure 8: Right-tail χ^2 area, simulation results, "SW" simulation regime, by cluster solution size (columns) and transition parameter (rows)

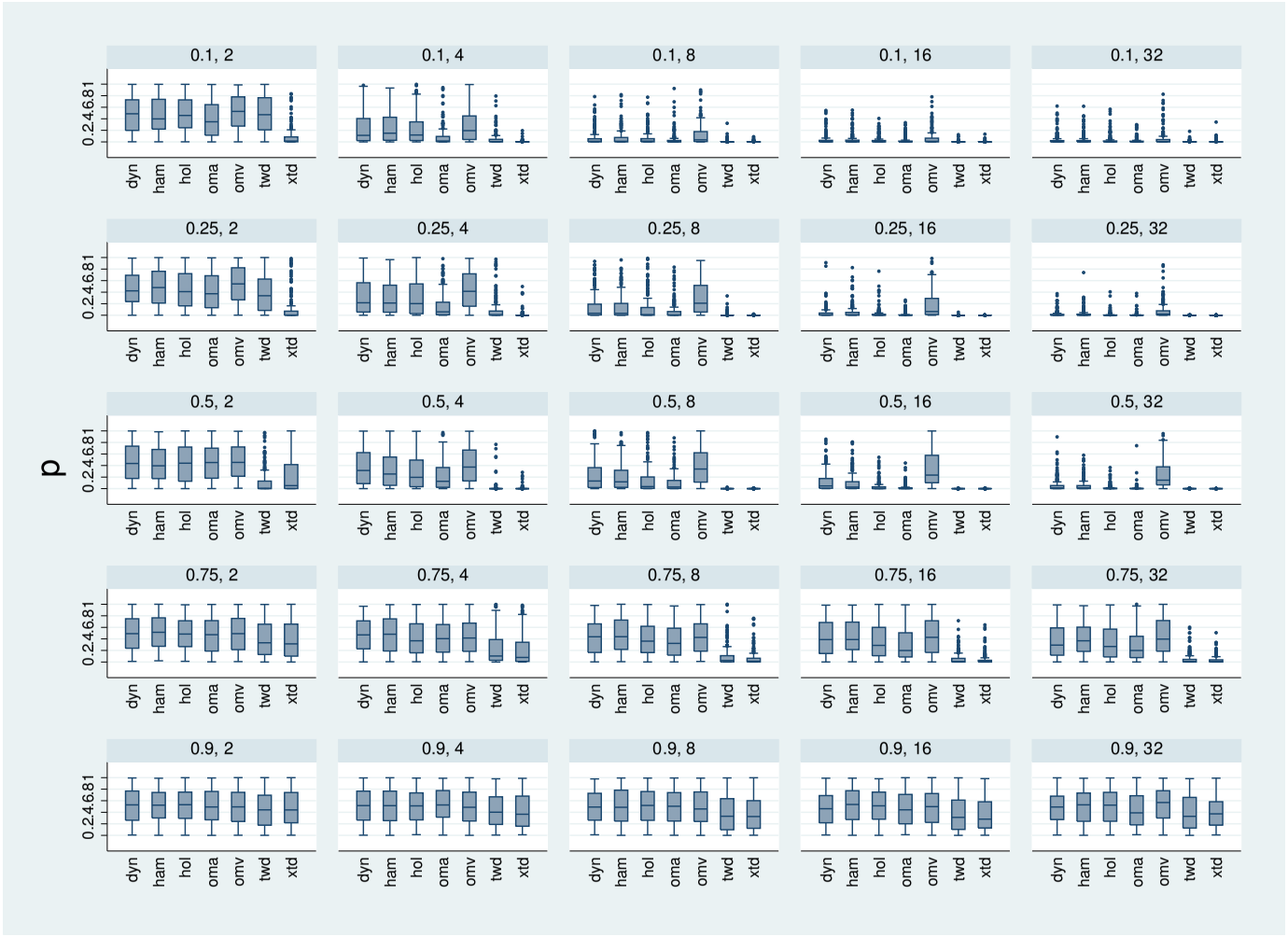

Figure 9: Right-tail χ^2 area, simulation results, "SD" simulation regime, by cluster solution size (columns) and transition parameter (rows)

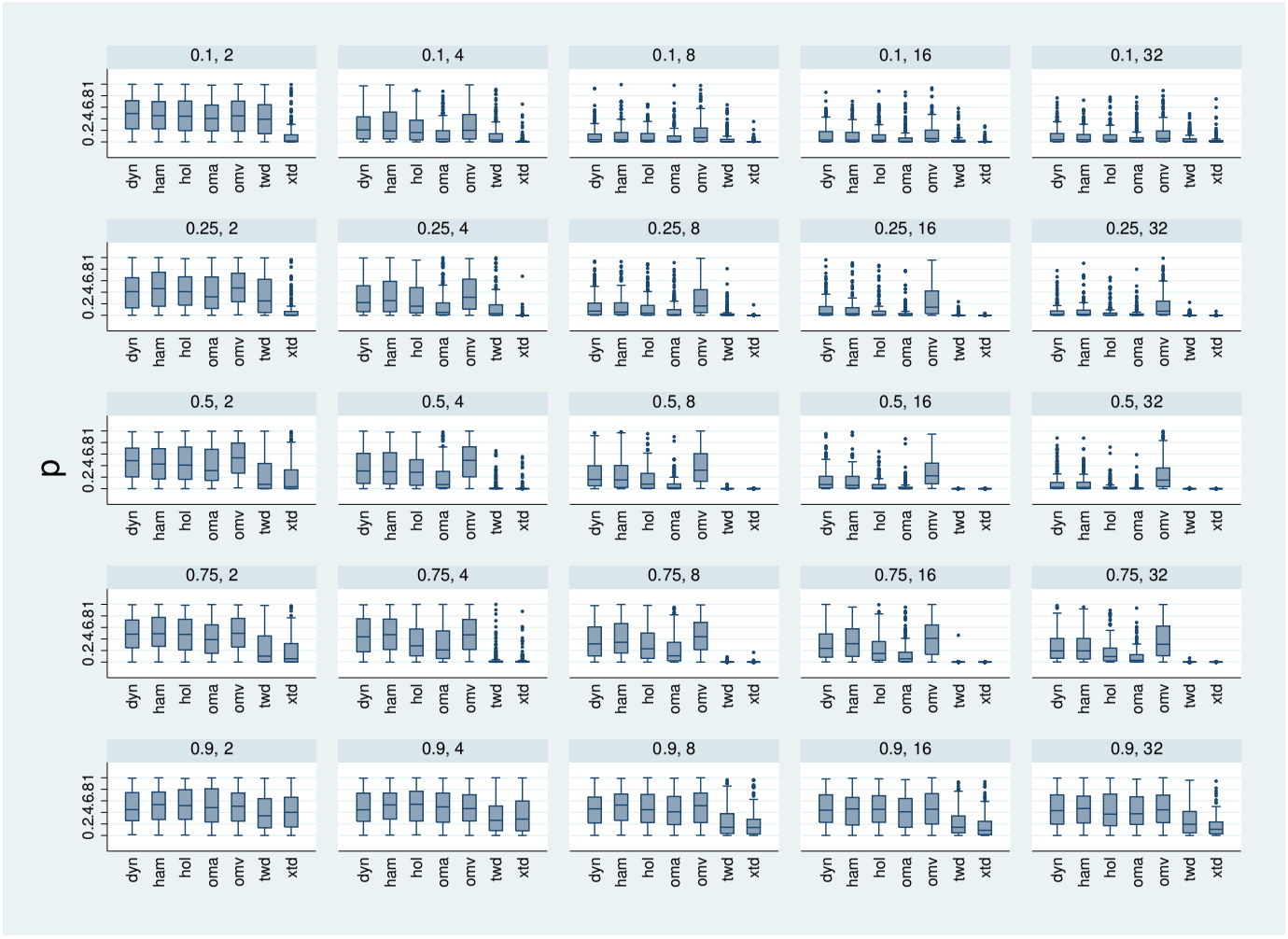

Figure 10: Right-tail χ^2 area, simulation results, "SF" simulation regime, by cluster solution size (columns) and transition parameter (rows)

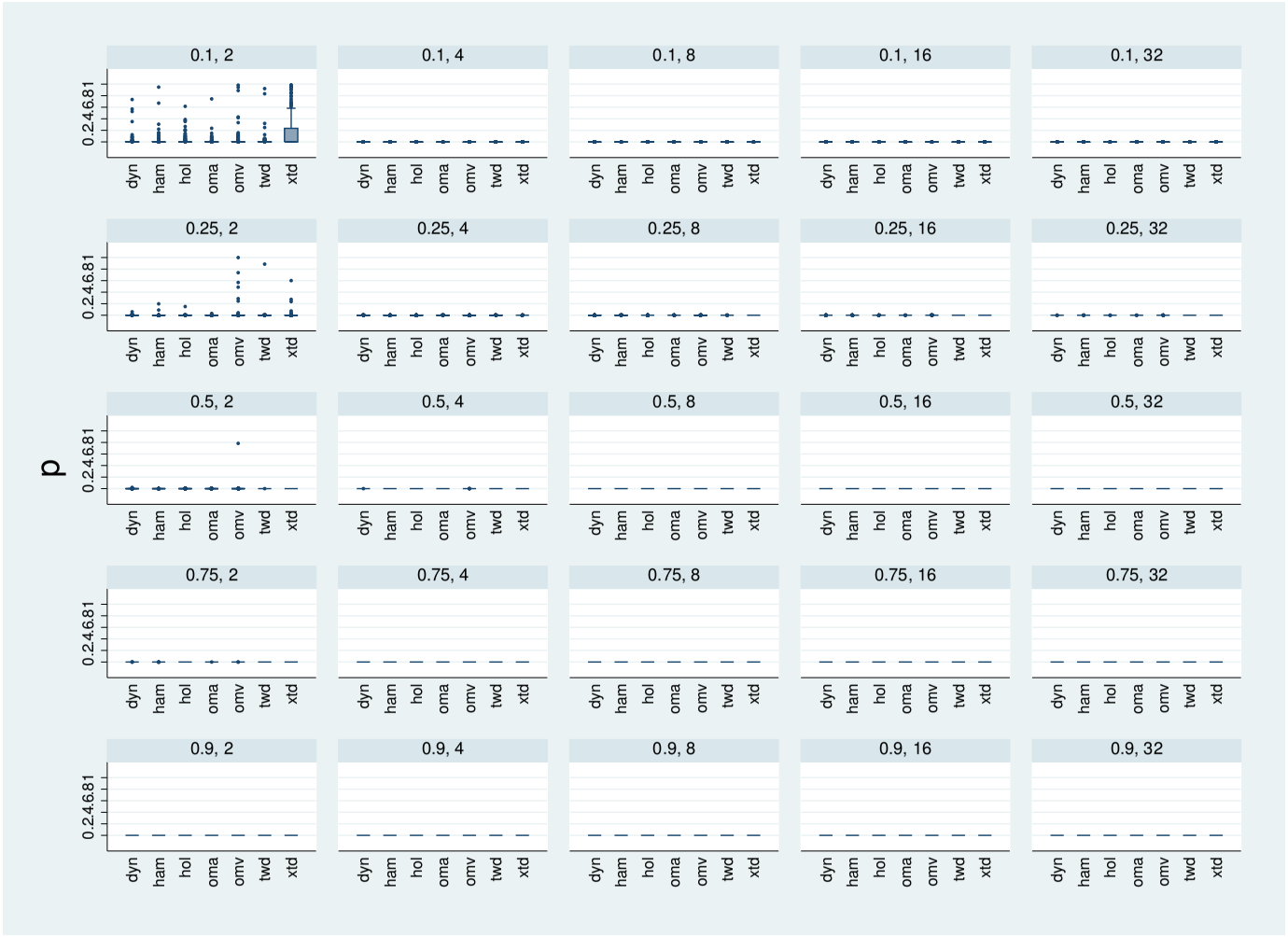

Figure 11: Right-tail χ^2 area, simulation results, "PC" simulation regime, by cluster solution size (columns) and transition parameter (rows)

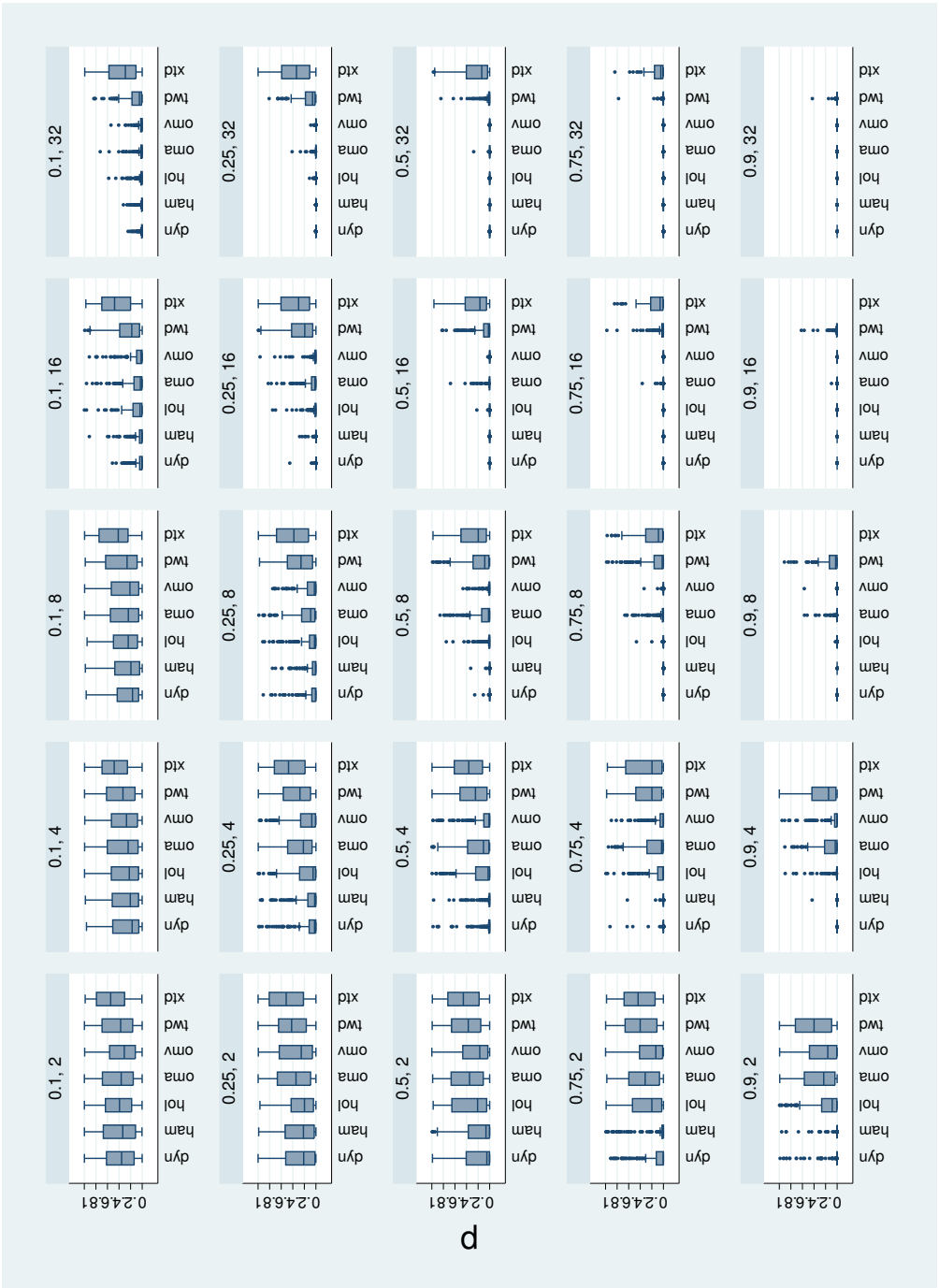

Figure 12: Right-tail χ^2 area, simulation results, "PS" simulation regime, by cluster solution size (columns) and transition parameter (rows)

Bibliography

Abbott, A. (1995). A comment on "Measuring the agreement between sequences". *Sociological Methods and Research*, 24(2):232–243.

Abbott, A. and Forrest, J. (1986). Optimal matching methods for historical sequences. *Journal of Interdisciplinary History*, XVI(3):471–494.

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.

Abbott, A. and Tsay, A. (2000). Sequence analysis and optimal matching methods in sociology. *Sociological Methods and Research*, 29(1):3–33.

Chan, T. W. (1995). Optimal Matching Analysis: A methodological note on studying career mobility. *Work and Occupations*, 22:467–490.

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. (2006). Sequence analysis: Metric representations of categorical time series. Technical report, Amsterdam.

Elzinga, C. H. and Wang, H. (2012). Versatile string kernels. Paper presented at LaCOSA conference, Lausanne, June 6-8 2012.

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

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

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

Hollister, M. (2009). Is optimal matching suboptimal? *Sociological Methods and Research*, 38(2):235–264.

Kruskal, J. B. (1983). An overview of sequence comparison. In Sankoff and Kruskal (1983).

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 conomie et Statistique 2006-01, Institut Nationale de la Statistique et des tudes conomiques, Paris.

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.

Table 7: Summary characteristics of the simulations

Simulation type	Rate factor	N spells		Duration					
				State 1		State 2		State 3	
		T1	T2	T1	T2	T1	T2	T1	T2
plaincirc	.1	1.78	1.78	0.33	0.33	0.33	0.33	0.34	0.33
	.25	2.95	2.94	0.33	0.34	0.33	0.33	0.33	0.33
	.5	4.90	4.90	0.33	0.33	0.33	0.33	0.33	0.33
	.75	6.86	6.86	0.33	0.33	0.33	0.33	0.33	0.33
	.9	8.03	8.03	0.33	0.33	0.33	0.33	0.33	0.33
plaincycle	.1	1.56	2.20	0.37	0.29	0.20	0.42	0.42	0.29
	.25	2.21	3.93	0.38	0.27	0.13	0.45	0.49	0.27
	.5	3.18	6.82	0.38	0.27	0.09	0.47	0.53	0.27
	.75	4.07	9.64	0.37	0.26	0.07	0.47	0.56	0.26
	.9	4.59	11.31	0.36	0.26	0.07	0.48	0.57	0.26
plainstep	.1	2.19	2.19	0.29	0.29	0.41	0.42	0.30	0.29
	.25	3.92	3.95	0.28	0.27	0.44	0.45	0.28	0.27
	.5	6.63	6.74	0.27	0.27	0.46	0.46	0.27	0.27
	.75	9.23	9.39	0.27	0.27	0.46	0.46	0.27	0.27
	.9	10.80	10.95	0.27	0.27	0.47	0.46	0.27	0.27
plainswitc	.1	1.78	2.43	0.33	0.23	0.33	0.23	0.33	0.55
	.25	2.95	3.56	0.33	0.26	0.33	0.25	0.33	0.49
	.5	4.90	5.48	0.33	0.28	0.33	0.28	0.33	0.43
	.75	6.84	7.37	0.33	0.30	0.33	0.30	0.33	0.40
	.9	8.02	8.52	0.33	0.30	0.33	0.30	0.33	0.39
slowdown	.1	1.78	2.23	0.33	0.33	0.33	0.33	0.33	0.33
	.25	2.95	4.06	0.33	0.33	0.33	0.33	0.33	0.34
	.5	4.90	7.15	0.33	0.33	0.33	0.33	0.33	0.33
	.75	6.85	7.98	0.33	0.33	0.33	0.33	0.33	0.33
	.9	8.03	8.48	0.33	0.33	0.33	0.33	0.33	0.33
slowfast	.1	1.77	2.17	0.33	0.33	0.33	0.33	0.34	0.33
	.25	2.94	3.92	0.33	0.34	0.33	0.33	0.33	0.33
	.5	4.90	6.85	0.33	0.33	0.33	0.33	0.33	0.33
	.75	6.86	8.80	0.33	0.33	0.33	0.33	0.33	0.33
	.9	8.02	8.79	0.33	0.33	0.33	0.33	0.33	0.33

Levine, J. H. (2000). But what have you done for us lately? Commentary on Abbott and Tsay. *Sociological Methods and Research*, 29(1):34–40.

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.

McVicar, D. and Anyadike-Danes, M. (2010). Does optimal matching really give us anything extra for the analysis of careers: An application to British crime careers. *(under review)*.

Müller, N. S., Lespinats, S., Ritschard, G., Studer, M., and Gabadinho, A. (2008). Visualisation et classification des parcours de vie. *Revue des Nouvelles Technologies de l'Information*, II(E-11):499–510.

Piccarreta, R. and Lior, O. (2010). Exploring sequences: A graphical tool based on multidimensional scaling. *Journal of the Royal Statistical Society Series A*, 173(1):165–184.

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.

Pollock, G. and Roberts, K. (2012). Employment trajectories in the South Caucasus during the transition from communism to post-communism: A comparison of OMA and non-sequence based methods of producing typologies. Paper presented at LaCOSA conference, Lausanne, June 6-8 2012.

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

Scherer, S. (2001). Early career patterns: A comparison of Great Britain and West Germany. *European Sociological Review*, 17(2):119–144.

Wu, L. L. (2000). Some comments on "Sequence analysis and optimal matching methods in sociology: Review and prospect". *Sociological Methods and Research*, 29(1):41–64.