

Experiments with activity pattern classification: alignment versus non-alignment methods

Clarke Wilson

Paper presented at the
International Association for Time Use Research Annual Meeting
Washington, D.C.
October 2007

Abstract

The presentation will report on some experiments in the classification of Monte Carlo character sequences using dynamic programming methods and some alternatives, including time budgets. The test sequences are generated with a mixture of probabilistic patterns and noise and the presentation will report on the relative reliability of different classification strategies according to the difficulty of the classification problem. Synthetic character sequences are used because they can be created according to known rules and thus can identify the most successful among competing methods. As dynamic programming algorithms are not well known in social sciences, this set of experiments offers evidence of their reliability in the classification of sequences of events. The paper is mainly based on the report in Wilson (2006) with some extensions.

1. Introduction

This paper describes a number of tests of reliability of measures of character sequence similarity. The procedure is to generate a number of sets of synthetic sequences that display known patterns and compare the classifications produced by different analytical methods to see which retrieves the known patterns most accurately. The experiments include an alignment strategy, a measure based on Hamming similarity and a measure based on counts of characters. The latter is analogous to a time budget for the synthetic sequences. Success in finding known patterns is interpreted as indicating that a classification method is reliable

Any mathematical procedure will produce some result if data is supplied as required. Researchers using relatively new procedures have to assure themselves and the wider research community that the results are valid and reliable, and that such results are not generated by the methods themselves. Consequently, while interesting and suggestive results have been reported, it is important to address the question of reliability and validity of the methods.

Social researchers frequently face problems in measuring abstract concepts and there is a large literature that examines the validity and reliability of measurement methods. To quote from Carmines and Zeller (1979, p. 16), "Reliability is basically an empirical issue, focusing on the performance of empirical measures ... Validity ... is evidenced in the degree that a particular indicator measures what it is supposed to measure rather than reflecting some other phenomenon." Concerns regarding the use of alignment algorithms and of the accuracy of results based on alternative sets of parameter settings for alignment software are questions about the *reliability* of alignment methods. Discussions of the *validity* of alignment methods relate to the fidelity of, for example, diary data as a medium that conveys information about daily activity. Given that data on activity, migration, careers or whatever has been collected and has been recorded as character sequences, the validity bridge has been crossed. Researchers should rightly be concerned that alignment methods in general and parameter settings in particular identify similar individuals reliably. Examining this question is the objective here.

2. Sequence comparison methods

2.1 Sequence alignment

Pairwise alignments

Writing a pair of sequences, one above another, creates some degree of alignment between sequence elements. Finding an optimal alignment involves transforming elements of one sequence into elements of the other using a defined set of operations. Optimality of alignment means that no alternative arrangement of matching characters and inserted gaps can give a higher similarity (or lower distance) score than the one found using only eligible operations. Eligible operations in most pairwise alignment algorithms are identities (or exact matches), substitutions (or inexact matches), insertion of an element from one sequence into the other, and the converse operation of deletion of an element. Gaps are created in either sequence, as necessary, to

accommodate insertions and deletions. Insertions and deletions are descriptions of the same operation from the perspective of one or other sequence in the pair. They always occur in pairs and are usually called indels.

Alignment algorithms can be designed to calculate either distances between sequences or similarities. In this paper, the discussion will be mainly in terms of similarities but all comments apply equally to analyses based on distances. A matrix of similarities is computed for all pairs of members in the experimental sequences and these form the basis of the comparisons of classification effectiveness.

Indel and matching penalties

The experiments use a similarity value of 10 for matches, zero for mismatches, and separate opening and extension penalties for indels. Previous research indicates that lower indel costs give better results than higher costs so values of 1.0 for the opening penalty and 0.1 for the extension penalty are used.

Substitutions and partial matches

Concern has been expressed by some writers that alignment applications should incorporate carefully constructed and theoretically sound substitution premiums (or penalties). One strategy often suggested is to examine transition probabilities among events. The argument is that events with low transition probabilities are unrelated and their substitution should be penalized more severely than mismatches among events that are related in the form of event chains.

An experiment with a set of similarities derived from transition matrices is included. The similarity values employed are based on the ratios of the observed transition probabilities. A transition of a character to itself is a match and scores 10. The substitution score for the most frequent transition to another character is set to 6 and less frequent transitions are given proportionately lower scores. The actual transition rates did not vary much so the tabulated scores took values only from 4 to 6.

2.2 Hamming (Euclidean) sequence similarity

The Hamming similarity is simply a count of matches between characters at corresponding positions in the sequences \mathbf{a}_i and \mathbf{b}_j . Sequences must be the same length, a constraint not applicable to alignments. Where more than one dimension is involved and the indicators are measured by real numbers, a distance can be derived that has Euclidean properties, but this collapses to a count of matches or mis-matches (i.e. the Hamming measure) where one binary measure is available as in the case of character comparisons.

The Hamming measure can be interpreted as an extreme case of alignment in which all indels are suppressed by high penalties and substitutions are scored as zero, leaving only exact matches as positive scores. Tests of the alignment algorithm that set the gap penalty to 100 gave exactly the same scores as the Hamming measure. If sequences are very similar, we may expect to find that the Hamming measure performs fairly well in comparison with alignments because the

probability of a match at any position is high. Alignment measures should be relatively better for fuzzier sequence generation processes, which present more difficult classification problems.

2.3 Character counting

Another way of comparing sequences is simply to count the number of characters of each type, creating a vector of frequencies that would add up to the sequence length. This approach ignores the sequential arrangement of elements and involves only counts of elements. It is analogous to the time budget representation of daily activities, which adds up episode durations, ignores their timing and transition, and reports only total time spent by activity category. At face value this would seem to be a poor representation of the information in sequential data, but its simplicity and ease of use demands that its reliability be tested also. No inter-item similarity measures are necessary because each item already consists of numerical measurements and clustering methods can be applied directly to the input data.

3. Software and data

3.1 Pairwise similarity and clustering

The pairwise alignment scores were computed using ClustalG. This is a version of ClustalW, which was developed at the European Molecular Biology Laboratory (Thompson et al, 1994), amended for use outside of biochemistry and is available from cwilson@cmhc-schl.gc.ca. ClustalG has deleted the explicitly biochemical features of its parent packages. ClustalG uses the Needleman-Wunsch algorithm for its computation of pairwise similarities. The results may be replicated by any software that implements that algorithm.

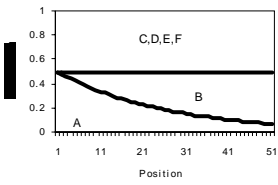
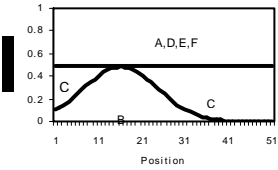
ClustalG writes pairwise similarity values to an output file. Hamming similarity measures were calculated from input sequences by the *Statistical Analysis System (SAS)*. We transformed all similarity matrices to distance matrices by subtracting similarity scores from the maximum similarity, then used the Ward clustering algorithm implemented in the *SAS CLUSTER* procedure to compare how well different measurement approaches identified known patterns in the test sequences. The Ward algorithm was one of a small number that perfectly classified the simplest set of synthetic sequences and so was used in the whole experiment.

3.2 Monte Carlo sequence generation

The Monte Carlo simulator written for this project produced sequences of the six letters, A to F, in a format known as “Pearson” or “Fasta” shown below. The experiments used 50 character sequences. The Courier font is convenient for illustrating sequence data because all characters are the same width. The “greater than” sign begins the record followed by the identifier. The data sequence follows on the next line.

```
> #1  
ADBEBDDAAADAABAACBCFBABFCFCDACBFECFAFBBBACECBCBCC
```

The patterns in the Monte Carlo sequences are produced by eight generation rules. The generators define patterns on the basis of negative exponential probabilities to the first and second powers and uniform probability distributions. The probability of the occurrence of a patterned character is a function of its position in the sequence. A parameter called PROB varies from zero to one and controls the portion of the probability space that is used for patterned characters, versus the proportion used for residual or “noisy” characters. Residual characters are equally probable in the space not occupied by patterned characters. Higher values of PROB reduce noise and create clearer patterns. Table 2 defines the generation rules and shows the probability spaces for patterned characters and for the residual characters.

Table 2: Eight Monte Carlo Sequence Generation Rules			
Rule identifier	Function	Description	Probability space
ABnex	Negative exponential	[A] located in the initial positions and [B] located towards the right end of the sequence. Other characters are uniformly located across the sequence. Pattern letters: A, B Residual: C, D, E, F	 $P = \text{PROB} * \exp(-.046 * k)$
BAnex EFnex FEnex	Negative exponential	[B] and [A] reversed from ABnex. Pattern letters: B, A Residual: C, D, E, F	As ABnex
BCnx2 DEnx2	Second order negative exponential	[C] is concentrated in left and right positions. [B] located in left-centre. Pattern letters: B, C Residual: A, D, E, F	 $P = 0.2 * \text{PROB} * \exp(.21 * k - .007 * k^2)$
ABC75 DEF75	Uniform	Majority of letters A, B, and C but no left-right trend. PROB not used.	Probability (A or B or C) = 0.75 Probability (D or E or F) = 0.25

Sequences generated by the same rule will all be different because the selection of a letter at each position is probabilistic. However, sequences will share characteristics written into the generation rules. Sequences generated by different rules should be recognizably different but, given the probabilistic nature of the processes, it is possible for individual sequences to resemble those made by different rules. Figure A1 in the appendix shows two examples of each of the eight sequence types for PROB set to 0.80. The sequences are highly distinct and identification of the generators is left as an exercise for interested readers. Identification of examples from other datasets is considerably more difficult.

3.4 Performance of the generator

Table 3 shows the probabilities of each character expected in a 50-character sequence with PROB set to 0.5 and the default exponential parameters. The 240 sequences contain 12,000 characters and the table also shows the expected and actual character counts. The residual characters in the exponential generation rules will always have theoretical probabilities of one quarter of the value (1 – PROB), or 12.5 percent when PROB is 0.5. The probabilities of the characters set by the exponential functions are derived by integrating the functions from zero to 50. Chi-squared tests of the actual character counts indicate that frequencies are not significantly different from those expected.

Table 3: Probabilities of characters generated by eight rules							
Generation rule	Character distribution (%)						# chars
	A	B	C	D	E	F	
ABC75	25	25	25	8	8	9	1500
DEF75	8	8	9	25	25	25	1500
ABnex	20.1	29.9	12.5	12.5	12.5	12.5	1500
BAnex	29.9	20.1	12.5	12.5	12.5	12.5	1500
EFnex	12.5	12.5	12.5	12.5	20.1	29.9	1500
FEnex	12.5	12.5	12.5	12.5	29.9	20.1	1500
BCnx2	12.5	19.8	30.2	12.5	12.5	12.5	1500
DEnx2	12.5	12.5	12.5	19.8	30.2	12.5	1500
Character counts							
expected	1995	2105	1900	1729	2261	2110	12000
observed	1983	2105	1903	1744	2223	2042	12000

3.5 Test data

Sequence similarity was measured for sets of 240 sequences composed of 30 sequences generated under each of the eight rules. Each experiment employed four sequence sets using different levels of the PROB parameter. The PROB80 set, based on a PROB value of 0.8, distinguishes very clearly among the eight generation rules. PROB was set to 0.50, 0.35, and 0.20, presenting progressively more difficult classification problems.

All sets were aligned with ClustalG to calculate similarity matrices using the global pairwise algorithm with a gap opening penalty of 1.0 and an extension penalty of 0.1. The Hamming similarity matrices and character counts for each set were also calculated. The experiment thus evaluates three analytical strategies for each sequence set. An analysis of variance of the similarity matrices produced very high F statistics for models that identified all possible combinations of within group and between group comparisons, indicating that an eight-way subgroup structure exists in the test data. Cluster analysis was then conducted on all matrices to see which similarity measure produced the best reconstruction of the original eight groups.

4. Evaluation of measures of sequence similarity

The inter-sequence distance data generated by the similarity algorithms was subjected to hierarchical clustering to produce tree structures defining similar groups. Clusters are formed at all levels of aggregation from the initial joining of individual pairs to the combination of large groups into a final tree. The selection of a correct or meaningful set of clusters is usually part of the objective of the research. Eight clusters are built into this experiment by the generator but the number of clusters in a research project would have to be justified in the context of the research problem.

Identification of clusters on the basis of their membership is simple when subsets in the data are distinct, as in the PROB80 sets, but becomes difficult when subsets are indistinct. The probabilistic determination of character frequency and location means that sequences generated by different rules may have a similar appearance, allowing clustering algorithms to join sequences from different generation rules in the tree building process. The BCnx2, ABnex, and BANex rules tend to produce some sequences that are similar at low PROB values. Developers of classification trees identify nodes by the characteristic that contributes most of the members and call this practice the “plurality rule” (Breiman et al, 1984). We follow this practice, identifying a cluster with the rule that contributes the most members, but call the latter the main rule.

For the PROB80 datasets, the identification of clusters in terms of the proportion of members contributed by the main rule can be considered definitive because the sequence appearance can be reliably associated with particular generation rules. In fact, PROB80 was chosen because some similarity measures could cluster the datasets perfectly at that level of uniformity. However, as the classification problem becomes more difficult, a given sequence may be generated by one rule but may resemble those created by another. At PROB35 or PROB20, the generation of a sequence by a rule is no longer sufficient reason to believe that the sequence must look like others made by the same rule and the assignment of the sequence to a different rule is not necessarily an error. At the lower PROB values, the classification problem evolves from one of evaluating a candidate assignment against a true criterion, to a problem of measuring the agreement of independent raters.

The literature on rating agreement is extensive as the problem is central to many fields, including medical diagnostics, psychological testing, educational testing and others (Agresti, 1996, Mackinnon, 2000). The consensus regarding testing is that simple measures of agreement between raters (or concordance) should be used.

An 8x8 contingency table will disaggregate the membership of each identified cluster into the rules that generated the sequences. The marginal values of such a table are the rule counts and cluster sizes. Marginal totals are all 30 for the generation rules and are variable but sum to 240 for the clusters. The diagonal cells count the number of agreements between the generation rules and the cluster membership. The sum of the diagonal elements gives the total concordance, which, divided by 240, gives the proportional agreement or the proportion of the sequences from the main rule in all clusters. These ratios will be the main indices of classification success.

It should be emphasized that the objective here is to measure relative agreement rates of different methods, not to test the independence of ratings. If the classification procedures are effective we would expect statistical tests to reject a hypothesis of independence between generation rules and clusters. Chi-squared statistics for some tables exceeded 400, which for 49 degrees of freedom, does in fact reject.

The mixing of appearances of sequences generated by different rules has parallels in ordinary behaviour. Diary data that records activities will contain some random content. The activities of, for example, an employed man could be the same as those of a retired woman on a particular day if unusual events occur, even though the typical patterns of these two types are vastly different. A sequence classification would, properly, include both of these patterns in one group for that day. Identification of individuals grouped in this way according to their socioeconomic characteristics is another research problem.

5. Results

5.1 Similarity statistics

Table 4 shows the averages of the means and standard deviations of the alignment scores and the Hamming similarities for the ten sets of PROB80 and PROB20 sequences. The table contains statistics for the whole matrix and for the sub-matrices of within-rule and between-rule sequence comparisons.

The 240 sequences in a set generate 28,680 unique sequence similarity comparisons. No similarity is defined for a sequence with itself. Every similarity matrix is composed of 36 sub-matrices of within-rule and between-rule comparisons. The eight within-rule sub-matrices are symmetrical so the 30 sequences have 435 unique comparison pairs. These are the diagonal cells of Table A1. The 28 between-rule matrices contain 30x30 combinations of inter-rule comparisons and thus each has 900 comparison pairs. Thus, the observed similarities from a sample of sequences will be dominated by inter-group comparisons (if meaningful groupings can be shown to exist).

The alignment mean score was 182 for the PROB80 dataset and increased to 233 for PROB20. Means for the whole matrices are significantly lower than any of the within-group means, indicating that the within-group similarities are higher than between-group similarities, as was intended. The Hamming similarities decline slightly as the sequences become less similar.

An unexpected feature of the data in Table 4 is that average similarities are higher and less dispersed for the PROB20 sequences than for the PROB80 sequences, which are more distinctive. The explanation is that, although more distinctive sequences have higher mean within-rule similarities, they have lower between-rule similarities. Since the between-rule similarity matrices are twice as large as the within-rule matrices and are more numerous, the lower similarity values dominate the whole matrix. The means increase by less than five percent from the PROB50 to PROB35 datasets. This suggests that as the proportion of noise in the generation process becomes very large (more than 50%), the average similarity values stabilize.

Table4					
Descriptive statistics (PROB80, PROB50, PROB35 sequence sets)					
Classification problem	n	Alignment similarity		Hamming similarity	
		mean	stdev	mean	stdev
Easy (PROB80)					
Whole matrix	28680	182	74	88	59
Within rule	3480	316	37	195	63
Between rule	25200	164	58	74	41
Very hard (PROB20)					
Whole matrix	28680	233	23	84	28
Within rule	3480	245	17	96	28
Between rule	25200	230	23	82	27

To give some context to the descriptive statistics for the experimental data set, we coded 391 randomly selected diaries from the Statistics Canada 1992 General Social Survey of time use into six activity categories using 30 minute time intervals. The resulting activity sequences averaged 53 characters (because short episodes were rounded up), which is quite close to the length of the experimental sequences. An alignment with the local, high penalty settings gave a mean pairwise similarity score of 260 with a standard deviation of 65. The Monte Carlo sequences thus display lower similarities than time use diary data.

5.2 Cluster analysis of three methods

Table 5 shows all clusters present in the first PROB80 dataset. Note that alignments employ indels only and give no partial scores for substitutions. The table identifies the clusters and gives the size and the count of sequences contributed by the main rule for each cluster. For a simple classification problem, an ideal clustering will produce eight clusters of 30 sequences with all sequences from one rule in each cluster. For more complex classification problems, solution quality can only be defined in terms of a more reliable procedure, since sequences made by different rules can look very much alike in terms of the number, order and transition patterns of characters. Note that the character count results come from a clustering of character frequencies, not similarity matrices.

Table 5 PROB80 cluster analyses						
Cluster name	Similarity matrix					
	Alignment		Hamming		Character count	
	Size	# main rule	Size	# main rule	Size	# main rule
ABC75	30	30	29	29	29	29
DEF75	30	30	30	30	30	30
ABnex	30	30	31	30	33	30
BAnex	30	30	30	30	27	27
EFnex	30	30	30	30	36	29
FEnex	30	30	30	30	24	23
BCnx2	30	30	30	30	31	30
DEnx2	30	30	30	30	30	30
Total	240	240	240	239	240	228
% main rule		100		99.6		95.0

All similarity measures, including the Hamming matrices, were highly reliable. Since at least one option could distinguish the generation rules perfectly, assignment of a sequence to a cluster generated by a different rule can be treated as an error. Table 6 shows the agreement rates for the four sequence sets based on pattern percentages of 80, 50, 35 and 20.

Table 6 Percent Agreement rates for cluster analysis of Monte Carlo Sequences			
Score	Alignment	Hamming	Character count
PROB80	100.0	99.1	95.0
PROB50	90.4	77.1	68.8
PROB35	63.3	50.0	45.4
PROB20	55.0	43.8	51.7

Alignment similarity measures are more effective in recovering known patterns than either Hamming similarities or character counts at all levels of classification difficulty. The Hamming similarity and character counting gave excellent agreement for the simpler PROB80 classification problem, but their agreement rates declined substantially for the PROB35 sets.

A purely random assignment of 240 objects to eight clusters would be expected to produce 30 correct assignments or 12.5 percent agreement. The PROB80 sequences present an analytically simple classification problem and all methods produce excellent agreement rates. As the classification problem becomes more subtle, the advantage of the alignment method increases until the noise proportion of the information rises to 80%. For the PROB20 sequences the relative advantage of alignment declines but it still remains most effective. Surprisingly, for the PROB20 data character counting (time budgets) increases its reliability and is nearly as accurate as alignment.

The results indicate that the clearest, least ambiguous classification strategy is to align with low gap penalties. This does not discriminate clusters perfectly even for the PROB80 datasets, although here it is highly reliable, exceeding 99% in multiple tests and scoring 100% in this example.

5.3 Cluster analysis of alignments with substitutions

Two additional alignments were performed on the PROB50 and PROB35 sequences. One employed the transition scoring matrices for the two sequence sets as described above with indel penalties set to 500. This completely suppressed indel operations and gives a variant of the Hamming results with much larger similarities. The second run for each sequence set used both the substitution matrix and the original indel values (1.0; 0.1). The substitution scores for each sequence set are shown in Table 7. As noted above, rewarding the most frequent non-identical transition with a score of 6 and less frequent transitions with lower scores yielded scoring matrices with little variability. Other systems are of course possible.

Table 7													
Similarity scores for character substitutions													
PROB35							PROB50						
	a	b	c	d	e	f		a	b	c	d	e	f
a	10	6	6	5	5	5	a	10	6	5	4	5	4
b		10	5	5	4	4	b		10	4	4	4	4
c			10	5	5	5	c			10	5	6	5
d				10	5	6	d				10	5	6
e					10	6	e					10	6
f						10	f						10

The agreement rates for the substitution experiments are shown in Table 8. The first row of results is reprinted from Table 6. The addition of these particular substitution scores reduces the effectiveness of the alignment classifications.

Table 8			
Percent Agreement rates for alignments with substitution scores			
Alignment		PROB50	PROB35
Gap values	Substitution matrix		
1, 0.1	none	90.4	63.3
1, 0.1	yes	78.3	57.5
500,500 (no gaps)	yes	57.1	38.3

While it is possible that further testing may find substitution rules that improve the accuracy of classification employing both substitution scores and gap penalties, I am not optimistic. If the objective is to find groups in samples that display similar behaviour as indicated by similar activity patterns, then it would be expected that the significance of substitution of one activity for another would be different for different groups. Appropriate substitution scores could only be determined after the groups have been identified and these should differ from group to group. From this perspective, the search for an ex ante general purpose substitution matrix is misguided.

6. Strategies for alignment analysis of event sequences

The most successful alignment strategy for identifying generation rules using all test datasets was to use very low gap penalties. Most social science alignment applications have used the algorithm with relatively high gap penalties. The conclusion from this experiment is that where the objective is classification the similarity and distance measures of event sequences that employ very low gap penalties can be clustered into clearer behavioural patterns than measures based on Hamming similarity or time budgets.

An unexpected result that emerged from the analysis was that average sample similarities are higher among sequences with weaker patterns than those with stronger patterns. Strong patterns mean that inter-group similarities are low and there are more possible inter-group comparisons in a sample than within-group comparisons. Researchers may find highly distinctive behavioural groupings in samples with relatively low average similarity measures. Conversely, samples with relatively high average similarity may contain groups with indistinct behavioural patterns. Consideration of average similarity values points the researcher to populations with lower rather than higher scores.

In real research situations, the analyst will not normally know how many, if any, patterns are present in the data because finding such patterns is probably the object of the research. There would then be no justification for picking a certain number of clusters to find. A possible strategy would be to deliberately set the number of clusters to be fairly high, say, the square root of the sample size. In this case that would be about 16. Groups identified in this way would generally be more homogeneous than the larger final clusters extracted in this experiment. By stopping the clustering procedure early, analysts can determine for themselves the best way to put the building blocks together.

We do not interpret a mix of sequences from different rules as a classification error unless we have reason to believe that some method could have classified the groups perfectly, as in the case of the PROB80 dataset. Rather, we argue that different generation processes with some random elements simply do not produce highly distinct character sequences. We interpret the agreement rates quoted here as the lower bounds of the classification reliability. The true assessment of classification reliability can be made only in terms of a better classifier. To date, for broad classes of sequential social events, the most reliable classifiers are alignment algorithms.

Acknowledgements

I would like to thank Julie Thompson-Maaloum and Chang-Hyeon Joh for their careful reading and detailed comments on a draft of this paper. Remaining weaknesses are my own responsibility. Development of ClustalG was conducted by the *Activity Settings, Sequencing, and Measurement of Time Allocation Patterns* research project funded by the Social Science and Humanities Research Council of Canada at the Time Use Research Centre, St. Mary's University. Andrew Harvey was the principal investigator and I gratefully acknowledge his interest and support.

References

- Agresti, A, 1996, *An Introduction to Categorical Data Analysis*, New York: Wiley.
- Breiman, L., Friedman, J., Olshen, R., and Stone, C. 1984, *Classification and Regression Trees*, Wadsworth, Belmont California.
- Carmines, E. G. and Zeller, R. A. 1979, *Reliability and Validity Assessment*, Sage University Paper 17, Sage Publications, Beverly Hills.
- Mackinnon, A., 2000, A spreadsheet for the calculation of comprehensive statistics for the assessment of diagnostic tests and inter-rater agreement, *Computers in Biology and Medicine*, vol. 30, no. 3, pp. 127-134.
- SAS Institute Incorporated, 1985, *SAS Users Guide: Statistics, Version 5*, Cary, North Carolina.
- Thompson, J.D., Higgins, D.G. and Gibson, T.J. 1994 CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673-4680.
- Wilson, W.C. 2006, Reliability of sequence alignment analysis of social processes: Monte Carlo tests of ClustalG software, *Environment and Planning*, volume 38, pp. 187-204.

Appendix

Table A1		
Sample sequences from eight generation rules (PROB80 dataset)		
> #1	ADBBEBDDAAADAABAACBCFBABFCFCDACBFECFBBBACECBCBCC	
> #2	CEDBABDDCABBAACECCDCBFDCBCCFFDAABBBAFECBAFBDBBCAB	
> #3	AAFADADDCFCDEFDFEAEBEDEFDFDCFFBBDCEFEFCFCEDAFBDE	
> #4	FFEDDDFFDEEDDEDFCDCBCCFFBFDDBDCEFDDEDFDEBDFDFDCFF	
> #5	AAFAAFADABAACBBAABBFBBBABABBBBBBBBBBBBBBBBBBBEBA	
> #6	EEADAFACAAAFAFABADBAAEAFABBBAAAABBBABDBBBBFBFB	
> #7	BBBBBBDBABBADDEBBABFAAFFDABDDBAABAAEAAAAAAAAADBF	
> #8	BBBBBFBABBABCBBABAADADBABDAFFABAABAAAAAAAAAFAABDE	
> #9	ECEEEFBFEDEEEEFCEFABEAFEFEEFFFAEFFDEFBFAFFCFFFFF	
> #10	EEDCEDEEEEFCEFFFEFFFAEFCEFFCFEFFFFFEDDFFFAFFEF	
> #11	FFFFFFFFFFFFEAEDCFEFAEBDDFFEAEBEEEEEEEEEBBEEAE	
> #12	FFFFFAFFBEECAEFEEFEDEAFBEECEFEDEEEEEEEEEEAEEAD	
> #13	CCBBBCBBBCBCCBBBBBBBCCBBBCCCCDCFCDCCCCCCFEFCFDCC	
> #14	BCCCCBBBEBBBBBBFECBABBBCBECCCBDFCEACFCCCCACCFE	
> #15	DDEDEEDDBDDDCBDDDDDEECDECEDEEAFBECEFEFEDEEEEEE	
> #16	DEDEDEDEDDDDDDDEDDDDCEFEFEDEEEEEEDEFEDEEEEEE	

Note: For input to ClustalG, sequence labels (> #1 etc.) are on a line preceding the sequence data.

Table A2									
Counts of Pairwise comparisons									
(all combinations of generation rules)									
		ABC75	DEF75	ABnex	AB1-nx	EFnex	EF1-nx	BCnx2	DEnx2
	ABC75	435	900	900	900	900	900	900	900
	DEF75		435	900	900	900	900	900	900
	ABnex			435	900	900	900	900	900
	AB1-nx				435	900	900	900	900
	EFnex					435	900	900	900
	EF1-nx						435	900	900
	BCnx2							435	900
	DEnx2								435

Within group 3480
 Inter-group 25200
 All 28680