Spell Sequences, State Proximities, and Distance Metrics

ELZINGA, C. H., STUDER, Matthias

Abstract
Because optimal matching (OM) distance is not very sensitive to differences in the order of states, we introduce a subsequence-based distance measure that can be adapted to subsequence length, to subsequence duration, and to soft-matching of states. Using a simulation technique developed by Studer, we investigate the sensitivity, relative to OM, of several variants of this metric to variations in order, timing, and duration of states. The results show that the behavior of the metric is as intended. Furthermore, we use family formation data from the Swiss Household Panel to compare a few variants of the new metric to OM. The new metrics have been implemented in the freely available TraMineR-package.

Reference

DOI: 10.1177/0049124114540707

Available at:
http://archive-ouverte.unige.ch/unige:78562

Disclaimer: layout of this document may differ from the published version.
SPELL SEQUENCES, STATE PROXIMITIES AND DISTANCE METRICS
Abstract. Because OM-distance is not very sensitive to differences in the order of states, we introduce a subsequence-based distance measure that can be adapted to subsequence length, to subsequence duration and to soft-matching of states. Using a simulation technique developed by Studer (2012), we investigate the sensitivity, relative to OM, of several variants of this metric to variations in order, timing and duration of states. The results show that the behavior of the metric is as intended. Furthermore, we use family formation data from the Swiss Household Panel to compare a few variants of the new metric to OM. The new metrics have been implemented in the freely available TraMineR-package. (Gabadinho et al., 2011).

Key Words: sequence-analysis, OM, subsequence, soft-matching, duration-weighing
1. Introduction

Sequence analysis is the generic name for a variety of methods that subserve the analysis of state sequences like life courses and job careers.\textsuperscript{1} Today, sequence analysis has become one of the standard-toolboxes for those who analyze sequence data and sophisticated, user-friendly software for such methods is freely available (e.g. Brzinsky-Fay et al., 2006; Elzinga, 2009; Gabadinho et al., 2011).

To compare sequences, one needs a measure of distance or similarity between pairs of sequences and by far the most frequently used metric to generate such distances is the so-called OM-metric\textsuperscript{2}. The OM-metric expresses distances in terms of the minimum cost of a sequence of edit operations that turns the one sequence into an exact copy of the other sequence. In the sequel, we will write $d_{OM}$ to denote this metric.

Ample descriptions of the metric and the associated algorithm can be found in numerous sources, e.g. in Clote and Backofen (2000), in Martin and Wiggins (2009) and in Sankoff and Kruskal (1983). Largely motivated by the problems of determining the weight or cost of the edit-operations involved, many variants of the metric have been proposed, some quite general (e.g. Gauthier et al., 2009; Halpin, 2010; Moen, 2000), others more application-specific like in e.g. Lesnard (2008). For a comprehensive overview of OM-variants, the reader is referred to Studer (2012).

For various reasons, the use of OM in the social sciences has been widely criticized, most notably in Settersten and Mayer (1997); Dijkstra and Taris

\textsuperscript{1}Authors Note: This publication is part of the research works conducted within the Swiss National Centre of Competence in Research LIVES – Overcoming vulnerability: life course perspectives, which is financed by the Swiss National Science Foundation.

Part of the content of this paper was presented during the Lausanne Conference on Sequence Analysis (LaCOSA’12) at the University of Lausanne on June 6-8, 2012.

\textsuperscript{2}We will use the acronym OM to refer to the metric and to an algorithm required to calculate the metric.
(1995); Wu (2000); Levine (2000); Elzinga (2003) and in Lesnard (2008). The first major point of critique has been that, in the social sciences, edit operations have no interpretation; they cannot be interpreted as spontaneous or selection-driven mutations like in microbiology. However, OM can be interpreted in a way that does not involve edit-operations at all but instead refers to the concept of a longest common subsequence (lcs). When we suppose that the cost of inserting or deleting any character equals 1 and the cost of substituting one character for another, distinct character equals 2 (one deletion followed by one insertion), we have that

\[ d_{OM} = \ell(x) + \ell(y) - 2\ell(lcs(x,y)) \] (1)

wherein \( \ell(x) \) denotes the length of sequence \( x \). Hence, the OM-distance equals the number of characters in both sequences that do not belong to a longest common subsequence of the pertaining sequences \( x \) and \( y \). If we interpret \( lcs(x, y) \) as a “common backbone” or “common narrative”, then \( d_{OM} \) decreases with the relative length of that common backbone. So, we don’t really need to interpret the edit-operations that are often used to define the OM-distance and that explain the logic of the OM-algorithm. In case the edit-costs have been set differently, the interpretation of OM-distance is in terms of an \( lcs \) that does not equally weigh all states.

The second major criticism of OM pertains to the fact that often there is no objective way of establishing the edit costs of the edit operations. Various, more or less sophisticated, ways of deriving edit cost from state-transition frequencies have been devised (see e.g. Gauthier et al., 2009) but these methods do not resolve the basic issue: establishing the proximity or similarity of states and how this similarity can be derived and operationalized from social science theory. This matter has not been resolved and cannot be resolved
within the framework of sequence analysis (Studer, 2012). On the other hand, Hollister (2009) and Studer (2012) propose promising strategies to establish state-proximities through scaling strategies that are independent of sequence analysis. Despite the trouble we have in finding acceptable ways of establishing state-proximities or -similarities, we cannot do without them. To illustrate this point, we consider three toy-sequences from the domain of family formation, using the states Single, Unmarried cohabitation, Married and Married with Children:

\[ x: S \ M \ MC, \]
\[ y: S \ U \ MC, \]
\[ z: S \ S \ MC. \]

Whatever metric we use, we should find that \( x \) is closer to \( y \) than it is to \( z \), simply because the state \( U \) is more similar to the state \( M \) than to the state \( S \). The second reason to consider state-similarities is that it is a key feature to compute multichannel distances (Pollock, 2007; Gauthier et al., 2010).

According to some authors, it is more convenient to invest in defining OM costs rather than moving to a more analytical definition of the dissimilarities. Perhaps that would be a viable strategy when the issue of establishing edit cost would be the only challenge for OM in particular or sequence analysis in general.

However, not so well known or ignored is the fact (Elzinga, 2003; Studer, 2012) that OM is not very sensitive to differences in the order of the states of a pair of sequences. As an example, we consider the three toy-sequences below representing careers, using state \( d \) for “director”, \( m \) for “manager” and \( e \) for “employee”:
According to the OM-metric, sequences $x$ and $y$ are the closest pair because they share a long “common narrative”: the 7 time-units spent in the m-state. However, sequences $x$ and $y$ reflect opposite career dynamics: $x$ can be interpreted as an “ascending” career, $y$ is a “descending” one. There are two reasons for this lack of sensitivity to ordering. First, OM is context-insensitive: each state is handled separately without considering previous or following states (Halpin, 2010; Hollister, 2009). Edit operations are applied on symbols in strings, irrespective of their context (previous and next states notably). Hence, the dynamics of the trajectory is not explicitly handled by OM. Second, the only way to handle duration through OM is by repeating the same state with a frequency that corresponds to the number of time units intended.

In the social sciences, trajectories are often coded with a few relatively long spells. As a result, the “common” backbone is often strongly linked with the total time spent in some states, ignoring the underlying dynamics coded with small spells. This lack of sensitivity to ordering is problematic since sequence analysis is about differences between categorical time-series were event- or state-orderings are defining the sequences. Moreover, the ordering of the states reflects the internal dynamics of a trajectory; one of the important aspects that sequences analysis claims to take into account. Therefore, Elzinga (2003, 2005) proposed a distance metric that is based on a subsequence-based vector space. Subsequences allow analyzing states in their contexts and thus the dynamic of the trajectory. In our example, the subsequence $ed$ (“employee”–“director”)
provides essential information to study the dynamics of the trajectory. Simulations presented by Studer (2012) demonstrate that subsequence based metrics are much more sensitive to differences in state-orderings.

However, Elzinga’s metric does not allow for different state-proximities: all states are considered as equally different (see also Hollister, 2009). Therefore, we propose a very flexible generalization of Elzinga’s subsequence-based metric that does handle such state proximities. The metric has a number of interesting properties that are best explained through representing sequences as vectors in a vector space with a Euclidean norm.

As said before, one limitation of OM is the way it handles time. This limitation is caused by the fact that OM counts edits applied to symbols in a string and has no inherent mechanism to deal with quantities like duration. Therefore, the observation that someone was unemployed ($U$) for 10 months and then found herself a job for the next 30 months has to be translated to a sequence of the form

$$UU\ldots UEE\ldots E\ldots E_{10}^{30},$$

i.e. as a sequence of 40 observations of states. This mapping of durations to strings of states severely limits the way in which time or duration can be handled (see e.g. Halpin, 2010). However, representing the observations as

$$(U,10)(E,30),$$

i.e. as two states, one with a duration of 10 months and one with a duration of 30 months may seem more natural. Choosing a different time scale, say years, would then invite to write $(U, .833)(E, 2.50)$, or $(U, 300)(E, 900)$ when the scale were days instead of years or months.

Formally, a state sequence like e.g. $x = abac$ is a concatenation of states from
some alphabet $\Sigma = \{\lambda, a, b, \ldots\}$ and a $k$-spell sequence consists of a pair

$$(x, t) = (x_1, t_1)(x_2, t_2) \ldots (x_k, t_k),$$

i.e. a state sequence and a numerical sequence. Using a metric that can handle time as a quantity that can be separated from the states, would allow for a more sophisticated treatment of the time dimension (see e.g. Abbott and Hrycak, 1990; Halpin, 2010).

The purpose of this paper is to discuss a family of distance measures that is quite sensitive to differences in the sequencing of the states or events, that allows for proper duration handling and time transforms and that uses state proximities. We will discuss the metrics and we will demonstrate their sensitivity, relative to OM. Finally, we will demonstrate their practical use in an application to family formation.

Thereto, the next section introduces the representation of sequences through feature-vectors, the features being subsequences. Section 3 then discusses soft-matching, the use of state proximities, and the required transform of the vector-space. Section 4 discusses spell-sequences: sequences where duration is treated as a property of the states. In Section 5, we discuss the unifying framework of a feature vector representation and in Section 6, we assess the sensitivity of the metrics to differences in sequencing, timing and duration of the pertaining states. Finally, in Section 7, we apply the newly introduced metrics to family formation data and compare the results with those obtained using OM. In Section 8, we discuss our findings and the merits thereof.

2. Sequences as Vectors

2.1. Vectors and distances. What makes vector-representations so interesting? Vector representations are interesting because once we have vectors,
there is a whole family of distance measures that are proper metrics in the sense that these distance measures satisfy the axioms of a metric:

\begin{align*}
\text{D1: } & d(x, x) = 0, \\
\text{D2: } & d(x, y) > 0, \\
\text{D3: } & d(x, y) = d(y, x), \\
\text{D4: } & d(x, z) \leq d(x, y) + d(y, z).
\end{align*}

D1 states that an object has one location only and D2 states that two distinct objects cannot be in the same location. D3, the symmetry axiom, states that direction does not affect distance, and D4, the so-called triangle inequality, states that “a detour takes at least as much time” or, put differently, that when two objects \((x\) and \(z\)) are close to a third object \((y)\), they cannot be remote from each other.

The triangle inequality is not only important because it formulates an intuitive property of a quantified space. The triangle inequality also ensures that objects can be located with respect to each other without other objects being involved. If the triangle inequality would not hold, the observation of \(d(x, z)\) would not be very meaningful if there could exist some (unobserved) \(y\) such that \(d(x, y) + d(y, z) < d(x, z)\). When working with groups of sequences (such as clusters), the triangle inequality ensures that a particular observation \(y\) does not artificially create an homogeneity in the group by “attracting” objects that would be considered as very distant when \(y\) would not be observed.

Finally, the triangle inequality ensures that the space exhibits a certain regularity or smoothness in the sense that at least some of its properties are invariant in all directions. If this were not true, the space, the representation
of the sequences in a distance matrix, would not be very meaningful. Let us illustrate this remark: imagine that we have observed a set of sequences \( \{x, y, z, \ldots\} \), say \( N = 1000 \) sequences, and that we have somehow established distances between the pairs of sequences. Now suppose that we add a new observation, a new sequence \( p \), to our data set. We may not know yet how to localize this \( p \) in the spatial representation of the \( N \) sequences. However, if the space is metric in the sense that the distances satisfy the axioms D1-D4, we know that we have

\[
d(p, x) \leq d(p, y) + d(y, x)
\]

for all pairs \( (x, y) \) of known sequences. So, given a metric space with \( N \) objects, the distance of the new object to all known objects must satisfy \( \binom{N}{2} = N(N - 1)/2 \) restrictions of the form of Equation (2). Even if Equation (2) is satisfied for all pairs \( (x, y) \), this does not imply that

\[
d(x, y) \leq d(x, p) + d(p, y)
\]

holds for all pairs \( (x, y) \) too. Thus, the number of restrictions, i.e. inequalities, that the new sequence must satisfy, equals \( 2\binom{N}{2} = N(N - 1) \). For a moderate data set of say, thousand sequences, this amounts to almost one million inequalities to be satisfied.

So the triangular inequality severely limits the location of the new sequence \( p \) in sequence space: the distances to \( p \) cannot be wildly at variance with what we expect on the basis of what we already know about the space. Furthermore, when we add \( p \) to the space, i.e. when we enlarge our knowledge from \( N \) to \( N + 1 \) sequences, the next sequence \( q \) to be added will have to satisfy \( 2\binom{N+1}{2} = 2\binom{N}{2} + 2N \) inequalities. So, the more sequence distances we know, the more we know, in terms of the number of inequalities to satisfy, about
the structure of the space: adding more sequences and gauging their distances makes sense. Conversely, if we drop the triangular inequality as a requirement, gathering new data would not add much to our understanding of sequences through a spatial representation. Therefore, it is essential that the procedures with which we assign (distance-)numbers to pairs of sequences ascertains that the triangular inequality is satisfied. Moreover, many distance analysis methods such as clustering algorithms or discrepancy analysis are based on these axioms.

We know (see e.g. Clote and Backofen, 2000) that the OM-distance $d_{OM}$ satisfies all four axioms, provided that the edit cost function is a metric over the state-alphabet. So, the OM-algorithm as such does not guarantee a proper metric; one needs a metric cost-function as well. Examples of metric and non-metric cost functions are shown in Table 1. However, many of the variants of OM that somehow (dynamically) adapt the standard edit cost matrix may lead to violations of the triangle inequality (Studer, 2012).

Insert Table 1

Once vectors are available, it is easy to calculate Euclidean distance $d_E$: for vectors $\mathbf{x} = (x_1, x_2, \ldots)$ and $\mathbf{y} = (y_1, y_2, \ldots)$, we have that

$$d_E(\mathbf{x}, \mathbf{y}) = \left( \sum_i |x_i - y_i|^2 \right)^{\frac{1}{2}} \quad (4)$$

$$= \sqrt{\mathbf{x}'\mathbf{x} + \mathbf{y}'\mathbf{y} - 2\mathbf{x}'\mathbf{y}}. \quad (5)$$

In this paper we will only be concerned with Euclidean distances since they can be evaluated without even “knowing” or constructing the vectors explicitly: from Equation (5) we see that we can evaluate the distances provided
that we have access to the values of the inner products $x'x$, $y'y$ and $x'y$. The vectors that we will construct will appear to have extremely high dimension but fortunately, there exist efficient algorithms, so-called “kernels” (see e.g. Schölkopf and Smola, 2002), that evaluate inner products without requiring the coordinate values of the pertaining vectors.

Finally, vector spaces are very attractive to work with because they have been amply studied in linear algebra (see e.g. Meyer, 2000) and much of this knowledge is exploited in the standard multivariate statistical models.

In the next two subsections, we will discuss how to construct vectors from sequences. Essentially, this is a new presentation of Elzinga’s proposals as discussed in Elzinga (2003) and Elzinga (2005). These new presentations allow us to discuss vector representations without referring to algorithms for the evaluation of vector products. In Section 2.3 we discuss an easy extension and in Sections 3 and 4 we exploit the representation to discuss more advanced issues like the handling of time and state-matching.

2.2. The basic representation. We will construct vectors from sequences through using the concept of “subsequence” so we begin with elaborating on this concept. For a more formal treatment, the reader is referred to e.g. Apostolico and Cunial (2009); Crochemore et al. (2007) or Elzinga et al. (2008).

Consider the toy-sequence $x = x_1x_2x_3x_4 = abac$ over a 3-state alphabet $\Sigma = \{a, b, c\}$. In this and the next subsections, we will use this toy-sequence to illustrate the basic principles of constructing vectors from sequences and some elementary variants thereof.

We may take any nonnegative number of states from $x$ and we will then be left with a subsequence of $x$: a subsequence $u$ of states that have the same order in $x$ and we will write $u \subseteq x$ to denote such fact. For example, when we take out the $a$’s from $x$, we will be left with $u = bc$, one of the four 2-long
subsequences of $x$. At most, we can take away all states from $x$ and we will then be left with an empty sequence for which we use the symbol $\lambda$. We might also take the smallest nonnegative number of states from $x$, zero states, and we would be left with $x$ itself and hence we conclude that $x \subseteq x$. The reader might want to verify that $x$ has 13 distinct subsequences, including $\lambda$ and $x$ itself.

We will now use the concept of subsequence to construct a vector-representation $x$ for the sequence $x$. We do this by first defining coordinates that correspond to all possible sequences that can be constructed from the alphabet $\Sigma$ and then construct binary vectors by setting those coordinates to 1 that correspond to sequences that occur as a subsequence in $x = abac$

$u : a \ b \ c \ aa \ ab \ ... \ cc \ aaa \ ... \ aba \ ...$

$r(u) : 1 \ 2 \ 3 \ 4 \ 5 \ ... \ 12 \ 13 \ ... \ 16 \ ...$

$x_{r(u)} : 1 \ 1 \ 1 \ 1 \ 1 \ ... \ 0 \ 0 \ ... \ 1 \ ...$

Formally, from $\Sigma$, we construct the set $\Sigma^*$ of all sequences that are constructible from $\Sigma$ and we fix the order of the elements of $\Sigma^*$, say in lexicographical order. Then we map the ordered sequences to the nonnegative integers $\mathbb{Z}^+$, i.e. each sequence $u \in \Sigma^*$ is mapped to a unique\(^3\) integer $r(u) \in \mathbb{Z}^+$ and we use these integers to index the coordinates of the vectors. So, for each sequence $x$, we construct a binary vector $x = (x_1, x_2, \ldots)$ such that

$$x_{r(u)} = \begin{cases} 1 & \text{if } u \subseteq x \\ 0 & \text{otherwise} \end{cases}. \quad (6)$$

\(^3\)Presuming $\Sigma$ is countable, the Kleene closure $\Sigma^*$ must be countable.
This construction characterizes strings by their subsequences and the resulting vectors are also called “feature vectors”, the subsequences being treated as features of the sequence.

The inner product $x'y = \sum_i x_i y_i$ counts the number of distinct common subsequences and therefore, the squared Euclidean distance $d^2(x,y) = x'x + y'y - 2x'y$ is measured in terms of the number of distinct subsequences that are unique to either $x$ or $y$. Intuitively, sequences are more similar when they have more features, more subsequences, in common.

In practice, this is a very appealing feature. It means that, using a cluster analysis, sequences grouped together will share the same subsequences. In a discrepancy analysis (Studer et al., 2011), a test would be significant if the subsequences of one group are significantly different from those of the other one. This would be similar to using MANOVA in the subsequence space.

The vectors so constructed have a countably infinite dimension since the index function $r$ is a bijection from $\Sigma^*$ to the nonnegative integers. Therefore, evaluating the inner product $x'y$ through $x'y = \sum_i x_i y_i$ is not feasible and one needs a kernel function (e.g. Elzinga et al., 2008; Schölkopf and Smola, 2002) to evaluate $x'y$ without explicitly constructing the vectors.

The representation in Equation (6) is very simple in the sense that it just uses the presence or absence of subsequences to represent the sequences and thus it is tempting to use substantially more interesting properties of the subsequences (provided that kernel functions exist that evaluate inner products of the resulting vectors). This is exactly what we will do in the subsections to come: define more sophisticated properties of the subsequences and use these to modify the distance measure according to its application.

2.3. Mapping Embedding Frequency. Returning to our toy-sequence $x = abac$, we observe that the subsequence $u = ac$ is embedded twice in $x$: as $x_1x_4$
and as $x_3x_4$. We denote this fact by writing $|x|_u = 2$.

Unfortunately, the sequence “Imprisoned, Probation, Convicted” is a subsequence that is embedded more than once in many a criminal career and we know that frequency of embedding of such subsequences is a relevant feature when comparing criminal careers. Similarly, the embedding frequency of the subsequence “Unemployed, Vocational Training, Employed” is an interesting feature of labor market careers.

From the above examples, we conclude that taking embedding frequency into account when comparing sequences may be a sensible thing to do and it is easily accomplished by constructing vectors through defining coordinates according to

$$x_{r(u)} = \begin{cases} 
|x|_u & \text{if } u \subseteq x \\
0 & \text{otherwise} 
\end{cases} \quad (7)$$

To interpret the meaning of $x'y$, it is convenient to introduce a new concept: the set $S(x, y)$ of distinct common subsequences of the sequences $x$ and $y$. When using the representation (7), evaluating the inner product $x'y$ amounts to calculating

$$x'y = \sum_{i} x_iy_i \quad (8)$$

$$= \sum_{u \in S(x, y)} |x|_u \cdot |y|_u \quad (9)$$

for if $u \notin S(x, y)$, either $|x|_u = 0$ or $|y|_u = 0$ or both. So, we interpret the value of $x'y$ as “the number of matching subsequences” of $x$ and $y$: for each $u \subseteq x$, there exist $|y|_u$ matches in $y$ hence the total number of matches equals $|x|_u \cdot |y|_u$ and we add these quantities for all $u \in S(x, y)$ when calculating $x'y$. 
The similarity and distance as proposed in Elzinga (2003) and Elzinga (2005) are in fact derived from the representation (7).

2.4. **Mapping Subsequence Lengths.** Most people share, in most kinds of careers, a lot of single states. For example, when studying family formation careers, we know that most people started living with their parents, then became parents themselves and before that, live together with a partner. Similarly, most people go to school before starting to work, etc. So, we may expect that many careers share the same short subsequences. Therefore, when counting the number of common or matching subsequences, i.e. when using representations (6) or (7), it might be interesting to weigh the counts according to the length of the subsequences counted by some convex function $L(\ell(u))$ of the subsequence lengths $\ell(x)$. This can be accomplished by the representation

$$x_{r(u)} = \begin{cases} \sqrt{L(\ell(u))} & \text{if } u \subseteq x \\ 0 & \text{otherwise} \end{cases}. \quad (10)$$

The square root is arising since we are interested in evaluating an inner product $x'y$ resulting in the proper weighing of the counts. When weighing the number of matching subsequences, we now attain

$$x'y = \sum_{u \in S(x,y)} \sqrt{L(\ell(u))} \sqrt{L(\ell(u))} \quad (11)$$

$$= \sum_{u \in S(x,y)} L(\ell(u)). \quad (12)$$

For example, by setting $L(a) = (a - 1)^p$ for $a \geq 1$ and $p > 1$, one would ignore common single states and assign progressively more weight to longer subsequences.
3. State Matching and Inner Product Spaces

In this section, we will extend the methods dealt with to using non-perfect matchings between states and, consequently, matchings between subsequences. All of the methods discussed so far construct vectors from sequences in order that the inner product of such vectors is equivalent to a weighted count of the common subsequences. Such counts are then used to construct distances and similarities. Incorporating subsequence matchings will allow us to also count non-perfect matches and weigh these appropriately.

We first have to define such matchings and this is the subject of the first subsection. Once defined, we will have to investigate how we can use them. This is nontrivial since the standard inner product counts the common subsequences that are perfect matches: the vector coordinates are indexed by the set of distinct subsequences and hence the inner product $\mathbf{x}'\mathbf{y} = \sum_i x_i y_i$ counts the number of common subsequences, “common” meaning that an exact copy of a particular subsequence occurs in the other sequence too. Therefore, we will need to extend the notion of an inner product in order to allow for counting not only exact copies but also approximate matches. This counting problem will be dealt with in the second subsection.

3.1. Matchings. We already argued that generating meaningful distances between sequences is not well possible without assessing the similarity or substitutability of the states or events involved. On the other hand, the actual assessment of such quantities is highly dependent upon the subject matter of the sequences so, in a methodological essay, it is not possible to detail the evaluation of state similarity. On the other hand, we have seen authors (e.g. Chen et al., 2009; Elzinga, 2014; Elzinga et al., 2011; Emms and Franco-Penya, 2012; Gower, 1971; Gower and Legendre, 1986; Tversky, 1977; Wang, 2006)
dealing the general issue of similarity measures and their properties. However, a detailed account of their ideas is far beyond the scope of the present paper. Here it suffices to state that we assume that we have somehow defined or constructed similarities between the states of the alphabet. With an alphabet \( \Sigma = \{\sigma_1, \ldots, \sigma_d\} \), this implies that we have a \((d \times d)\)-matrix \( M = (m_{ij}) \) such that \( m_{ij} \) denotes the degree of matching between states \( \sigma_i \) and \( \sigma_j \). We assume that the matchings satisfy \( 0 \leq m_{ij} = m_{ji} < 1 \) and \( m_{ii} = 1 \). Hence, \( M \) is a positive symmetric matrix of the form

\[
M = \begin{pmatrix}
1 & \ldots & m_{1d} \\
\vdots & \ddots & \vdots \\
m_{d1} & \ldots & 1
\end{pmatrix}
\]

For example, for the alphabet of living arrangements \( \Sigma = \{S, U, M, UC, MC\} \), we might have that \( M \) looks like (omitting zero’s)

\[
\begin{align*}
S & \begin{pmatrix} 1 \\ U \\ M \\ UC \\ MC \end{pmatrix} \\
& = \begin{pmatrix} 1 & .8 \\ .8 & 1 \\ 1 & .9 \\ .9 & 1 \end{pmatrix},
\end{align*}
\]

implying that being Married is very similar to living in Unmarried cohabitation and that this similarity even increases when there are children in the household. We not only compare states, we also compare (sub)sequences and we express the degree of matching \( m(x, y) \) of two equally long sequences \( x \) and \( y \) as the product of the matching coefficients of the consecutive states:

\[
m(x, y) = \prod_i m(x_i, y_i) \quad (13)
\]
For sequences $x$ and $y$ of unequal length, we set $m(x, y) = 0$. The reader notes that for two identical sequences, we always obtain $m(x, y) = 1$ and that, when two sequences have two states $x_i$ and $y_i$ with $m(x_i, y_i) = 0$, we obtain $m(x, y) = 0$. For the sequences of living arrangements, we obtain

$$m(x = S\ U\ UC, y = S\ M\ MC) = 1 \cdot 0.8 \cdot 0.9 = 0.72.$$  

Most importantly, we observe that given the matrix $M$ of state matchings, the matching of any pair of sequences, of whatever lengths, can be determined through using Equation (13), resulting in a matrix $M^*$. This matrix has entries that can be indexed by all sequences that can be constructed using the pertaining alphabet and has a structure that is illustrated in Figure 1 for an alphabet of just three states $a$, $b$ and $c$ where the sequences are ordered lexicographically to index the entries.

The reader observes that, due to the multiplicative structure of the matchings, the matrix has a very regular structure. The reader also notes that the submatrix containing the single-state matchings regularly reoccurs, in Figure 1 as $M$ in the upper left corner of $M^*$. It is not very difficult but beyond the scope of the present paper to prove that $M^*$ is singular only if this upper-left submatrix is singular. This equivalent to saying that the inverse of $M^*$ exists whenever the determinant of this submatrix is positive. In the example of Figure 1, this implies that $(M^*)^{-1}$ exists if $|M| > 0$. 

Insert Figure 1
3.2. Generalizing the Standard Inner Product. So far, we have discussed a basic vector representation of sequences that utilizes more or less sophisticated properties of the subsequences. The distance $d(x, y)$ corresponds to the length or “norm” of the line $\|x - y\|$ between $x$ and $y$ and we evaluated lengths of lines between vectors as

$$d(x, y) = \|x - y\| = \sqrt{\sum_i (x_i - y_i)^2} = \sqrt{x'x + y'y - 2x'y} \quad (14)$$

wherein $x'y$ denotes the inner product $\sum_i x_i y_i$. However, this way of defining the norm and inner product are not very helpful when we want to employ state similarities in evaluating distances. The reason is that the coordinates of the vectors refer to separate (concatenations of) states and evaluating $x'y = \sum_i x_i y_i$ is confined to comparing the values on the same coordinates as indexed by $i$. Therefore, we now turn our attention to a more general way of defining inner products and vector norms (see e.g. Meyer, 2000, Chapter 5). We say that a function $\langle \cdot | \cdot \rangle$ that maps pairs of vectors $x, y$ in a vector space $V$ to the nonnegative real numbers is an inner product, precisely when it satisfies, for all vectors $x, y \in V$, the conditions

i) $\langle x| x \rangle \geq 0$, equality holding if and only if $x = 0$

ii) $\langle x| y \rangle = \langle y| x \rangle$

iii) $\langle x| \alpha y + z \rangle = \alpha \langle x| y \rangle + \langle x| z \rangle$ for any scalar $\alpha$

The reader easily verifies that the standard function $x'y = \sum_i x_i y_i$ indeed satisfies the above requirements. To open up the possibility to incorporate
comparisons between unequally indexed coordinates, we first write the standard inner product, using the identity matrix $I$ in a trivial way:

$$x'y = x'Iy = \begin{pmatrix} x_1, \ldots, x_n \end{pmatrix} \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

But this trivial extension invites to exchange $I$ for the matrix $M^*$ of matchings and calculate

$$\langle x|y \rangle = x'M^*y = \sum_j \sum_i x_im_{ij}y_j$$

$$= x'y + \sum_{i\neq j} x_im_{ij}y_j$$

In Figure 2, we demonstrate that using an inner-product $\langle x|y \rangle = x'My$ with some non-trivial $M$ will “distort” Euclidean distance through stretching or compressing the vector space in particular directions. The plots show how equidistance contours in $\{0,1\} \times \{0,1\}$ change as a result of changing $m_{12} = m_{21}$ in $M = \begin{pmatrix} 1 & m_{12} \\ m_{21} & 1 \end{pmatrix}$, i.e. in a vector space representing sequences defined over just two states\(^4\).

Just like actually constructing vectors from sequences is hardly practically feasible, it is not feasible either to generate the full matrix of matchings $M^*$ since it has precisely as many rows and columns as $x$ has coordinates: as

\(^4\)The elliptical inner product is often used in statistical pattern recognition in the guise of the Mahalanobis distance (see e.g. Duda et al., 2001; McLachlan, 1992). Both the Euclidean distance and the Mahalanobis distance are special cases of the class of Bregman divergences (e.g. Banerjee et al., 2005) that, in general, do not satisfy the triangle inequality.
many as there are nonnegative integers! Fortunately, a kernel function exists (Elzinga and Wang (2013)) that allows for evaluating $x'M^*y$ without actually constructing $M^*$ or either of the vectors.

4. Spell sequences: handling durations

4.1. Durations of subsequences. At first sight, handling durations in the context of vector representations is easy to conceptualize. For let $T_x(u)$ denote the total duration of some subsequence $u \subseteq x$. A representation $x = (x_1 \ldots)$ of a sequence $x$ is easily defined as

$$x_{r(u)} = \begin{cases} T_x(u) & \text{if } u \subseteq x \\ 0 & \text{otherwise} \end{cases},$$

(18)

leading to an inner product of the form $x'y = \sum_{u \in S(x,y)} T_x(u)T_y(u)$. However, $T_x(u)$ may be ill-defined since when $u$ has multiple embeddings in $x$, it is not evident how $T_x(u)$ should be measured. For example, consider $(x = abac; t_x = (4,3,2,3))$. Then, for $u = ac$, we have that the state $a$ has two different durations and hence the duration $T(u)$ cannot be well-defined. To properly deal with embeddings, we have to formally define the concept as follows (see also Elzinga et al. (2008) or Elzinga and Wang (2013)): we refine our notation through writing $i(u)$ to denote an embedding, i.e. a sequence of position numbers that spells $u$ in $x$. For example, for $x = abac$ and $u = bc$, we would have $i(u) = 2, 4$. For $u = ac$ we have $i(u) = 1, 4$ and $i'(u) = 3, 4$ and therefore we introduce the set $I_x(u)$ of all embeddings of $u$ in $x$. When multiple embeddings do not occur, it is straightforward to define

$$T_x(u) = \sum_{j \in i(u)} t_j,$$

(19)
i.e. as the sum of the lengths of the spells.

When multiple embeddings do exist, we could set $T_x(u)$ equal to the average of the durations of all embeddings:

$$T_x(u) = |x|^{-1}_u \sum_{i(u) \in I_x(u)} \sum_{j \in i(u)} t_j$$

(20)

but this will rarely be an appealing option since it could imply mapping quite different sequences onto the same vector. Alternatively, we might use the durations of all embeddings. So, we define the sum of all durations of all embeddings of a particular subsequence $u$ as

$$T_x(u) = \sum_{i(u) \in I_x(u)} \sum_{j \in i(u)} t_j$$

(21)

which can be interpreted as mapping embeddings, weighed for duration. The inner product resulting from this construction will then have the form

$$x^t y = \sum_i x_i y_i$$

(22)

$$= \sum_{u \in S(x,y)} \left( \sum_{i(u) \in I_x(u)} \sum_{j \in i(u)} t_{j;x} \right) \left( \sum_{i(u) \in I_y(u)} \sum_{j \in i(u)} t_{j;y} \right).$$

(23)

$$= \sum_{u \in S(x,y)} \left( |x|_u \mathcal{T}_x(u) \right) \cdot \left( |y|_u \mathcal{T}_y(u) \right).$$

(24)

Equation (24) hints to an easy interpretation of the representation: vector coordinates are averages of the durations of subsequences, weighed by their embedding frequencies.

Some authors (e.g. Abbott and Hrycak, 1990; Halpin, 2010) have suggested to

---

5In an unpublished manuscript (Elzinga, 2006), it was suggested to use the "Minimum Amount of Shared Time" instead of the inner product of Equation (23). As this alternative does not derive from an inner product, we do not deal with this alternative here.
transform time through a convex or concave function. This can be incorporated in Equation (23) by writing $f(t)$ instead of $t$.

Let us consider the function $f(t) = t^a$. If $a = 0$, no timing information will be used, and the algorithm is strictly equivalent to computing the distance between distinct states sequences. If $0 < a < 1$, then longer spells will weight comparatively less. Halpin (2010) has argued that this is an interesting feature. Finally, if $a > 1$ then small spells will be less important in sequences comparison. For instance, one may be interested in ignoring small unemployment episode while taking into account the longer ones.

4.2. **Practical Considerations.** All of the metrics discussed above have been implemented in the freely available software package TraMineR (Gabadinho et al., 2011) and the required algorithms have been amply described in Elzinga et al. (2008) and in Elzinga and Wang (2013); here we will not deal with algorithmic issues.

TraMineR imposes no practical limitations on the size of the alphabet or the number of sequences in the data set to analyze. However, with $N$ sequences, the number of distinct pairs of sequences amounts to $\binom{N}{2} = N(N - 1)/2$. This implies that the computation time for the distance matrix is roughly quadratic in the number of sequences: doubling the size of the data set will lead to an almost fourfold amount of computation time required. For this reason, Studer (2013) propose a procedure to analyze the data relying only on unique sequences by weighing them accordingly.

Let $n$ denote the length of the sequences involved in a single distance computation; then the calculation of each inner product will be proportional to $n^3$. Hence, the total computation time involved in calculating the full distance matrix for a data set consisting of $N$ sequences of length $n$ will be roughly

---

TraMineR is freely downloadable from http://mephisto.unige.ch/traminer/
proportional to $N^2n^3$. A detailed analysis of the computational complexity of the algorithms involved can be found in Elzinga and Wang (2013). OM requires computation time proportional to only $N^2n^2$ but this difference is not very relevant as the following case illustrates.

McVicar and Anyadike-Danes (2002) published a data set consisting of 712 sequences of school-to-work transitions, each covering 72 months. Calculating a full distance matrix using TraMineR for this data set requires only 1.06 seconds for SVR(spell) while it requires 3.24 seconds for OM. The difference can be explained by the conversion from state sequences to spell sequences. As it greatly reduces the average length $n$ of the sequences (decreasing from 72 to 3.5), computation time reduces by factor of $\frac{72^2}{3.5^2} = 121$. However, with fast modern PC’s, these differences will be insignificant in most applications.

Converting equally long state sequences to spell sequences will normally generate spell sequences of unequal length. However, contrary to OM, working with vector representations does not require the sequences to be equally long. The reason is that the vector representing the shorter of the sequences will have zero-valued coordinates for all subsequences that are longer than the sequence itself. Therefore, multiplying vectors representing sequences of unequal length will only result in zero-valued products of coordinates referring to longer sub-sequences. Hence, there is no theoretical or practical objection whatsoever to calculating distances between sequences of unequal lengths.

5. The general framework: feature vectors

So far, we presented several examples of a very general model for representing sequences as vectors, the coordinates indexed, given the state alphabet, by the (sub-)sequences that are constructible from this alphabet. Given a sequence $x = x_1 \ldots x_n$, we constructed vectors $\mathbf{x} = (x_1, x_2, \ldots)$ such that the coordinate
values are set to
\[
x_{r(u)} = \begin{cases} 
  f(u, x) & \text{if } u \subseteq x \\
  0 & \text{otherwise}
\end{cases}, 
\]
wherein \( f \) is some function that maps the pair \((u, x)\) to some number. In computer science, vectors that contain quantified information about discrete structures like graphs or strings, are called “feature vectors” and sometimes the resulting vector space is called a “feature space”. Here the set of features corresponds to the set of constructible sequences. In Equation (25), the definition of the feature vectors is very general: the “weighting function” \( f \) may operate on both the subsequence indexed by \( r \) and on the sequence in which it is embedded. However, we have seen examples where \( f \) only operates on the subsequence \( u \) and not on \( x \). We illustrate this in Table 2.

Insert Table 2

In the first entry of Table 2, \( f(u, x) = 1 \) whenever \( u \subseteq x \) and regardless of the features \( u \) and regardless of the sequence \( x \): the result is that the standard inner product \( x'y \) equals the count of the number of distinct common subsequences. In the second entry of this table, the subsequences are weighed according to their length and hence \( f \) only operates on \( u \): \( f(u, x) = \sqrt{L(\ell(u))} \). In the third entry, the weighing depends on both \( x \) and \( u \): \( f(u, x) \neq f(u, y) \) precisely when \(|x|_u \neq |y|_u\). The same is true for duration weighing: both \(|x|_u\) and \( T_x(u) \) depend on both \( u \) and on \( x \).

In the last two entries, we mention two kinds of weighing not dealt with in this paper. Weighing according to gap-width is relevant when one considers common subsequences with big time-gaps between the states as less relevant.
Weighing according to the state composition of the subsequences might be relevant when the occurrence of particular states is more salient than the occurrence of other states. The point here is that any kind of weighing can be accommodated in the general representation and it can be applied as long as we can find algorithms that allow us to evaluate the inner products of the vectors. Furthermore, it is important to stress the fact that any number of these weightings may be applied simultaneously, again provided suitable algorithms are available.

What is not shown in Table 2 is that each of these weightings can be applied with or without soft-matching of states, i.e. with either an inner product of the form $x'y$ or of the form $x'M^*y$ as long as $M^*$ is positive semi-definite. So, relying on a subsequence vector representation (“SVR” for short) allows for an enormous versatility in weighing features, warping time, applying soft-matching and dealing with sequences of unequal lengths. Furthermore, the interpretation of the results of well-known methods in sequence analysis is made easier. For instance, using “Ward”-clustering with such a metric is equivalent to finding clusters minimizing the residual variance of the features, i.e. minimizing the variability of the subsequences. Using discrepancy analysis is equivalent to running a multivariate analysis of variance (MANOVA) in which the dependent variables are the features (i.e. the subsequences).

In the next two sections, we will compare SVR-metrics with OM. In particular, we will use weighing of subsequence lengths by varying the parameter $a$ in $L(\ell(u)) = \ell(u)^a$. If $a = 0$, no length-weighing is applied and if $a > 0$, more weight is given to longer subsequences and the resulting SVR-metric will be more sensitive to ordering. Furthermore, we will use time-transforms of the form $f(t) = t^b$ previously introduced. The various metrics to be used are listed in Table 3.
6. Assessing metric sensitivity

Common *order* of states is the basic property that defines similarity between sequences as temporal successions of states or events (Elzinga, 2003). However, common order is not the only angle from which to look at sequence similarity. Another important aspect is *duration*. For example, \( a \ldots ab \) and \( abbb \ldots b \) are quite different sequences, although the order in which \( a \) and \( b \) appear is the same. For instance, a difference in the duration of a poverty spell may have a huge impact on the rest of the life course, because poverty may act as a trap (Pollak, 2010). Finally, *timing* of events can be the feature of interest as Lesnard (2010) and Rousset and Giret (2007) argued. For example, work during day-time is socially quite different from work during a night-shift and early unemployment may have quite a different effect than unemployment that occurs later in the career (Mooi-Reci, 2012). Therefore, we will compare OM to different configurations of the newly introduced distance measures and evaluate how sensitive these measures are to differences in state order, in state duration and in state timing through using simulated, short sequences with controlled variations on these facets. A more detailed presentation of this simulation framework is available in Studer (2012).

To evaluate these sensitivities, we proceed as follows. We generate two groups of sequences that differ in only one of the facets: in ordering, in timing or in duration. We then evaluate the ability of each distance measure to discriminate between these two groups using a Discrepancy Analysis. This analysis evaluates the strength of the association between the sequences as described by
a distance measure and a partition (here, our two groups). This association is measured using a pseudo-$R^2$ defined as

$$0 \leq R^2 = \frac{SS_B}{SS_T} \leq 1$$

wherein $SS_B$ and $SS_T$ are sums of distances (for details see Studer et al., 2011). Given a fixed set of sequences, the size of this $R^2$ will depend on the distance metric used and the relative size of $R^2$ can thus be interpreted as a measure of how well a particular metric discriminates between groups of sequences. If this pseudo-$R^2$ is close to one, the distance measure is very sensitive to the facet on which the two groups differ. On the other hand, if the pseudo-$R^2$ is close to zero, the distance measure is relatively insensitive to the pertaining facet.

In order to get stable results, one million sequences were generated in each group of sequences. Each simulation is repeated one thousand times and the results proposed here show the average pseudo-$R^2$ over all runs. Confidence intervals are not plotted because standard errors are extremely small (maximum standard error of $1.9 \cdot 10^{-4}$).

In the present context, we ran three different types of simulations as summarized in Table 4. Each of these separately evaluates the sensitivity of the metric to perturbations of one of the facets previously introduced: ordering, timing or duration. Below, we present the details of each of these simulations and discuss the results.

---

Insert Table 4

---

7Recently, Bonetti et al. (2013) published a similar approach under the acronym ANODI.
8In order to reduce computation time, the simulations were computed on unique sequences only by weighting unique sequences by their frequencies (see Studer et al., 2011, for weighting formulae). The sequence aggregation procedure is fully described in Studer (2013) and can also be used for clustering or any other distance based sequences analysis.
6.1. **Ordering and State Proximities.** For each simulation, we created two groups of spell sequences: \((x, t_x)\) and \((y, t_y)\) with distinct but fixed patterns \(x\) and \(y\). The duration vectors were randomly generated with the only restriction that the sum of the durations was fixed to 20 units of time, i.e. \(1't_x = 20 = 1't_y\) for all sequences. For example, with \(x = ca\) and \(y = cb\), the simulated data set would look like

\[
\begin{align*}
[ca, (3, 17)] \\
[ca, (12, 8)] \\
&\vdots \\
[cb, (6, 14)] \\
[cb, (15, 5)] \\
&\vdots
\end{align*}
\]

A metric that is very sensitive to differences in order or pattern will easily separate the two groups by generating a high value of \(R^2\), whereas a metric that is less sensitive to pattern will generate a lower \(R^2\) because it will generate only a small distance between e.g. \([ca, (19, 1)]\) and \([cb, (17, 3)]\) because the time spent in \(c\) is long in both sequences.

State proximities strongly affect the ordering. As a first example, consider again the spell sequences \(x = (ca, t_x)\) and \(y = (cb, t_y)\), all of length 20 and random durations. Now suppose that \(m(a, b) = 1\), implying, in fact, that the states \(a\) and \(b\) are indistinguishable. As a result, \(x\)- and \(y\)-sequences cannot be separated, and hence we expect a discrepancy analysis to produce \(R^2 = 0\). If, on the other hand, \(m(a, b) = 0\), \(x\)- and \(y\)-sequences are easily separable, and
we would expect $R^2$ to be close to 1.

Summarizing: we generate a set of sequences with two generators, calculate distances with one of the metrics from Table 3, calculate $R^2$ and repeat this for different values of $m(a, b)$ and plot the values of $R^2$ against the values of $m(b, c)$. The results of these exercises are shown in Figure 3.\(^9\)

\[\text{Insert Figure 3}\]

The results when $m(a, b) = 0$ show the difference in sensitivity to ordering. SVR(spell) variants are the most sensitive to ordering. As expected, this sensitivity increases with subsequence length weighting ($a = 1$), and it decreases when spell durations are squared ($b = 2$), the latter having a stronger effect. OM is much less sensitive to ordering, and NMS (SVR with replicated states) fails to identify a difference in the orderings. Similar results pertaining to OM and NMS were already discussed in Studer (2012). Since we are measuring relative sensitivity, the distance measure could be more sensitive to other facets of sequence comparisons.

The shapes of the curves convincingly show the effect of soft-matching. In a qualitative sense, the SVR-metrics show the same behavior as the OM-metric and they all behave as expected. In all cases, the $R^2$ are maximal when $m(a, b) = 0$ and the $R^2$ tends towards zero when $m(a, b) = 1$. SVR(spell) variants react more pronounced than OM, because (soft-)matching subsequences are proportionally weighed by their durations. On the contrary, NMS underperforms, because the number of (soft-)matching subsequences $x'y'$ is always small relative to the quantities $x'x$ and $y'y$.

\(^9\)Since the OM-metric cannot be varied with a matching coefficient, we varied the substitution costs in the above calculations by setting these costs equal to $2 - 2m(a, b)$. 
6.2. **Timing.** Timing simulations follow the same logic as the one for ordering. Patterns and durations are random in both groups, but the spell in the state \( b \) always starts at time \( 2 + t \). We set \( t = 0 \) in the first group and progressively change \( t \) in the second one. Here again, we are measuring the relative sensitivity of the metrics to timing.

The first panel of Figure 4 presents the evolution of the \( R^2 \) when the time difference between both groups increases for each of these simulations. NMS is the most sensitive to timing, but the slope of the curve decreases, meaning that it comparatively fails to discriminate between very high difference of timing. SVR(spell, \( a=0, b=2 \)) performs very well while the other show very similar results.

---

Insert Figure 4

---

6.3. **Duration.** We used the same strategy again for the duration simulations. Patterns and timing are random, while the duration of the second spell (in the state \( b \)) is set to 4 in the first group. In the second group, this duration is set to \( 4 + t \) and \( t \) is progressively changed in the interval \( t \in 2\ldots8 \). Here again, we are measuring the relative sensitivity to duration of the metrics, since the metric may still be more sensitive to other facets (such as ordering, for instance).

The second panel of Figure 4 presents the results for duration-related simulations. By far, OM is the most sensitive to duration. NMS and SVR(spell, \( a=0, b=2 \)) present, here again, an intermediary position. Regarding SVR(spell), we
note that $b$ parameter is strongly linked with sensitivity to timing and duration. The $a$ parameter lowers the sensitivity too, but the effect is not very pronounced.

6.4. Conclusion. According to our simulations, the distance measures are sensitive to different facets. SVR(spell) variants are most sensitive to differences in ordering, NMS is most sensitive to timing and OM to duration. This means that the choice of a distance measure always has to be justified in the context of the application in which it is applied.

These simulations allowed us to measure the effects of the SVR(spell) parameters and to demonstrate that they behave as expected. The $a$ parameter raises the sensitivity to ordering and lowers sensitivity to timing and duration. On the contrary, raising $b$ leads to a distance measure that is more sensitive to timing and duration and less to ordering.

We now turn to an application of these SVR-metrics to real data in order to highlight the contributions of the newly introduced distance measures.

7. An application to family formation

7.1. Data and Distances. In this section, we apply the different configurations of the SVR-metrics to well-known data and compare the results with those obtained when applying OM and NMS to the same data. The data were first presented in Müller et al. (2008).

Briefly, these data represent family formation trajectories of Swiss individuals who were at least 30 years old at the time of the survey$^{10}$. One of the goals of this study was to highlight the change of the social norms constraining these trajectories. The states in the sequences were built using a combination of

$^{10}$The data were collected through the Swiss Household Panel (www.swisspanel.ch) in 2002 using a retrospective biographical survey.
four distinct events: Leaving home, Marriage, having a first Child and Divorce. For the sake of simplicity, some very rare states were merged resulting in eight possible states. An individual is in the state “P” (living with Parent) if no event has occurred, in the state “L” if the event ”Left parental home” occurred, in the state “LM” for “Left and Married”, and “LMC” for “Left, Married and with a first Child”. Similarly, state “M” is for an individual who just Married (without leaving parental home), and so on. Finally, state “D” is for all individuals who have married and divorced (without making difference for having left the parental home and/or having or not having children).

To determine the substitution costs needed for the calculation of an OM-distance matrix, we proceeded as follows. First, we created a 4-dimensional vector for each state; the coordinates corresponding to the events shown in Table 5 by assigning 0 to “no”, 1 to “yes” and 0.5 to “yes/no”. Then, we calculated the Manhattan-distance between all pairs of states and normalized these distances to the maximum distance found (= 3). This amounts to assigning a value of $\frac{1}{3}$ to each coordinate distance. These costs can thus be interpreted as the difference in the events that already happened. For example, the substitution costs between states “P” having coordinates $(0, 0, 0, 0)$ and “LM” with coordinates $(1, 1, 0, 0)$ equals $\frac{|1-0|+|1-0|+|0-0|+|0-0|}{3} = \frac{2}{3}$. Since the SVR-metric needs proximities instead of costs, we set the soft-matching coefficients to 1 minus the OM-cost. Taking the same example, the proximity between states “P” and “LM” is thus $1 - \frac{2}{3} = \frac{1}{3}$.

In order to compare the results obtained by using different metrics, we calculated the different distance matrices using the proximities (or costs) defined
above: SVR(sp, b=1) (SVR based on spells), SVR(sp, b=2) (SVR based on spells, squared durations), SVR(sp, a=1) (SVR based on spells with subsequence weighting) and the OM-distance. In order to highlight the effect of proximities, we also added the distance SVR(sp, b=1, c), the SVR(sp, b=1) distance computed using constant differences (i.e. a similarity of zero between all states). Finally, we included the NMS distance as defined by Elzinga (2003), i.e. with constant cost, in order to highlight the the distinctive features of the newly proposed metrics. These are the same metrics as used in the simulations (see Table 3), but with shortened names in order to generate useful plots.

7.2. Distance Disagreements. To investigate the differences between the various metrics, we started by looking for pairs of sequences where the different metrics generate very different distances. Thereto, we first standardized the metrics in order to get rid of different distance units; for each metric, say metric $a$, we divided all distances by the standard-deviation of the distances as generated by that metric. Thus, we calculated, for all pairs of sequences, $d'_a(x, y) = \frac{d_a(x, y)}{s_{d_a}}$, with the effect of creating dimensionless or unit-free distances $d'_a$ that can be compared across metrics.\footnote{This operation has no effect on any distance analysis method.} Next, for two metrics, say $a$ and $b$, we looked at the pair of sequences $(x, y)$ for which the difference $d'_a(x, y) - d'_b(x, y) = \Delta(a, b)$ is maximal. Since $\Delta(a, b)$ may be negative, we also looked for pairs of sequences $(x', y')$ for which $\Delta(b, a) = d'_b(x', y') - d'_a(x', y')$ is maximal. This procedure generates a matrix of pairs of sequences and values of $\Delta$ as shown in Table 6. The $\Delta$ is computed by subtracting the distance given in column to the one given in the row. Thus, in each cell, we find pairs of sequences for which the pertaining metrics generate extremely different distances.
Let us discuss an example by looking at the strongest disagreement between standardized OM and standardized SVR(sp, b=1). In the first column fourth row, we have “OM - SVR(sp, b=1) = 4.74” for the comparison of the sequences $P^{15} - LMC^1$ and $P^2 - LMC^{14}$. According to OM, these sequences are far away because OM is strongly related to the total time spent in each state, which are very different in this case. According to SVR(sp, b=1), these sequences are close, because SVR(sp, b=1) is more linked to the order of the states, which is the same in both sequences. We can also have a look at the reverse, that is when SVR(sp, b=1) is greater than the OM-distance. This is found when comparing the sequence $P^1 - L^5 - LM^1 - LMC^8 - D^1$ and $L^8 - LMC^8$. According to SVR(sp, b=1), these sequences are far away, because the ordering of the states is different. According to OM, the sequences are close because the time spent in states $L$ and $LMC$ are more or less the same.

We can identify the contribution of soft matching coefficients by looking for the differences between SVR(sp, b=1) and SVR(sp, b=1, c) (SVR(sp, b=1) with or without soft matching coefficients). Using states proximities, the distance between $P^2 - L^7 - LMC^6 - D^1$ and $P^7 - LM^4 - LMC^3 - D^2$ is lower than using constant proximities, because states $L$ and $LM$ are close according to our soft matching coefficients. Accounting for states proximities allows to consider that sequences $P^5 - L^2 - LM^1 - LMC^3 - D^5$ and $P^{16}$ are comparatively more distant. This is exactly what soft matching coefficients are intended to do.

We can also identify the contribution of SVR distances parameters such as time transform by looking at the differences between SVR(sp, b=1) and SVR(sp, b=2). As expected and confirming our simulation results, SVR(sp, b=2) is
more sensitive to time spent in each state whereas SVR(sp, b=1) is more sensitive to ordering. Subsequences length weighting [SVR(sp, a=1)] has the effect of weighting the comparison of states in sequences containing many different spells. As a result, $P^7 - L^1 - LM^1 - LMC^3 - D^4$ and $P^8 - M^2 - LM^1 - LMC^1 - D^4$ are considered to be farthest by SVR(sp, a=1). On the contrary, SVR(sp, b=1) is comparatively more sensitive to difference in short spell sequences ($L^{15} - LC^1$ and $P^2 - M^{14}$). However, in both cases, the differences are small (less than 1).

Finally, let us look at the difference between SVR-metric and NMS (Elzinga, 2003). Since NMS only accepts constant state proximities, we will compare distances SVR(sp, b=1, c) and NMS$^{12}$. According to NMS, sequences $L^{16}$ and $P^{16}$ are farthest, because NMS will count many different subsequences while SVR(sp, b=1, c) will only consider one subsequences in each sequences. On the contrary, SVR(sp, b=1, c) is comparatively more sensitive to difference in long spell sequences ($P^8 - M^2 - LM^1 - LMC^1 - D^4$ and $P^3 - L^7 - LC^1 - LMC^2 - D^3$).

The analysis of distance disagreement confirms the results of the simulations. SVR(sp) variants are the most sensitive to ordering while OM distance is strongly linked with the time spent in a state. This analysis also highlights more precisely the effect of the SVR parameters. While $b$ increases the sensitivity to duration and timing, $a$ makes the distance measure more sensitive to the ordering of complex sequences. Finally, this analysis has confirmed that soft matching has the desired effect. It highlights the main judgment differences between distances measures. However, in practice, all differences, even the smallest ones, are taken into account. We slightly varied the state proximities as discussed in the present paper and found no results that were

---

$^{12}$The framework presented here allows to use state proximities in NMS.
unpredicted; we do not report on these results since they are too limited to warrant firm conclusions on the sensitivity of the methods to small perturbations of the state proximities. However, a more thorough investigation of the sensitivity issue would unduly elongate this already length paper.

7.3. Clustering. We used the PAM-algorithm (Kaufman and Rousseeuw, 1990) to cluster the sequences on the basis of the four distance matrices using the sampling weights. To get an indication of the optimal number of clusters, we calculated all solutions with the number of clusters varying between 3 and 20. Table 7 summarizes the results. Both the ASW- and the HC-index are dimensionless measures, each depending on a ratio (of differences) of distances, and therefore, these indices can be used to compare partitions based upon different distance-matrices. They can be interpreted as the capacity of a clustering method to match the structure of the data, the structure being defined by the features of each metric. The computations were carried out with the WeightedCluster library (Studer, 2013). SVR-based clusterings usually identify more clusters and the best clustering quality is found with SVR(sp, a=1).

Table 8 presents the medoids of the clusters obtained using this optimal number of groups. SVR-metrics provide very similar clustering (Cramer’s $V \geq 0.89$ between these solutions). SVR(sp, b=1) and SVR(sp, a=1) identify two small groups of trajectories leading to divorce that are not identified with other distances. SVR(sp, a=1) also finds a small group of non-married parents (ending in state $LC$). These are important features, since all of these patterns may
have gained in importance during the 20th century. If divorce is negligible, it should not be used to build the sequences. Otherwise, it should be included in subsequent analyses. SVR(sp, b=2) makes some distinctions between sequences according to the time spend in each state of the pattern $P - L$.

Confirming the sensitivity to timing highlighted by the simulations, NMS makes several distinctions according to the timing of transitions. However, confirming the results presented by Aisenbrey and Fasang (2010), all complex sequences are regrouped in a big, quite heterogeneous “residual” cluster (“$P^3 - L^4 - LM^3 - LMC^4 - D^2$” that contains 37% of the sequences).

Next, we closely scrutinize the difference between the clustering results for OM and SVR(sp, b=1) through visually rendering the clusters.

7.4. **Cluster visualization.** To visually render the clusters, we will use parallel-coordinate plots, chronograms and sequence-index plots. As many readers may not be familiar with parallel-coordinate plots (for short: PC-plots), we first spend a few lines on them (see also Bürgin and Ritschard, 2014; Bürgin et al., 2012; Inselberg, 2009).

A PC-plot renders multivariate objects on a flat surface by first drawing as many vertical lines as there are variables or dimensions, each of which may have a different scale. Individual objects are depicted as a line, drawn in left-to-right direction, crossing the vertical (parallel) lines at the appropriate height. Often, the thickness of the object-representing lines is proportional to the number of objects that share the same coordinates. A toy-example of a PC-plot is shown in Figure 5.
Here, we use the PC-plots to render the sequences by the order of the events, ignoring durations. To attain this, we use as many identical, parallel scales as there are events (states) in the individual sequences. Hence, an individual’s position on the first of the scales corresponds to the first event, her position on the second parallel scale corresponds to the second event, etc.

Figure 6 presents the PC-plots of the sequences plotted according to the SVR(sp, b=1) clustering. Let us discuss some examples in order to illustrate the interpretation of these plots. In the plot called “P-LM”, the brown line indicates one of the patterns of the four events. It starts at position 1 in state “P” (living with parents) before going to the events “left parental home” and “marriage” at position 2. Since “left parental home” and “marriage” happen simultaneously, the line is vertical. In the group called “P-L-LM”, the green line indicates that the pattern is “P”, leaving at position 2) and marrying later on (position 3). In both plots, the size of the squares and the width of the lines are plotted according to the relative frequency of the pattern.

Using these plots, we can see that the SVR(sp, b=1) clustering is very homogeneous according to the ordering of the underlying events. Only the clusters leading to divorce group different patterns, but they all end with divorce. The clusters distinguish the sequences according to the synchronization of events,
notably marriage and leaving home. In a first set of clusters, leaving home is experienced before marriage, while in another set these events occur simultaneously. These are important distinctions; Billari et al. (2001) argued that the simultaneity of marriage and leaving home should be interpreted as one distinct state.

Figure 7 presents the chronograms of the six clusters found from the OM-distances. From these chronograms, the clusters seem easy to interpret. Indeed they are, but only on the basis of the time spent in the states and not on the basis of the orderings of the underlying events. Figure 8 presents the PC-plots of the same clustering. The underlying orderings of the events are very diverse in each cluster. For instance, looking at the cluster called “Late LMC”, at least four patterns can be identified (events in parenthesis occurs simultaneously): P-(LM)-C (in rose), P-(LMC) (dark-blue), (PLMC) (yellow), P-L-(MC) (green). Using the chronogram, we are tempted to call the first group “Staying with parents”, because the mean time spent in state “P” is large. However, the PC-plots show that many distinct patterns are grouped here.

The wide use of chronograms and index plots may be one of the reasons of the popularity of OM. As we have shown with our simulations and through this example, OM is strongly linked with duration differences. The is shown
in chronograms and index plots too, because the area plotted in single colour depends of the total time spent in the associated state.

Comparing both clustering solutions, using OM leads to some distinctions according to time spent in each state while SVR(sp, b=1) is strongly linked with the ordering of the underlying events.

7.5. **Metrics and the Evolution of family trajectories.** If there would be an evolution of family trajectories, we would expect to see the size of clusters change over time in a systematic way (see e.g. Elzinga and Liefbroer, 2007). Here we evaluate these changes as revealed by both clustering on the basis of OM-distances as well as on the basis of SVR(sp,b=1). In Tables 9 and 10, we present the relative distributions of cluster membership per cohort, for OM- and SVR-distances respectively. The association is highly significant in both cases but stronger for SVR-based clustering (Cramer’s $V = .193$) than for OM-based clustering (Cramer’s $V = .147$).

More interesting is the question if and what qualitative differences show up when we study the evolution of cohorts through OM or through an SVR-based metric.

Using OM, the evolution is seems dominated by state duration changes. Older cohorts were staying longer with their parents (clusters $P-M$, $P$ and “Late LMC”) while younger cohorts leave the parental home earlier. Moreover, the last cohort seems to distinguish itself by not marrying and not having children. However, caution is needed, because of the heterogeneity of the orderings.

Insert Table 9
Clustering SVR-distances provides for an alternative view on this evolution by highlighting changes in the ordering of the events. Older cohorts stand out through the synchronicity of leaving the parental home and marriage. These two events were frequently occurring simultaneously but this is much less frequent in the youngest cohorts. This “de-synchronization” has been interpreted as the result of the raise of non-marital union in Switzerland and the introduction of a new intermediary stage of “partial independence” in the road toward autonomy (Thomsin et al., 2004). Contrary to OM, here the latest cohort does not distinguish by not marrying nor having children, but by different patterns leading to these situations.

Clearly, in this analysis of Swiss family formation sequences, the SVR(sp,b=1)-based metric has provided new insights through revealing the underlying ordering of the events. Optimal matching leads to interesting results when we are interested in the durations spent in each state.

8. Conclusion and Discussion

We motivated this paper by pointing at the poor performance of the OM-metric with respect to a basic property of sequences: the order of the states or events involved. OM is not very sensitive to differences in the sequencing of the pertaining states. This lack of sensitivity is nicely demonstrated through the application described in the previous section through the PC-plots that show very different orderings of the underlying events within clusters. This is not to say that OM cannot be a useful metric: it is useful when state durations are more important than state ordering. This too is shown in the chronograms of the previous section.
According to our simulations, the NMS-based metric is mostly sensitive to differences in timing. However, in the application presented, one of the big NMS cluster regroups all “complex” sequences which is not very meaningful. Such a phenomenon was already noted by Aisenbrey and Fasang (2010).

We presented a very flexible, versatile metric that does well when ordering of states is the key-issue. This too was demonstrated in the previous section and in the simulations presented in Section 6. Contrary to OM, the SVR-based metrics are less sensitive to duration and more sensitive to the sequencing, the ordering of the states. The exact behaviour of the metric can be adjusted using two parameters. The exponential transformation of time (the $b$ parameter) raises the sensitivity to duration and timing, while subsequence length weighting ($a$ parameter) makes the distance measure more sensitive to the ordering of complex sequences.

Our simulations and application have highlighted the difference between OM and SVR metrics. This can be used to justify the use of one or the other metric and to further interpret differences in results produced by different metrics. Finally, it also helps to interpret the structure of the data. If SVR-based distances produce better results, it might be because the data is more structured according to ordering than according to state durations. Therefore, we believe that the SVR-family is a useful alternative to alignment-based methods.


Table 1. Illustration of the metric properties of the OM standard edit cost matrix. The reader verifies that, in the left hand matrix, \( c(\cdot, \cdot) \) satisfies the axioms D1 - D4. For example, we have that \( c(w, y) \leq c(w, x) + c(x, y) \) for every \( x \). However, perturbations of this matrix may easily lead to violations of the triangle inequality D4. This is shown in the right matrix, where we have \( c(x, z) > c(x, y) + c(y, z) \).

<table>
<thead>
<tr>
<th></th>
<th>( \lambda )</th>
<th>( w )</th>
<th>( x )</th>
<th>( y )</th>
<th>( z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( w )</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>( x )</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>( y )</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>( z )</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
**Table 2.** Weighing functions for feature vectors. The middle column shows the evaluation of $f(u, x)$ for the kind of weighing as indicated in the leftmost column. The rightmost column shows where this kind of weighing is discussed in the main text. The last two kinds of weighing are not discussed in the present paper.

<table>
<thead>
<tr>
<th>feature weighing</th>
<th>$f(u, x)$ if $u \subseteq x$</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>1</td>
<td>2.2</td>
</tr>
<tr>
<td>length</td>
<td>$\sqrt{L(\ell(u))}$</td>
<td>2.4</td>
</tr>
<tr>
<td>embedding frequency</td>
<td>$</td>
<td>x</td>
</tr>
<tr>
<td>duration</td>
<td>$</td>
<td>x</td>
</tr>
<tr>
<td>gap width</td>
<td>(see Elzinga and Wang, 2013)</td>
<td>–</td>
</tr>
<tr>
<td>state composition</td>
<td>(see Elzinga and Wang, 2013)</td>
<td>–</td>
</tr>
</tbody>
</table>
Table 3. Metrics used in assessing sensitivity; all of them weigh for embedding frequency.

<table>
<thead>
<tr>
<th>acronym</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMS</td>
<td>NMS distance as defined in Elzinga (2003, 2005), i.e. duration coded as replicated states and no subsequence length weighting. The present article extends this metric by allowing states proximities and durations.</td>
</tr>
<tr>
<td>SVR(emb, spell, $a = 0, b = 1$)</td>
<td>Spell sequences, no subsequence length weighting, no time transform.</td>
</tr>
<tr>
<td>SVR(emb, spell, $a = 1, b = 1$)</td>
<td>Spell sequences, subsequence length weighting using $L(ℓ(u)) = ℓ(u)$, no time transform.</td>
</tr>
<tr>
<td>SVR(emb, spell, $a = 0, b = 2$)</td>
<td>Spell sequences, no subsequence length weighting, time transform using $f(t) = t^2$.</td>
</tr>
<tr>
<td>OM</td>
<td>Standard optimal matching algorithm, indel set as half the maximum substitution cost.</td>
</tr>
</tbody>
</table>
Table 4. Patterns, onset and duration variations used in assessing the sensitivity to perturbations of ordering, timing and duration. Total duration of all patterns is restricted to 20 units of time.

<table>
<thead>
<tr>
<th>Simulation</th>
<th>Description</th>
<th>Group 1</th>
<th>Group 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordering</td>
<td>Time spent in each state is random</td>
<td>ca</td>
<td>cb</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ac</td>
<td>cb</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cac</td>
<td>cbc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>caca</td>
<td>cbc</td>
</tr>
<tr>
<td>Timing</td>
<td>Random patterns abc or cba. b starts at time $2 + t$. $t = 0$ $t \in 2 \ldots 8$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duration</td>
<td>Random patterns abc or cba. Duration of b equals $4 + t$. $t = 0$ $t \in 2 \ldots 8$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5. State definitions of family formation trajectories from the Swiss Household Panel.

<table>
<thead>
<tr>
<th>States</th>
<th>Events</th>
<th>Leaving home</th>
<th>First marriage</th>
<th>First child</th>
<th>First divorce</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>L</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>M</td>
<td>no</td>
<td>yes</td>
<td>yes/no</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>LM</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>LC</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>LMC</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>yes/no</td>
<td>yes</td>
<td>yes/no</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>
Table 6. Analysis of biggest standardised distance differences. The Table below specifies the distances and the pertaining sequence pairs.

<table>
<thead>
<tr>
<th></th>
<th>SVR(sp, b=1)</th>
<th>SVR(sp, b=2)</th>
<th>SVR(sp, a=1)</th>
<th>OM</th>
<th>SVR(sp, b=1, c)</th>
<th>NMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVR(sp, b=1)</td>
<td></td>
<td>∆ = 2.21</td>
<td></td>
<td></td>
<td>∆ = 9.99</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>L3-LMC7</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P3-L1-LMC1-LMC4-D1</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P2-M14</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P3-L1-LMC1-LMC4-D1</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td>SVR(sp, b=2)</td>
<td></td>
<td>∆ = 3.69</td>
<td></td>
<td></td>
<td>∆ = 4.69</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P1-L1-LMC1-LMC1</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P4-L3-LMC4-LMC4</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P2-M14</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td>SVR(sp, a=1)</td>
<td></td>
<td>∆ = 0.21</td>
<td></td>
<td></td>
<td>∆ = 4.74</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P7-L1-LMC1-LMC4-D1</td>
<td></td>
<td></td>
<td>P1-L3-LMC3-LMC4-D1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P8-M2-LMC1-LMC1-D1</td>
<td></td>
<td></td>
<td>P1-L3-LMC3-LMC4-D1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P6-LMC8</td>
<td></td>
<td></td>
<td>P5-L2-LMC1-LMC3-D1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P1-L3-LMC1-LMC4-D1</td>
<td></td>
<td></td>
<td>P5-L2-LMC1-LMC3-D1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P5-C11</td>
<td></td>
<td></td>
<td>P1-L3-LMC3-LMC4-D1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P15-LMC1</td>
<td></td>
<td></td>
<td>P1-L3-LMC3-LMC4-D1</td>
<td></td>
</tr>
<tr>
<td>OM</td>
<td></td>
<td>∆ = 4.74</td>
<td></td>
<td></td>
<td>∆ = 4.24</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P2-LMC14</td>
<td></td>
<td></td>
<td>P1-L3-LMC10-LMC2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P15-LMC1</td>
<td></td>
<td></td>
<td>P4-M2-LMC1-D9</td>
<td></td>
</tr>
<tr>
<td>SVR(sp, b=1, c)</td>
<td></td>
<td>∆ = 2.27</td>
<td></td>
<td></td>
<td>∆ = 4.51</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P2-L7-LMC6-D1</td>
<td></td>
<td></td>
<td>P3-L1-LMC11-D2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P7-LM3-LMC9-D1</td>
<td></td>
<td></td>
<td>P3-L1-LMC11-D2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P4-L3-LMC5-LMC3-D1</td>
<td></td>
<td></td>
<td>P3-L1-LMC11-D2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P10-LM3-LMC1-D1</td>
<td></td>
<td></td>
<td>P3-L1-LMC11-D2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P8-M2-LMC1-D9</td>
<td></td>
<td></td>
<td>P3-L1-LMC11-D2</td>
<td></td>
</tr>
<tr>
<td>NMS</td>
<td></td>
<td>∆ = 9.99</td>
<td></td>
<td></td>
<td>∆ = 8.99</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>L16</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P16</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>L16</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P16</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>L16</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P16</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P16</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P16</td>
<td></td>
<td></td>
<td>L16</td>
<td></td>
</tr>
</tbody>
</table>
Table 7. Clustering quality measured through Average Silhouette Width (ASW, to be maximized, Kaufman and Rousseeuw (1990)) and the HC-index (HC, to be minimized, Hubert and Levin (1976)) with various metrics as indicated below. \( nc \) denote the optimal number of clusters for each of the metrics used.

<table>
<thead>
<tr>
<th>metric</th>
<th>( nc )</th>
<th>ASW</th>
<th>HC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVR(sp, b=1)</td>
<td>11</td>
<td>.55</td>
<td>.05</td>
</tr>
<tr>
<td>SVR(sp, b=1, c)</td>
<td>10</td>
<td>.53</td>
<td>.08</td>
</tr>
<tr>
<td>SVR(sp, b=2)</td>
<td>10</td>
<td>.42</td>
<td>.09</td>
</tr>
<tr>
<td>SVR(sp, a=1)</td>
<td>12</td>
<td>.65</td>
<td>.02</td>
</tr>
<tr>
<td>OM</td>
<td>6</td>
<td>.37</td>
<td>.07</td>
</tr>
<tr>
<td>NMS</td>
<td>17</td>
<td>.35</td>
<td>.07</td>
</tr>
</tbody>
</table>
Table 8. Medoids of the clusters found with different distance metrics with varying state similarities. Durations of states are indicated as superscripts of the state-acronyms. Clusters have been placed on the same row when their state-orders match. Relative clustersizes ($N = 4191$) are shown in the column labeled “%”.

<table>
<thead>
<tr>
<th>SVR(sp, b=1)</th>
<th>%</th>
<th>SVR(sp, b=2)</th>
<th>%</th>
<th>SVR(sp, a=1)</th>
<th>%</th>
<th>OM</th>
<th>%</th>
<th>SVR(sp, b=1, c)</th>
<th>%</th>
<th>NMS</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>P$^{16}$</td>
<td>9.4</td>
<td>P$^{16}$</td>
<td>8.9</td>
<td>P$^{16}$</td>
<td>9.4</td>
<td>P$^{16}$</td>
<td>19.0</td>
<td>P$^{16}$</td>
<td>9.7</td>
<td>P$^{16}$</td>
<td>9.0</td>
</tr>
<tr>
<td>P$^{9}$-M$^7$</td>
<td>9.7</td>
<td>P$^{9}$-M$^7$</td>
<td>10.3</td>
<td>P$^{9}$-M$^7$</td>
<td>9.7</td>
<td>P$^{9}$-M$^8$</td>
<td>8.0</td>
<td>P$^{9}$-M$^7$</td>
<td>10.0</td>
<td>P$^{9}$-M$^7$</td>
<td>4.0</td>
</tr>
<tr>
<td>P$^{7}$-L$^9$</td>
<td>8.8</td>
<td>P$^{10}$-L$^6$</td>
<td>8.3</td>
<td>P$^{7}$-L$^9$</td>
<td>18.7</td>
<td>P$^{5}$-L$^{11}$</td>
<td>21.9</td>
<td>P$^{10}$-L$^6$</td>
<td>8.5</td>
<td>P$^{10}$-L$^6$</td>
<td>4.2</td>
</tr>
<tr>
<td>P$^{10}$-LM$^6$</td>
<td>7.1</td>
<td>P$^{10}$-LM$^6$</td>
<td>7.8</td>
<td>P$^{10}$-LM$^6$</td>
<td>7.2</td>
<td>P$^{10}$-LM$^6$</td>
<td>7.8</td>
<td>P$^{10}$-LM$^6$</td>
<td>1.9</td>
<td>P$^{10}$-LM$^6$</td>
<td>2.7</td>
</tr>
<tr>
<td>P$^{8}$-LMC$^8$</td>
<td>7.9</td>
<td>P$^{8}$-LMC$^8$</td>
<td>7.9</td>
<td>P$^{8}$-LMC$^8$</td>
<td>8.0</td>
<td>P$^{8}$-LMC$^8$</td>
<td>8.5</td>
<td>P$^{8}$-LMC$^8$</td>
<td>2.8</td>
<td>P$^{8}$-LMC$^8$</td>
<td>4.3</td>
</tr>
<tr>
<td>P$^{7}$-LM$^3$-D$^6$</td>
<td>1.9</td>
<td>P$^{7}$-LM$^3$-D$^6$</td>
<td>P$^{6}$-LM$^2$-LC$^5$</td>
<td>1.1</td>
<td>P$^{7}$-LM$^3$-D$^6$</td>
<td>2.0</td>
<td>P$^{7}$-LM$^3$-D$^6$</td>
<td>P$^{6}$-LM$^2$-LC$^5$</td>
<td>P$^{6}$-LM$^2$-LC$^5$</td>
<td>P$^{6}$-LM$^2$-LC$^5$</td>
<td>11.6</td>
</tr>
<tr>
<td>P$^{6}$-LM$^2$-LC$^5$</td>
<td>8.6</td>
<td>P$^{4}$-LM$^2$-LC$^6$</td>
<td>8.3</td>
<td>P$^{5}$-LM$^2$-LC$^5$</td>
<td>7.6</td>
<td>P$^{6}$-LM$^2$-LM$^4$</td>
<td>P$^{5}$-LM$^2$-LC$^6$</td>
<td>P$^{5}$-LM$^2$-LC$^6$</td>
<td>P$^{5}$-LM$^2$-LC$^6$</td>
<td>11.6</td>
<td>P$^{11}$-LM$^2$-LC$^3$</td>
</tr>
<tr>
<td>P$^{5}$-LM$^2$-LM$^4$-D$^4$</td>
<td>1.1</td>
<td>P$^{5}$-LM$^3$-LM$^4$-D$^4$</td>
<td>P$^{5}$-LM$^2$-LM$^4$-D$^4$</td>
<td>1.1</td>
<td>P$^{5}$-LM$^3$-LM$^4$-D$^4$</td>
<td>1.1</td>
<td>P$^{5}$-LM$^3$-LM$^4$-D$^4$</td>
<td>P$^{5}$-LM$^2$-LM$^4$-D$^4$</td>
<td>P$^{5}$-LM$^2$-LM$^4$-D$^4$</td>
<td>P$^{5}$-LM$^2$-LM$^4$-D$^4$</td>
<td>13.6</td>
</tr>
<tr>
<td>P$^{3}$-LM$^3$-LM$^3$-D$^2$</td>
<td>37.0</td>
<td>P$^{3}$-LM$^3$-LM$^3$-D$^2$</td>
<td>P$^{3}$-LM$^3$-LM$^3$-D$^2$</td>
<td>37.0</td>
<td>P$^{3}$-LM$^3$-LM$^3$-D$^2$</td>
<td>37.0</td>
<td>P$^{3}$-LM$^3$-LM$^3$-D$^2$</td>
<td>P$^{3}$-LM$^3$-LM$^3$-D$^2$</td>
<td>P$^{3}$-LM$^3$-LM$^3$-D$^2$</td>
<td>P$^{3}$-LM$^3$-LM$^3$-D$^2$</td>
<td>37.0</td>
</tr>
</tbody>
</table>
Table 9. Distributions of relative cluster frequencies (%’s) per cohort for OM-based clusters. Cells are colored in blue if the standardized Pearson residuals is higher than 1.96 and in red if lower than −1.96. Clusters are characterized by their medoids. Cramer’s $V = .147$.

<table>
<thead>
<tr>
<th></th>
<th>$&lt; 30$</th>
<th>$30 - 39$</th>
<th>$40 - 49$</th>
<th>$50 - 59$</th>
<th>$\geq 60$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P - M$</td>
<td>13.4</td>
<td>12.8</td>
<td>10.7</td>
<td>5.4</td>
<td>3.8</td>
</tr>
<tr>
<td>$P$</td>
<td>35.5</td>
<td>21.1</td>
<td>15.0</td>
<td>13.8</td>
<td>18.9</td>
</tr>
<tr>
<td>Late LMC</td>
<td>17.7</td>
<td>23.0</td>
<td>18.7</td>
<td>16.0</td>
<td>13.6</td>
</tr>
<tr>
<td>$P - L - LM$</td>
<td>13.6</td>
<td>15.8</td>
<td>19.5</td>
<td>17.7</td>
<td>14.5</td>
</tr>
<tr>
<td>Early LMC</td>
<td>9.1</td>
<td>17.1</td>
<td>23.6</td>
<td>20.0</td>
<td>15.1</td>
</tr>
<tr>
<td>$P - L$</td>
<td>10.7</td>
<td>10.3</td>
<td>12.5</td>
<td>27.1</td>
<td>34.2</td>
</tr>
</tbody>
</table>
Table 10. Distributions of relative cluster frequencies (%’s) per cohort for SVR(sp, b=1)-based clusters. Cells are colored in blue if the standardized Pearson residuals is higher than 1.96 and in red if lower than −1.96. Clusters are characterized by their medoids. Cramer’s $V = .193$.

<table>
<thead>
<tr>
<th></th>
<th>&lt; 30</th>
<th>30–39</th>
<th>40–49</th>
<th>50–59</th>
<th>≥ 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P − M$</td>
<td>19.4</td>
<td>15.9</td>
<td>12.0</td>
<td>5.6</td>
<td>4.6</td>
</tr>
<tr>
<td>$P − LM$</td>
<td>11.7</td>
<td>11.4</td>
<td>10.8</td>
<td>5.1</td>
<td>2.5</td>
</tr>
<tr>
<td>$P − LM − LMC$</td>
<td>12.8</td>
<td>17.2</td>
<td>14.4</td>
<td>11.9</td>
<td>6.1</td>
</tr>
<tr>
<td>$P − LMC$</td>
<td>11.6</td>
<td>10.8</td>
<td>10.2</td>
<td>7.8</td>
<td>3.9</td>
</tr>
<tr>
<td>$P$</td>
<td>19.1</td>
<td>10.1</td>
<td>8.3</td>
<td>7.1</td>
<td>8.3</td>
</tr>
<tr>
<td>$P − LM − D$</td>
<td>2.2</td>
<td>1.5</td>
<td>2.9</td>
<td>2.0</td>
<td>1.5</td>
</tr>
<tr>
<td>$P − L − LM − D$</td>
<td>0.2</td>
<td>0.4</td>
<td>0.7</td>
<td>1.2</td>
<td>1.8</td>
</tr>
<tr>
<td>$P − L − LMC$</td>
<td>2.6</td>
<td>6.4</td>
<td>8.3</td>
<td>9.7</td>
<td>11.2</td>
</tr>
<tr>
<td>$P − L − LM − LMC$</td>
<td>4.4</td>
<td>11.4</td>
<td>13.2</td>
<td>15.2</td>
<td>17.0</td>
</tr>
<tr>
<td>$P − L − LM$</td>
<td>5.8</td>
<td>6.6</td>
<td>8.6</td>
<td>12.8</td>
<td>12.8</td>
</tr>
<tr>
<td>$P − L$</td>
<td>10.2</td>
<td>8.3</td>
<td>10.7</td>
<td>21.5</td>
<td>30.4</td>
</tr>
</tbody>
</table>
Figure 1. Structure of the matrix $M^*$, given an alphabet $\Sigma = \{a, b, c\}$ and lexicographic ordering of $\Sigma^*$. $m(a, b) = p$, $m(a, c) = q$ and $m(b, c) = r$.

$$M = \begin{pmatrix} 1 & p & q \\ p & 1 & r \\ q & r & 1 \end{pmatrix}, \quad M^* = \begin{pmatrix} M & \cdots & 0 & \cdots & 0 & \cdots \\ \vdots & M & pM & qM & \vdots \\ 0 & pM & M & rM & 0 & \cdots \\ \vdots & qM & rM & M & \vdots \\ \vdots & \vdots & M & \cdots \\ 0 & \cdots & 0 & \cdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & \vdots & q^2M \end{pmatrix}$$
Figure 2. Unit-distance plots in $(x_1, x_2)$-plane for various values of the coordinate matching-measure $m_{12}$ of the elliptical inner product $\langle \mathbf{x}|\mathbf{x} \rangle = \mathbf{x}'\mathbf{M}'\mathbf{x}$. The circle arises when $m_{12} = 0$, i.e. it represents the unit circle in “flat”, standard inner product space. As $m_{12}$ gets bigger and approaches 1, the circle is ever more elliptically deformed. For more than 2 dimensions, the unit-sphere becomes an ellipsoid (not shown).
Figure 3. Plots of discrepancy analysis’ $R^2$ (vertical axis) vs. $0 \leq m(a, b) \leq 1$ (horizontal axis) for the metrics of Table 3. The plots result from analysing the pairs of spell sequences constructed from the patterns $ca-cb$, $ac-cb$ $cac-cbc$, and $caca-cbcb$ (right panel).
Figure 4. Evolution of the $R^2$ (vertical axis) while varying time difference (horizontal axis) for different metrics (different lines), resulting from the discrepancy analysis of the time related simulation summarized in Table 4.
Figure 5. Parallel-coordinate plot of a multivariate object \((u, v, w, x) = (6, p, 1750, -4)\) and 4 multivariate objects \((u, v, w, x) = (3, b, 250, -4)\).
Figure 6. Parallel-coordinate plots of the 11 clusters found from the SVR(sp, b=1)-distances.
Figure 7. Chronograms of the six clusters found from the OM-distances
Figure 8. Parallel-coordinate plots of the six clusters found from the OM-distances.