Measuring Divergence and Convergence in Sequence Analysis: A Spell-Based Extension of Longest Common Prefixes

Yuqi Liang[1,2] and Jan H. E. Meyerhoff-Liang[2,3]

[1]Department of Sociology, University of Oxford, Oxford, United Kingdom.
[2]Institute for New Economic Thinking, University of Oxford, Oxford, United Kingdom.
[3]Blavatnik School of Government, University of Oxford, Oxford, United Kingdom.

Abstract

Measuring sequence dissimilarity is central to sequence analysis, yet empirical applications remain dominated by optimal matching (OM), a widely used but contested approach. Longest common prefix (LCP) measures offer an alternative by capturing shared beginnings and endings, making them useful for studying divergence and convergence, important concepts in social science. However, existing LCP variants are position-wise: they measure how long trajectories remain synchronized on a common clock-time axis. As a result, small shifts in transition timing can break the common prefix between trajectories that follow the same ordered spell structure, making them appear to diverge early. This study develops a spell-based perspective on sequence comparison. We introduce OMspellRS, a reference-scaled extension of spell-based OM, and propose LCPspell, a spell-based extension of LCP that measures shared ordered episodes rather than shared time positions. Using simulations and an empirical demonstration, we show that whole-trajectory measures such as the OM family capture broad trajectory profiles, whereas LCP-type measures identify where similarity breaks down or reappears. The paper provides practical guidance for choosing distance measures according to research questions and data characteristics.

Keywords: Sequence analysis, categorical time series analysis, dissimilarity measures, optimal matching, divergence, convergence

1 Introduction

Sequence analysis (SA) (Abbott et al. 2000), also referred to as categorical time series analysis (Elzinga 2010), comprises a set of methods for studying ordered sequences of discrete states

observed over time. As a method that has gained increasing popularity (Liao et al. 2022), its typical applications focus on life-course research (Mortimer et al. 2007), such as employment histories (Malo et al. 2003), family formation pathways (Barban et al. 2018), healthcare sequences (Flothow et al. 2023), and trajectories spanning multiple life domains (Pollock 2007; Aisenbrey et al. 2017). Derived from computer science and bioinformatics, SA was introduced to the social sciences by Abbott et al. (1986) to capture the holistic and processual nature of trajectories. Since its adoption, the most common analytical workflow has been to compute pairwise sequence dissimilarities, use them to cluster sequences into typical trajectory patterns, and then relate these patterns to covariates using regression models (Liao et al. 2026). This is in contrast to approaches such as event history analysis (Yamaguchi 1991), which focus on isolated transitions (e.g., death or marriage).

In empirical research, however, sequence analysis is often treated as nearly synonymous with optimal matching (OM). OM is an edit-distance approach that measures dissimilarity as the minimum cost of transforming one sequence into another through insertions, deletions, and substitutions. Since its adoption by Abbott et al. (1986), OM has become the dominant dissimilarity measure in applied work. For instance, when SA was introduced to management research by Biemann et al. (2014), it was largely framed through the use of OM, with limited discussion of alternative dissimilarity measures. Yet extensive methodological research has highlighted several limitations of OM. These include its sensitivity to the specification of substitution and indel costs (Hollister 2009; Studer et al. 2016) and the difficulty of assigning substantive meaning to edit operations in social-scientific sequence data (Wu 2000; Levine 2000; Lesnard 2010; Halpin 2014). More importantly for the present study, OM may make it difficult to disentangle timing, sequencing, and duration effects, because the same context-free operation can represent a change in spell length in one comparison and a change in sequence order in another (Hollister 2009; Halpin 2014; Studer et al. 2016).

These limitations point to a more general issue: sequence dissimilarity is not defined by a distance formula alone, but also by the representation on which that formula operates. A trajectory can be represented as a time-expanded sequence of states, where each element corresponds to a position on a shared time scale, or as a sequence of spells, where each element records a state and its duration. These representations encode different assumptions about what should count as similarity. Position-wise representations emphasize comparing sequences at each time point; spell-based representations emphasize the ordered structure of contiguous states and their durations. As a result, the same pair of trajectories may appear dissimilar under one representation but similar under another.

This distinction is especially important because empirical trajectories may differ in several ways. They may differ in timing, when the same states or transitions occur at different ages or calendar years; in sequencing, when states occur in different orders; or in duration, when spells last for different lengths of time. We refer to these as core sources of sequence variation. In addition, researchers are often interested in how differences unfold along a trajectory:

whether initially similar trajectories diverge into different pathways, or whether initially different trajectories converge toward similar outcomes. Divergence and convergence are therefore not additional sources of variation alongside timing, sequencing, and duration. Rather, they describe where similarity breaks down or reappears along a trajectory.

This framing suggests that prefix- and suffix-based measures deserve more attention. The longest common prefix (LCP) (Elzinga 2007) measures the length of the shared initial segment between two sequences and therefore provides a direct operationalization of divergence: it captures how long initially similar trajectories remain aligned before they separate. Its reverse variant, RLCP, applies the same logic from the end of the sequence and captures convergence as shared terminal structure. Although such ideas are closely related to substantive discussions of divergence and convergence in life-course research (Rehnberg et al. 2019; Liao et al. 2021), social stratification (Grusky et al. 1984), healthcare (Hembram et al. 2020; Stevens 2010; Graham et al. 2016), and the literature on institutional change (Meyer et al. 1975; Kiess et al. 2017; Rubin 2023), prefix-based measures remain comparatively underused in applied sequence analysis.

This focus on where trajectories separate, or come together, is also related to existing methodological work on the temporal directionality of sequence similarity. One relevant approach is the Euclidean metric for longitudinal timelines proposed by Rousset et al. (2012), discussed in Studer et al. (2016) as `CHI2fut`.[1] This measure compares time-stamped states through their transition profiles toward future situations, so that situations are treated as close when they lead to similar futures. By reversing sequences, the same logic can also emphasize trajectories that unfold from a similar beginning. A second related approach is the sliding-window discrepancy analysis of Studer et al. (2011), which studies how group-level differences evolve across successive time windows. These approaches are important complements to the present study: `CHI2fut` defines similarity through future-oriented transition profiles, whereas sliding-window discrepancy analysis localizes group differences along calendar time.

LCP-type measures address a different but related question. Rather than defining similarity through future profiles or estimating group-level discrepancy within moving windows, they define pairwise dissimilarity through shared beginnings or endings. They therefore provide a direct pairwise operationalization of divergence and convergence. However, the basic LCP is position-wise. It compares sequences by matching the first time point with the first time point, the second with the second, and so on. This can lead to misleading interpretations when trajectories share the same ordered pathway but differ in transition timing. For example, consider employment trajectories in which E denotes employment and U denotes unemployment. Two trajectories may both follow the pathway $E \rightarrow U \rightarrow E$, that is, the same sequence of distinct successive states (DSS), but one may enter unemployment one period earlier than the other. Under position-wise LCP, this timing shift can immediately break the common prefix, making

[1] For further details, please see the function `seqemlt` in the `TraMineRextras` package (Ritschard et al. 2021).

the trajectories appear to diverge even though they follow the same underlying pathway. The problem is not only that LCP is sensitive to timing; it is that the position-wise representation prevents the measure from distinguishing a timing shift from a difference in spell order.

This limitation suggests moving prefix comparison from time positions to spells. This idea is related to Elzinga (2007), who already considered spell-oriented LCP variants based on the longest common prefix of DSS, with duration adjustments based on minimal shared time (LCPmst) and product duration weighting (LCPprod). Our proposal builds on this insight but handles duration differently. Instead of folding duration directly into the prefix score through fixed minimal-time or product-based accumulation rules, LCPspell separates spell-order matching from duration weighting. It first asks how many consecutive spells two trajectories share from the beginning or the end, and then allows duration differences within the matched prefix or suffix to enter through an optional, reference-scaled duration component. In substantive terms, state-sequence LCP measures synchronicity in clock time, whereas LCPspell measures structural similarity in the life course. This makes it possible to distinguish trajectories that differ only in transition timing from trajectories that differ in spell order, while allowing researchers to decide explicitly how much spell-length differences should matter.

The duration component of LCPspell builds on spell-based optimal matching (OMspell) (Studer et al. 2016). In this paper, we follow the OMspell cost convention, in which expansion costs apply only to duration beyond the one-unit spell base, and introduce OMspellRS as a reference-scaled extension of this logic. The motivation is simple: in OMspell, duration costs are expressed in raw time units, so the same parameter value has different meanings when time is measured in months rather than years. OMspellRS rescales duration terms by a fixed reference duration scale, such as the total observation window or the maximum possible spell duration implied by the study design, making duration effects easier to compare across time scales.

The paper makes four contributions. First, it clarifies the link between sequence representation and dissimilarity measurement by distinguishing position-wise and spell-based comparison, and by relating these representations to timing, sequencing, duration, divergence, and convergence. Second, it introduces LCPspell and RLCPspell as spell-based prefix and suffix measures for studying divergence and convergence on the spell level. Third, it proposes OMspellRS as a reference-scaled extension of the OMspell cost convention and uses this reference-scaled logic as the basis for the duration component in LCPspell. Fourth, we introduce a chance-corrected pseudo-R^2 criterion for evaluating sensitivity in simulation experiments, which removes the non-zero baseline of the raw between-group discrepancy share and makes zero interpretable as no separation beyond chance. The proposed methods are implemented in the `Sequenzo` package (Liang et al. 2025).

The remainder of the paper is organized as follows. We first provide a conceptual review of data representations, the three core sources of sequence variation, and divergence/convergence as foundations for understanding sequence comparisons. We then show how standard position-

wise OM can be reformulated at the spell level, introducing OMspell and OMspellRS, before presenting LCPspell. Section 3 reports a simulation study that examines how the proposed and related measures respond to different sources of sequence variation. Section 5 applies the methods to an empirical example. Section 6 concludes.

2 Spell-Based Sequence Comparison: Data Representations, Sources of Sequence Variations, and Dissimilarity Measure Refinements

2.1 Sequence data representations

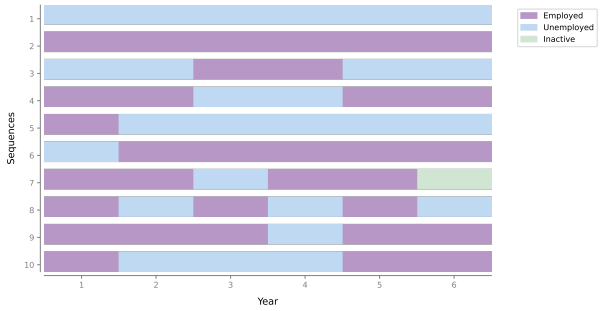


Figure 1: Ten sequences for illustration

Life-course sequence data are commonly stored as ordered series of categorical states observed at regular time intervals, such as yearly observations from age 15 to 39. A state denotes the status occupied at a specific time point, such as being employed (E), unemployed (U), or inactive (I) in a given year. To illustrate, Figure 1 displays ten sequences observed over six years. The first sequence can be written as `U, U, U, U, U, U`, whereas the tenth sequence is `E, U, U, U, E, E`. In this expanded state-sequence representation, sequences are most naturally compared position by position: the first state of one sequence is compared with the first state of another, the second with the second, and so forth. Such position-wise comparisons implicitly prioritize timing because similarity depends on whether individuals occupy the same state at the same observed time point. In the figure, Sequences 1 and 10 match only at the second through fourth positions, and are therefore measured as relatively dissimilar under a purely timing-sensitive comparison.

Position-wise comparison, however, is not the only way to compare trajectories. In many life-course applications, researchers are also interested in how states unfold as spells, also referred to as episodes: when states occur (timing), the order in which states unfold (order/sequencing) and how long states persist (duration). This shifts attention from individual time positions to spells. A spell is a maximal uninterrupted stretch of the same state, and its duration is the number of consecutive time units spent in that state. For instance, the sequence `E`

E E U U E E contains three spells: an employment spell of length three, an unemployment spell of length two, and another employment spell of length two.

This shift from positions to spells is conceptually simple, but it requires some terminological care. The sequence-analysis literature uses several closely related formats to represent the same underlying trajectory. The expanded state-sequence representation records the state observed at each time position. In this paper, we refer to this as the state-sequence representation. By contrast, spell-based representations collapse consecutive repetitions of the same state and describe trajectories as ordered episodes. Some spell-based formats retain only the order of distinct successive states, while others also retain spell durations or explicit start and end positions. Following the terminology used in Gabadinho et al. (2011), Table 1 summarizes these distinctions.

The paper mainly uses three representations for clarity: state-sequence, DSS, and SPS. The state-sequence representation is needed as the baseline for standard position-wise sequence comparison. DSS is useful for isolating spell order after durations have been removed. SPS is the central representation for the spell-based measures developed below, because it retains both the order of spells and their durations without requiring comparisons to be anchored at identical calendar positions. The SPELL format is included in Table 1 for terminological completeness.

Table 1: Four representations of the same state sequence

Representation	**Definition**	**Example**	**Main information retained**
Position-wise representation			
State sequence (STS)	An expanded sequence in which each element records the state observed at a given time position.	E E E U U E E	Calendar-time alignment and position-wise state information.
Three forms of spell-based representation			
Distinct successive states (DSS)	The ordered list of state labels obtained after consecutive repetitions of the same state have been collapsed.	$\mathtt{E} \rightarrow \mathtt{U} \rightarrow \mathtt{E}$	Spell order only; durations and exact timing are removed.
State-Permanence Sequence (SPS)	A sequence of distinct successive states together with their associated durations.	$(\mathtt{E}, 3) \rightarrow (\mathtt{U}, 2) \rightarrow (\mathtt{E}, 2)$	Spell order and spell duration; exact start and end positions are implicit if the sequence origin is known.
SPELL format	A long-format representation in which each row corresponds to one spell and records its state, start position, and end position.	Spell 1: E, positions 1–3 Spell 2: U, positions 4–5 Spell 3: E, positions 6–7	State, duration, and explicit start and end positions for each spell.

2.2 Three core sources of sequence variation

The distinctions between representations indicate that meaningful sequence comparison requires separating the main sources of sequence variation that are often conflated in practice: timing, sequencing, and duration. Although empirically intertwined, these sources address different substantive questions and require different approaches in sequence comparison. Existing research has introduced these three sources in various ways (Studer et al. 2016; Liao 2021; Liao et al. 2021; Liao et al. 2022); to facilitate further understanding, we link each source to data representations, and provide example research questions and data patterns that capture each source.

Timing. Timing concerns when states or transitions occur on a shared external clock, such as age or calendar year (Liao et al. 2022). For instance, in Figure 1, Sequences 1 and 5 share very similar sequencing because only the state at the first position does not match (`U` vs. `E`). Two trajectories may share the same sequencing but differ in timing if the same transitions occur earlier or later in life, such as entering marriage at age 20 rather than age 30.

Timing becomes particularly important when the research question focuses on whether transitions occur earlier, later, or simultaneously on a common time scale. For example, one may ask whether a policy change following German reunification in 1990 led to simultaneous shifts in employment trajectories in East and West Germany, such as movements from stable employment into unemployment, inactivity, or retraining, or whether one region experienced these changes earlier than the other. In such cases, differences in the timing of transitions, rather than their order, are of primary substantive interest.

Timing is also especially relevant when sequences contain similar states or transitions but differ in the time points at which these states or transitions occur. In extreme cases, as illustrated by Sequences 1 and 2 in Figure 1, two sequences may occupy the same states for comparable durations but never match at the same positions, making them entirely dissimilar under strict position-wise comparison. In empirical data, the importance of timing often depends on how rapidly states change over time (Figure A.1 in Appendix A). For instance, in the carbon emissions per capita trajectories, many countries remain in the same state for extended periods, leading to relatively synchronized timing patterns. In contrast, in the development aid trajectories, countries frequently transition between states over short intervals, indicating substantial timing variation. In such contexts, dissimilarity measures that are sensitive to timing differences are essential for capturing meaningful distinctions between trajectories.

Sequencing/order. Sequencing refers to the order of distinct successive states, or equivalently the ordered state labels of spells. Two trajectories share the same sequencing if they consist of the same ordered list of spells, regardless of how long each spell lasts or when transitions occur. For example, in Figure 1, the sequencing of Sequences 4 and 10 is identical because they both have the same spell structure (`E`→`U`→`E`).

Sequencing is particularly important when the research question concerns the order in

which states or transitions occur. Put simply, sequencing asks whether the same states appear in the same order, rather than whether they occur at the same time or last for the same duration. To give an example from economic research on structural change in cities, one may ask whether cities typically develop capabilities in computing before moving into artificial intelligence, or whether the reverse order is more common. Similar logic applies to life-course trajectories, where the ordering of events, such as employment and marriage, may differ even when they occur at similar ages.

Sequencing also becomes salient when the data exhibit substantial variation in the order of states while other sources of variation, such as timing and duration, remain relatively stable. In such contexts, differences between trajectories are primarily driven by variation in order rather than by when states occur or how long they last. For example, Liao et al. (2020) show that migrant domestic workers in Hong Kong share similar starting and ending states but differ in their intermediate trajectories.

Duration. Duration refers to how long spells last, that is, the length of time spent consecutively in a given state. It is most meaningfully compared when the corresponding spells involve the same state. For example, Sequences 9 and 10 have the same spell order (E→U→E) but differ in duration: Sequence 9 remains employed for three time units before a short unemployment spell, whereas Sequence 10 enters unemployment earlier and remains there for three time units. Thus, the two sequences are similar in sequencing but different in duration. In social science research, duration is particularly important when sustained exposure has different implications than intermittent exposure. For example, short-term unemployment may be more transient and less consequential than long-term unemployment.

Duration is essential when the research question concerns how long individuals remain in a given state. In practice, this duration is typically represented as spells, that is, contiguous segments of identical states. An example research question is whether long-term unemployment increases the likelihood of unstable health trajectories, distinguishing sustained disadvantage from short-term, transient experiences.

In empirical data that exhibit substantial variation in spell lengths, trajectories may follow similar pathways and occur at similar points in time, but differ in how long individuals remain in each state. Empirical patterns often reflect this form of variation. As shown in Figure A.2 in Appendix A, many regions follow similar sequences but differ in how long they remain in the "no patents" state.

It is important to note that real-world sequence data rarely vary along a single source of variation. Timing, sequencing, and duration are intertwined, and the analytical challenge lies in identifying which source is most relevant for a given research question and dataset. These sources are not explicitly parameterized in most dissimilarity measures; rather, they provide a conceptual framework for understanding how different measures implicitly weight and capture distinct aspects of sequence differences. Distinguishing them is therefore essential for selecting an appropriate sequence comparison method, as different measures systematically emphasize

different sources.

No distance measure is neutral. Each embodies specific assumptions about whether similarity should be driven primarily by temporal alignment of transitions (timing), ordered spell structure (sequencing), or spell lengths (duration). These assumptions have substantive consequences: different measures can lead to different interpretations of the same data, depending on which source of variation they highlight. This is especially consequential in spell-based comparisons, where duration enters the representation directly and can dominate the resulting dissimilarities if not carefully controlled. Accordingly, the choice and parameterization of a distance measure should be guided by which source is substantively central, as well as by how strongly the method weights timing and duration relative to sequencing.

Divergence and convergence. Timing, sequencing, and duration describe what differs between trajectories: whether states occur at different points in time, follow different orders, or last for different lengths. Divergence and convergence refer to a different aspect of sequence comparison. They describe where along a trajectory similarity breaks down or reappears. Two trajectories diverge when they initially follow the same pathway but later separate; they converge when they follow different earlier pathways but later become similar. These concepts are therefore relational and position-dependent: they concern not only whether two trajectories differ, but also where differences emerge or disappear.

The distinction can be seen in Figure 1. Sequences 9 and 10 both follow the same spell order, E→U→E, but differ in timing and duration. In calendar time, they diverge immediately after the first position: Sequence 9 remains in E, whereas Sequence 10 enters U. In spell order, however, they do not diverge in terms of the ordered states, because both sequences pass through the same ordered succession of spells. What differs is the duration of the first two spells. This example shows that divergence is feature-dependent: the same pair of trajectories may diverge early with respect to calendar-time alignment, remain identical with respect to spell order, and differ only in spell duration. By contrast, Sequences 6 and 7 differ from the first time point and also begin with different spells, and therefore show immediate divergence under both position-wise and spell-based prefix comparison.

This distinction is important because divergence and convergence can involve differences in timing, sequencing, or duration. Two trajectories may differ because the same transition occurs earlier in one sequence than in the other, because the next spell involves a different state, or because one trajectory remains longer in a state before moving on. These sources of difference are not equivalent. Timing places divergence or convergence on the calendar-time axis, while sequencing places it in the ordered succession of spells. Duration affects the length of the relevant spells: it can make a difference persist for more time units, or make a similarity last longer before the trajectories separate. In this sense, duration shapes the persistence and magnitude of divergence or convergence, whereas timing and sequencing indicate where it occurs.

Empirical illustrations of such patterns are provided in Figure A.3 in Appendix A. In family

formation trajectories from the German Family Panel (Brüderl et al. 2020; Raab et al. 2022), which tracks 1,027 individuals from ages 18 to 40, individuals often start from a similar state, such as being single without children, but subsequently diverge into different family-formation patterns involving cohabitation, marriage, or parenthood. Standard dissimilarity measures may still separate such trajectories into different clusters, but they usually do so by summarizing overall dissimilarity across the full sequence. They do not, by themselves, directly identify how long trajectories remain similar before separating, or whether the separation occurs early or late. In this context, research questions naturally focus not only on which later pathways individuals enter, but also on when and how initially similar trajectories begin to diverge. Conversely, convergence patterns arise when initially different trajectories become similar later. For example, youth transition trajectories in Northern Ireland (McVicar et al. 2002; Gabadinho et al. 2011) illustrate how individuals with heterogeneous early pathways may converge toward similar later-life states, such as employment or joblessness.

From a methodological perspective, this distinction matters because different dissimilarity measures encode the location and persistence of differences in different ways. Position-wise measures locate divergence and convergence on a shared time axis. Spell-wise measures locate them in the ordered sequence of spells. Prefix- and suffix-based measures are especially relevant for operationalizing divergence and convergence because they directly measure shared beginnings and endings: a long common prefix indicates that two trajectories remain similar for an initial segment before diverging, whereas a long common suffix indicates that initially different trajectories converge toward a shared terminal segment. A spell-based prefix measure can therefore ask whether two trajectories share the same initial spell structure even if their transitions occur at different times, while a spell-based suffix measure can ask whether trajectories with different early histories converge toward the same terminal spell structure.

This framing also clarifies the role of duration. Duration can strengthen or weaken the substantive importance of divergence and convergence by determining how long separation or alignment persists, but it should not be treated as an independent basis for locating divergence or convergence. The next step is therefore to examine how existing dissimilarity measures encode these distinctions, and why a spell-based perspective is useful for clarifying how timing, order, duration, divergence, and convergence are represented in sequence comparison.

2.3 From position-wise OM to spell-based OM

To make this link between representation and measurement concrete, we begin with the most widely used class of sequence dissimilarity measures: edit-distance methods. The most prominent representative is Optimal Matching (OM), which aligns sequences using three types of edit operations: substitutions (changing one state to another), insertions (adding a state), and

deletions (removing a state)[2].

Understanding how OM operates is essential because our proposed LCPspell method builds on a modified version of spell-based OM. We begin by examining how standard OM works and why it can become dominated by duration-related differences. Subsequently, we will introduce the spell-based OM (OMspell) and its modified version (OMspellRS). We use the term position-wise OM to refer to standard OM applied to time-expanded state sequences, where each element corresponds to the state observed at a particular time position. This contrasts with spell-based OM, which operates on spell sequences whose elements combine a state and its duration.

2.3.1 Why position-wise OM can become duration-dominated

Standard OM compares sequences in a time-expanded representation (e.g., `U, U, U, U, U, U` instead of $(\mathtt{U}, 6)$). A key feature of OM methods is that, by permitting indels, they allow a time warp: the temporal axis of one sequence can be stretched or compressed relative to the other, so that corresponding states need not occur at the same calendar positions. For instance, consider Sequences 9 and 10 in Figure 1. Both trajectories start in `E`, pass through `U`, and end in `E`, but the `U` spell starts earlier and lasts longer in Sequence 10. By introducing indels, OM can shift one sequence relative to the other, effectively stretching or compressing the time axis to improve alignment. This flexibility distinguishes edit-distance methods from strict position-wise comparisons (such as Hamming distance), which require matching at identical time indices.

Formally, let two state sequences be denoted by

$$\mathbf{x} = (x_1, \ldots, x_T), \qquad \mathbf{y} = (y_1, \ldots, y_{T'}),$$

where each element represents the observed state at a given time point. The OM distance on time-expanded sequences between $\mathbf{x}$ and $\mathbf{y}$ is defined as the minimum total cost required to transform $\mathbf{x}$ into $\mathbf{y}$ using a sequence of elementary edit operations:

$$d_{\mathrm{OM}}(\mathbf{x}, \mathbf{y}) = \min_{\pi \in \Pi} \sum_{o \in \pi} c(o), \tag{1}$$

where Π denotes the set of all admissible edit paths. The cost $c(o)$ of each operation o depends on its type and is defined at the level of individual time positions. Let c_{indel} denote the constant

[2]In practice, insertions and deletions are often grouped together and referred to as *indels* (a portmanteau of insertion and deletion). This grouping reflects the fact that, from the perspective of sequence alignment, an insertion in one sequence is equivalent to a deletion in the other sequence being compared. For example, transforming sequence A into sequence B by inserting a state at position t produces the same alignment as transforming sequence B into sequence A by deleting that state at position t. Because the cost structure and algorithmic treatment of insertions and deletions are typically symmetric, they are commonly combined into a single indel operation with a unified cost parameter, simplifying both notation and implementation.

indel cost and $\sigma(a, b)$ the substitution cost between states a and b:

$$\text{Insertion/Deletion of position } x_t: \quad c(o) = c_{\text{indel}}, \tag{2}$$

$$\text{Substitution } x_t \leftrightarrow y_{t'},\ x_t = y_{t'}: \quad c(o) = 0, \tag{3}$$

$$\text{Substitution } x_t \leftrightarrow y_{t'},\ x_t \neq y_{t'}: \quad c(o) = \sigma(x_t, y_{t'}). \tag{4}$$

Crucially, all edit operations are defined at the level of individual time positions. Costs depend only on state comparisons and do not include duration as a separate variable. Standard OM therefore does not compare spell lengths directly. Instead, duration enters the distance through repetition: a long spell appears as many repeated state observations in the time-expanded sequence. Because each mismatch at a time position contributes a separate cost, differences that persist over many consecutive time points accumulate these costs over time. As a result, a single long spell can contribute more to the overall distance than multiple short mismatches of the same type. Duration is thus not explicitly modeled but arises mechanically from the accumulation of position-level costs, and its influence cannot be independently scaled or controlled. Consequently, prolonged differences may dominate the distance even when the substantive focus lies elsewhere, such as in the ordering of states.

2.3.2 OMspell: Optimal matching on sequences of spells

A more direct way to avoid the mechanical accumulation of position-level costs is to compare sequences at the level of spells. This is the logic of spell-based optimal matching (OMspell), introduced by Studer et al. (2016). Instead of aligning repeated observations one by one, OMspell aligns spell-level units, where each spell combines a state and a duration. This changes how duration enters distance comparison. In standard OM, duration affects the distance indirectly because long spells contain many repeated positions. In OMspell, duration enters explicitly through the costs attached to spell-level insertions, deletions, and substitutions.

Formally, let two spell sequences be denoted by

$$\mathbf{s} = \big((a_1, d_1), \ldots, (a_{n_s}, d_{n_s})\big), \qquad \mathbf{r} = \big((b_1, e_1), \ldots, (b_{m_s}, e_{m_s})\big),$$

where a_k and b_ℓ denote spell states, d_k and e_ℓ their corresponding durations, and n_s and m_s denote the numbers of spells in $\mathbf{s}$ and $\mathbf{r}$, respectively. The OMspell distance is defined as the minimum total cost required to transform $\mathbf{s}$ into $\mathbf{r}$ through spell-level edit operations:

$$d_{\text{OMspell}}(\mathbf{s}, \mathbf{r}) = \min_{\pi \in \Pi} \sum_{o \in \pi} c(o), \tag{5}$$

where Π denotes the set of all possible spell-level edit paths transforming $\mathbf{s}$ into $\mathbf{r}$, and $c(o)$ denotes the cost of operation o.

The cost of each operation depends on whether the operation is an insertion/deletion, a

same-state substitution, or a different-state substitution. Let $c_{\text{indel}}(a)$ denote the cost of inserting or deleting a spell with state a, and let $\sigma(a, b)$ denote the substitution cost between states a and b. The parameter λ controls the contribution of duration differences. It corresponds to `expcost` in the implementation.[3] The OMspell cost convention is:

$$\text{Insertion/Deletion of } (a_k, d_k): \quad c(o) = c_{\text{indel}}(a_k) + \lambda\,(d_k - 1), \tag{6}$$

$$\text{Substitution } (a_k, d_k) \leftrightarrow (b_\ell, e_\ell),\ a_k = b_\ell: \quad c(o) = \lambda\,|d_k - e_\ell|, \tag{7}$$

$$\text{Substitution } (a_k, d_k) \leftrightarrow (b_\ell, e_\ell),\ a_k \neq b_\ell: \quad c(o) = \sigma(a_k, b_\ell) + \lambda\,(d_k + e_\ell - 2). \tag{8}$$

The terms $d_k - 1$ and $d_k + e_\ell - 2$ follow from the spell-level nature of the operation. Inserting or deleting a spell already accounts for the existence of one spell unit, so the duration-expansion part is charged only for the remaining duration beyond that unit. For example, deleting a spell $(a, 1)$ costs only $c_{\text{indel}}(a)$, whereas deleting a spell $(a, 4)$ costs $c_{\text{indel}}(a) + \lambda(4 - 1)$. The first unit is covered by the spell deletion itself, and only the remaining three units are treated as duration expansion. Similarly, when two spells have different states, one unit from each spell is covered by the state substitution, and only the remaining durations are added to the expansion component. This is why different-state substitution uses $d_k + e_\ell - 2$ rather than $d_k + e_\ell$.

Same-state substitution follows a different logic. If two spells have the same state, OMspell does not charge a state substitution cost. It only penalizes the difference in their durations through $\lambda|d_k - e_\ell|$. Thus, two spells $(a, 4)$ and $(a, 2)$ differ only by two duration units, whereas two spells $(a, 4)$ and $(b, 2)$ differ both in state and in the duration expansion needed around the spell-level substitution.

The expansion-cost parameter λ determines how strongly duration contributes to the total distance. When $\lambda = 0$, all duration-related terms vanish, and OMspell compares only the ordered sequence of spell states. As λ increases, spell-duration differences contribute more strongly to the distance. The practical influence of a given λ also depends on the substitution and indel costs and on the empirical distribution of spell durations. Thus, OMspell makes the role of duration explicit, but duration remains expressed in raw time units.

2.4 Reference-Scaled OMspell (implemented as OMspellRS)

Making duration explicit raises an additional issue. In the OMspell formulation, duration terms are measured in the raw time unit of the data. A duration difference of five years is therefore treated as 5 units if the data are coded yearly, but as 60 units if the same trajectories are coded monthly. If the same value of the expansion-cost parameter λ is applied directly to these raw durations, the monthly version assigns a much larger duration cost to the same substantive difference. In OMspell, λ therefore has the interpretation of a cost per observed time unit, and

[3] The abbreviation "expcost" stands for expansion cost, not exponential. It refers to the cost of expanding or compressing spell durations when aligning spells (Studer et al. 2016).

its practical meaning depends on the temporal unit used to code the data.

OMspellRS addresses this issue by expressing duration differences relative to a fixed reference duration. The idea is simple: instead of asking how many raw time units two spells differ by, we ask how large this difference is relative to the time scale of the study. For example, in a 20-year observation window, a five-year duration difference corresponds to $5/20 = 0.25$ of the full observation period. If the same trajectories are coded monthly, the same substantive difference is 60 months in a 240-month observation window, which is again $60/240 = 0.25$. Thus, duration enters the distance as a relative quantity rather than as a raw count of time units.

We denote this fixed reference duration by $\tau > 0$. In most applications, τ can be set to the total observation window, or equivalently to the maximum possible spell duration implied by the study design. If sequences are observed over 20 yearly positions, one may set $\tau = 20$; if the same trajectories are observed over 240 monthly positions, one may set $\tau = 240$. The reference duration is fixed before distance computation. It is not estimated from the maximum observed spell duration in the sample, and it is not defined separately for each pair of sequences.

With this notation, OMspellRS preserves the same spell-level alignment logic as OMspell. It uses the same edit operations, base insertion/deletion costs, substitution costs, and duration-weight parameter λ. The only modification is that each duration-expansion term is divided by the fixed reference duration τ before being multiplied by λ. This is what we mean by reference-scaled: duration costs are scaled by a fixed reference duration from the study design rather than being measured directly in raw time units.

The reference-scaled spell-level costs are:

$$\text{Insertion/Deletion of } (a_k, d_k): \quad c(o) = c_{\text{indel}}(a_k) + \lambda \frac{d_k - 1}{\tau}, \tag{9}$$

$$\text{Substitution } (a_k, d_k) \leftrightarrow (b_\ell, e_\ell),\ a_k = b_\ell: \quad c(o) = \lambda \frac{|d_k - e_\ell|}{\tau}, \tag{10}$$

$$\text{Substitution } (a_k, d_k) \leftrightarrow (b_\ell, e_\ell),\ a_k \neq b_\ell: \quad c(o) = \sigma(a_k, b_\ell) + \lambda \frac{d_k + e_\ell - 2}{\tau}. \tag{11}$$

The interpretation of λ is therefore similar in spirit to OMspell, but the unit of interpretation changes. In both formulations, $\lambda = 0$ removes the duration component, so that spell duration differences do not contribute to the distance. Larger values of λ give duration differences more weight relative to state substitutions and insertions/deletions. The difference is that, in OMspell, λ is a cost per raw time unit. In OMspellRS, λ is a cost attached to a duration difference measured relative to the reference duration τ. For example, if τ is the full observation window and two matched spells differ by one quarter of that window, the duration component is 0.25λ. Thus, λ remains the duration-weight parameter, but its effect is interpreted on a relative scale rather than in raw years, months, or other observed time units.

In `Sequenzo`, the reference duration τ is implemented as `duration_ref`. This parameter should be distinguished from `expcost`. `duration_ref` determines the duration scale used in the denominator, whereas `expcost` (λ) determines how much weight is attached to

the scaled duration difference. Unless there is a substantive reason to use a different value, `duration_ref` should be set to the total observation window T, which is also its default value in OMspellRS. Under this choice, a duration difference is interpreted as a proportion of the full period covered by the sequence. For example, a five-year difference in a 20-year sequence contributes $5/20 = 0.25$ before being multiplied by `expcost`. If the same sequence is coded monthly, the corresponding values are 60 months and 240 months, giving the same ratio $60/240 = 0.25$. The spell durations and `duration_ref` must therefore be measured in the same time unit, for example both in years or both in months.

This fixed choice of `duration_ref` avoids two problems. First, if τ were set to the largest spell duration observed in the dataset, the distance between the same two sequences could change simply because other sequences were added to or removed from the sample. Second, if τ were chosen separately for each pair of sequences, the same spell comparison could receive different costs depending on which pair is being compared. OMspellRS avoids these problems by fixing τ before distance computation.

Table 2 summarizes the difference between OM, OMspell, and reference-scaled OMspell. OM operates on time-expanded positions and has no explicit duration term. OMspell compares spells directly and incorporates duration in raw time units. Reference-scaled OMspell retains the spell-level comparison but divides all duration-expansion terms by τ, so that λ becomes interpretable relative to the reference duration.

Table 2: Cost formula comparison: OM, OMspell, and reference-scaled OMspell

Operation	**Cost type**	**OM**	**OMspell**	**Reference-scaled OMspell**
Insertion / Deletion	c_{indel}	c_{indel}	$c_{\text{indel}}(a) + \lambda(d_i - 1)$	$c_{\text{indel}}(a) + \lambda\frac{d_i-1}{\tau}$
Substitution (same state)	same state	0	$\lambda\lvert d_i - d_j\rvert$	$\lambda\frac{\lvert d_i-d_j\rvert}{\tau}$
Substitution (different states)	different states	$\sigma(s_i, s_j)$	$\sigma(s_i, s_j) + \lambda(d_i + d_j - 2)$	$\sigma(s_i, s_j) + \lambda\frac{d_i+d_j-2}{\tau}$

These cost formulas define a raw spell-level distance. Whether this raw distance should subsequently be normalized is a separate question. The reference-scaled modification concerns the operation costs themselves: it changes the scale on which duration enters the edit operations.

Table 3 summarizes the differences between OMspell and OMspellRS. Appendix B provides an illustration comparing standard OM, OMspell, and OMspellRS using two simple pairs of sequences. By varying the value of λ, examples show how the same pair of life-course trajectories can be interpreted as differing primarily in timing or primarily in duration, depending on the level at which optimal matching is applied.

Normalization. The reference-scaled formulation above changes how duration enters the operation costs of OMspell. It defines a raw spell-level distance: the dynamic programming algorithm first computes the minimum cost of transforming one spell sequence into another. A separate question is whether this raw distance should then be rescaled for reporting or comparison. In this paper, we use the term normalization for this second step. Thus, OMspellRS is

Table 3: OMspell and reference-scaled OMspell

Dimension	**OMspell**	**Reference-scaled OMspell**
Duration scale	Raw units; λ is tied to the time unit	Duration divided by fixed τ; λ interpretable relative to τ
Scaling factor	Not applicable	Fixed before distance computation; not dataset- or pair-dependent
Raw distance	Spell-level edit distance with raw duration costs	Same spell-level edit logic with relative duration costs
Normalization	Optional post-processing step	Optional post-processing step
Main change from standard OM	Duration enters as an absolute time quantity	Duration enters as a proportion of a reference duration

not itself a normalization of the final distance matrix. It is a modification of the cost formula before the distance is computed.

This distinction matters because two different scales are involved. `duration_ref` rescales duration terms inside the edit costs. By contrast, distance normalization rescales the final pairwise distance after all edit costs of sequences have been minimized. Researchers may want to normalize final distances when comparing several dissimilarity measures whose raw values are on different scales, or when a downstream procedure is easier to interpret with bounded distances (e.g., between 0 and 1). Such normalization is therefore a reporting and implementation choice, not part of the definition of OMspellRS.

For OMspell-type distances, normalization requires a denominator that represents a reasonable upper value for the raw distance. A simple choice is the cost of transforming one spell sequence into the other by pairing as many spells as possible, substituting paired spells when needed, and inserting or deleting the remaining spells. For two spell sequences with n_s and m_s spells, this upper value can be written as

$$d_{\text{upper}} = |n_s - m_s| \ \max(c_{\text{indel}}) + \max(n_s, m_s) \ \max_{a \neq b} \sigma(a, b). \tag{12}$$

Here, n_s and m_s denote the numbers of spells in the two sequences. The first term accounts for the extra spells in the longer sequence, which can be inserted or deleted. The second term accounts for the spells that can be paired and substituted. In other words, the formula gives the cost of a simple spell-level transformation: align the two spell sequences as far as possible, substitute paired spells at the largest possible substitution cost, and insert or delete the remaining unmatched spells at the largest possible indel cost.

Because OMspell is defined as a minimum-cost alignment, its computed distance cannot be larger than the cost of any available transformation strategy. If a simple strategy based on substitutions and insertions/deletions has cost d_{struct}, then the optimal OMspell distance should not exceed this value. The algorithm may choose this strategy or a cheaper one, but it should not choose a more expensive one.

This also explains why the duration-expansion component is not added as a separate term in this structural reference value. Expansion costs are part of particular spell-level operations, but the optimal alignment is free to choose the cheapest combination of operations. If using duration expansion would be more expensive than substituting, inserting, or deleting spells under the base costs, the optimal alignment should choose the cheaper alternative. The structural reference value therefore reflects an upper reference cost under the spell-level edit structure, rather than the cost of the most expensive conceivable sequence of operations.

The key point is that OMspellRS and normalization answer different questions. OMspellRS asks how duration should enter the edit costs before the distance is computed. Normalization asks how the final raw distance should be rescaled after it has been computed. The main contribution of OMspellRS is the first point: it expresses duration terms relative to a fixed reference duration τ, so that duration differences are not tied directly to raw years, months, or other observed time units. It does not introduce a new general normalization formula for the final distance matrix.

2.5 Beyond Edit Distances: The Four Variants of Longest Common Prefixes for Divergence and Convergence

The preceding subsections clarified how edit-distance methods compare sequences by minimizing transformation costs. We now turn to a different family of measures: longest common prefix (LCP) measures. Unlike edit-distance methods, which summarize dissimilarity over the whole alignment, LCP-based measures focus on where similarity breaks down. This makes them especially relevant for operationalizing divergence and convergence, as defined above. The basic LCP compares two sequences from the beginning and measures the length of their shared prefix. It therefore captures how long two trajectories remain identical before they diverge. Its reverse variant, RLCP, applies the same logic from the end of the sequence and captures how much terminal structure two trajectories share after potentially different earlier histories. In this sense, LCP and RLCP translate the concepts of divergence and convergence into prefix- and suffix-based distance measures.

Existing LCP variants build on this prefix/suffix logic but differ in how they treat duration and the underlying representation. Basic LCP compares time-indexed state sequences and ignores duration as a separate component. Elzinga (2007) also considered spell-oriented variants by applying LCP logic to sequences of distinct successive states (DSS) and adjusting the resulting prefix similarity for duration. In this paper, we refer to these duration-adjusted variants as LCPmst and LCPprod. LCPmst incorporates minimal shared time, whereas LCPprod weights matched elements by the product of their durations. The basic LCP variant is currently implemented in `TraMineR` (Gabadinho et al. 2011); for completeness, all three variants are implemented in `Sequenzo` (Liang et al. 2025).

Our proposed LCPspell builds on this DSS-based insight but handles duration differently.

Rather than folding duration into the prefix score through minimal-time or product-based accumulation, LCPspell first identifies the shared prefix or suffix of spell states and then adds duration differences as an optional penalty. This separates two questions that are analytically distinct: whether two trajectories share the same ordered spell structure, and how much the durations of the matched spells differ. The duration penalty follows the reference-scaled logic introduced above for OMspellRS, so that the duration component is expressed relative to a fixed reference duration rather than in raw time units.

This choice should be distinguished from other spell-focused dissimilarity measures, most notably SVRspell (Elzinga et al. 2015) and OMslen (Halpin 2010). SVRspell represents sequences in a high-dimensional feature space of matching spell subsequences and is designed to capture global similarities in subsequence structure. It is therefore well suited to studying overall ordering patterns, but it does not preserve the prefix/suffix interpretation that motivates LCP-type measures. Since LCPspell aims to operationalize divergence and convergence by identifying where shared spell structure breaks down or reappears, it requires a local spell-to-spell comparison rather than a global subsequence-feature representation. OMslen is also less suitable as a direct basis for the present extension, because it modifies edit costs within time-expanded sequences according to spell length rather than treating spells as explicit units of comparison; prior work has also noted its counterintuitive behaviour and lack of guaranteed metric properties (Studer et al. 2016).

The key shift is therefore not simply from positions to spells, but from fixed duration-adjusted DSS-prefix measures to a more flexible spell-prefix framework. LCPspell compares spells as structural units of trajectories and allows duration differences to be switched off or weighted explicitly through a reference-scaled parameter. This structure clarifies how LCP-type measures relate to timing, sequencing, duration, and the measurement of divergence and convergence.

2.5.1 LCP: Position-wise prefix comparison without explicit duration

Basic LCP compares sequences position by position from the start. Let two state sequences be denoted by

$$\mathbf{x} = (x_1, \ldots, x_n), \qquad \mathbf{y} = (y_1, \ldots, y_m), \tag{13}$$

where each element represents the observed state at a given time position. Here, n and m denote the lengths of the time-expanded sequences $\mathbf{x}$ and $\mathbf{y}$, not spell counts. The common prefix length L is the number of consecutive positions from the start at which both sequences have the same state:

$$L = \max\{k : x_1 = y_1, \ldots, x_k = y_k,\ k \leq \min(n, m)\}. \tag{14}$$

The raw LCP distance is

$$d_{\mathrm{LCP}}(\mathbf{x}, \mathbf{y}) = n + m - 2L, \tag{15}$$

which counts the unmatched positions beyond the common prefix. Duration is not represented as a separate component: each matched position contributes equally, and spell length matters only indirectly through the number of repeated positions in the time-expanded sequence. A simple normalization divides by the maximum possible distance $n + m$, yielding

$$d = \frac{n + m - 2L}{n + m} \in [0, 1]. \tag{16}$$

RLCP (reverse LCP) applies the same logic from the end of the sequence, using the longest common suffix:

$$L_{\mathrm{rev}} = \max\{k : x_{n-k+1} = y_{m-k+1}, \ldots, x_n = y_m,\ k \leq \min(n, m)\}, \tag{17}$$

with raw distance

$$d_{\mathrm{RLCP}} = n + m - 2L_{\mathrm{rev}}. \tag{18}$$

LCP is simple and computationally efficient: computing the common prefix requires checking at most $\min(n, m)$ aligned positions, giving a time complexity of $O(\min(n, m))$. It requires only a state matrix and is most appropriate when sequences are observed on a shared time grid and the focus is on how long trajectories remain identical from the beginning in calendar time.

2.5.2 LCPmst: DSS-based prefix comparison with minimal shared time

LCPmst extends LCP by incorporating duration into a prefix comparison of distinct successive states (DSS). In this representation, consecutive repetitions of the same state are collapsed into spells, and the prefix is computed over the ordered sequence of spell states rather than over repeated time positions. LCPmst then adjusts the shared prefix by Elzinga's principle of minimal shared time (MST), which defines the shared duration of two matched spell-state elements as the smaller of their two associated durations. The intuition is simple: if one sequence spends 5 years in a matched state and the other spends 2 years in the same matched state, the two sequences share 2 years in that state, not 5.

Let the common DSS-prefix length L be the number of consecutive matched spell states from the beginning. For each matched element $t = 1, \ldots, L$ in this common prefix, let $d_x(t)$ and $d_y(t)$ denote the durations associated with the corresponding spell-state elements in sequences $\mathbf{x}$ and $\mathbf{y}$, respectively. Here, t indexes elements of the DSS prefix, not repeated calendar-time observations in a time-expanded sequence. LCPmst then sums the minimal

shared duration across the common prefix:

$$A_{\text{mst}} = \sum_{t=1}^{L} \min(d_x(t), d_y(t)). \tag{19}$$

Thus, A_{mst} represents the amount of minimal shared time contained in the common DSS prefix.

Let

$$T_x = \sum_{t=1}^{n_s} d_x(t), \qquad T_y = \sum_{t=1}^{m_s} d_y(t)$$

denote the total durations of the two sequences, where n_s and m_s are the numbers of DSS elements (spells) being compared. The raw LCPmst distance is then

$$d_{\text{LCPmst}}(\mathbf{x}, \mathbf{y}) = T_x + T_y - 2A_{\text{mst}}. \tag{20}$$

This distance can be interpreted as the total duration in the two sequences that is not part of their minimally shared DSS prefix. For example, if both sequences match in the state "employed" at a prefix element, and the associated durations are 5 and 2 years, the shared contribution is $\min(5, 2) = 2$. Only the overlapping part of the two durations is counted as shared time. Because the shared component cannot exceed the duration of either sequence, the raw distance remains non-negative and bounded by $T_x + T_y$. A simple normalization therefore divides by $T_x + T_y$, although other normalization strategies can also be used.

RLCPmst applies the same principle from the end of the sequence, computing minimal shared time over the common DSS suffix rather than the common prefix. LCPmst is therefore appropriate when the comparison remains anchored to shared beginning or ending spell-state elements, and when the amount of minimally shared exposure within those matched elements is substantively important.

2.5.3 LCPprod: DSS-based prefix comparison with product duration

LCPprod is also a DSS-based duration-adjusted prefix measure, but it measures overlap through the product of durations rather than their minimum. Like LCPmst, it computes the common prefix over the ordered sequence of distinct successive states rather than over repeated time positions. Thus, the prefix considered in LCPprod is the DSS prefix, not the prefix of the time-expanded state sequence. Using the same total-duration notation as above, let $T_x = \sum_{t=1}^{n_s} d_x(t)$ and $T_y = \sum_{t=1}^{m_s} d_y(t)$. It then weights the matched prefix elements by multiplying the associated durations.

Its accumulation term is

$$A_{\text{prod}} = \sum_{t=1}^{L} d_x(t) \cdot d_y(t), \tag{21}$$

where L denotes the common DSS-prefix length, and $d_x(t)$ and $d_y(t)$ are the durations of the

matched spell-state elements. The raw distance is

$$d_{\mathrm{LCPprod}}(\mathbf{x}, \mathbf{y}) = T_x + T_y - 2A_{\mathrm{prod}}. \tag{22}$$

The key difference from LCPmst lies in how matched duration is rewarded. LCPmst uses $\min(d_x(t), d_y(t))$, whereas LCPprod uses $d_x(t) \cdot d_y(t)$. Unlike the minimum, this product is not bounded by the smaller duration and can grow quickly when both durations are large. Consequently, A_{prod} may exceed the total duration scale used for normalization, and the raw value $T_x + T_y - 2A_{\mathrm{prod}}$ can become negative. This conflicts with the usual interpretation of the value as a proper dissimilarity. Implementations may clamp normalized values to $[0, 1]$, but clamping collapses distinct raw distances and reduces interpretability. In short, LCPprod rewards long matched durations more strongly than LCPmst, but it is less intuitive and less stable in practice. As noted by Elzinga (2007), it is mainly of theoretical rather than practical interest.

2.5.4 LCPspell: Spell-wise prefix comparison with optional reference-scaled duration

LCPspell applies LCP logic to spell sequences. It compares the ordered sequence of spells rather than positions on a common time axis, and it allows duration differences within the matched prefix or suffix to enter as an optional reference-scaled penalty. This makes LCPspell different from both position-wise LCP and the duration-adjusted DSS variants LCPmst and LCPprod.

The distinction concerns how duration is linked to prefix comparison. In DSS-based variants, consecutive repetitions of the same state are collapsed, and the common prefix is computed over the ordered list of distinct successive states. Duration is then used to adjust the amount of shared prefix, either through minimal shared time in LCPmst or through product duration weighting in LCPprod. In this sense, LCPmst and LCPprod already represent spell-oriented extensions of LCP, but they incorporate duration through fixed accumulation rules.

LCPspell builds on this DSS-prefix insight but separates two components that are analytically distinct. First, it asks whether two trajectories share the same ordered spell-state prefix. Second, it allows the durations of the matched spells to contribute through a separate, optional penalty. This penalty is controlled by the parameter λ and scaled by a fixed reference duration τ. When $\lambda = 0$, LCPspell reduces to a pure spell-order prefix measure. When $\lambda > 0$, duration differences within matched spells are incorporated gradually. This provides a more flexible way to decide whether trajectories with the same spell order but different spell lengths should be treated as identical or partially dissimilar.

Let two spell sequences be denoted by

$$\mathbf{s} = \big((a_1, d_1), \ldots, (a_{n_s}, d_{n_s})\big), \qquad \mathbf{r} = \big((b_1, e_1), \ldots, (b_{m_s}, e_{m_s})\big), \tag{23}$$

where a_k and b_ℓ denote spell states, d_k and e_ℓ denote their corresponding durations, and n_s and m_s denote the numbers of spells in $\mathbf{s}$ and $\mathbf{r}$, respectively. These are spell counts, not the lengths of the corresponding time-expanded state sequences. The k-th spell of $\mathbf{s}$ is compared with the k-th spell of $\mathbf{r}$, and the two spells match if $a_k = b_k$. The common prefix length L is therefore the number of consecutive matched spells from the start:

$$L = \max\{k : a_1 = b_1, \ldots, a_k = b_k,\ k \leq \min(n_s, m_s)\}. \tag{24}$$

The raw LCPspell distance has two components: a spell-count component and an optional duration component. The spell-count component is identical in form to basic LCP:

$$d_{\text{spell-count}} = n_s + m_s - 2L. \tag{25}$$

For the matched prefix, define the reference-scaled duration component as

$$A_{\text{dur}} = \sum_{k=1}^{L} \frac{|d_k - e_k|}{\tau}, \tag{26}$$

where $\tau > 0$ is the fixed reference duration scale introduced in Section 2.4. In applications, τ can be set to the total observation window or to the maximum possible spell duration implied by the study design. It is fixed before distance computation and is not estimated from the observed maximum spell duration in the dataset. The raw LCPspell distance is then

$$d_{\text{LCPspell}}(\mathbf{s}, \mathbf{r}) = (n_s + m_s - 2L) + \lambda A_{\text{dur}}, \tag{27}$$

where $\lambda \geq 0$ is the duration-weight parameter, implemented as `expcost`. The reference scale τ is set through `duration_ref` with the same default and interpretation as in Section 2.4: `duration_ref` fixes scale, whereas `expcost` fixes weight.

When $\lambda = 0$, LCPspell compares only the ordered sequence of spell states. Two trajectories with the same spell order are then treated as identical in terms of spell-prefix structure, even if their transitions occur at different calendar times. When $\lambda > 0$, duration differences among matched prefix spells also contribute to the distance. Because duration differences are divided by the fixed reference duration τ, the duration penalty is reference-scaled: changing the observation unit from years to months does not change the ratio $|d_k - e_k|/\tau$, provided τ is recoded consistently.

For normalized LCPspell distances, the upper bound must include both the spell-count component and the duration component. This differs from OMspell normalization. In OMspell, duration expansion costs enter an optimal alignment problem and can be avoided if another edit path is cheaper. In LCPspell, by contrast, the duration penalty in Eq. 27 is part of the distance definition for matched prefix spells; there is no alternative edit path that can bypass

it. If τ is chosen as the maximum possible spell duration under the observation design, then $|d_k - e_k|/\tau \leq 1$ for each matched spell. Since $L \leq \min(n_s, m_s)$, a suitable upper bound is

$$d_{\max}^{\text{LCPspell}} = (n_s + m_s) + \lambda \min(n_s, m_s), \tag{28}$$

which can be used when a bounded normalized version is desired.

RLCPspell applies the same logic from the end, matching spells backward from the last spell to the first. It therefore captures shared terminal spell structure rather than shared initial spell structure. LCPspell is most useful when the substantive focus lies on spell structure rather than exact calendar timing, and when the researcher wants to decide explicitly whether duration differences should matter. Appendix C provides a worked illustration of how the four LCP variants compare Sequences 9 and 10 in Figure 1.

3 Sensitivity of Distance Measures to Sequence Variation: Simulation Design and Implementation

Having defined the four LCP variants and clarified their respective emphases on timing, sequencing, and duration, the remaining question is how these differences translate into their sensitivity to different types of sequence variation. In particular, to what extent do the measures respond to controlled variation in specific fundamental sources of sequence variation, and how sharply can they distinguish signal from background noise?

To address this, we conduct a systematic sensitivity analysis using a simulation design that isolates the key sources of sequence variation. This approach allows us to evaluate not only overall performance, but also the specific types of variation to which each measure is most responsive. This section specifies the simulation logic used to evaluate how different distance measures respond to controlled perturbations in sequencing, timing, duration, and divergence/convergence. Building on the strand-based sensitivity framework introduced by prior work (Studer 2012; Elzinga et al. 2015; Studer et al. 2016), we retain its core logic while extending and refining the design to accommodate spell-based and prefix/suffix-oriented distance measures.

3.1 Design Principles

Following particularly the simulation framework of Studer et al. (2016), sensitivity is evaluated strand by strand, with each strand designed to target one focal source of sequence variation. In each strand, we generate two groups of sequences under different conditions on the focal source, while allowing the remaining sources of variation to vary stochastically. The two groups therefore differ systematically in the targeted dimension, but sequences within each group are not identical; instead, they share the same focal design condition while still vary-

ing randomly in all non-target dimensions. For example, in the sequencing strand, Group 1 may be generated around the order A→B→C, whereas Group 2 may be generated around the order A→C→B. Within both groups, however, spell durations, transition timings, and other non-target features vary randomly. The focal sequencing contrast is therefore fixed at the group level, while background variation is introduced at the sequence level.

This design creates a controlled within–between contrast. Pairs of sequences drawn from the same group differ only because of background variation, whereas pairs drawn from different groups differ because of both background variation and the focal manipulation. Sensitivity is then evaluated by comparing the empirical distance distributions of within-group and between-group pairs. A distance measure is considered sensitive to the focal dimension if it assigns systematically larger distances to between-group pairs than to within-group pairs, despite the presence of stochastic variation in non-target dimensions.

3.2 Strand Construction

Building on the conceptual distinction introduced in the previous section, we treat timing, sequencing, and duration as the three core sources of sequence variation, and divergence/convergence as configurations that describe where similarity breaks down or reappears along a trajectory. The simulation design therefore contains two parts. First, we retain the three core strands of Studer et al. (2016): timing, sequencing, and duration. These strands isolate variation in calendar position, spell order, and spell length, respectively. Second, we introduce additional strands for divergence and convergence, defined at two comparison scales: calendar time and spell order.

Calendar-time divergence/convergence strands ask whether a measure is sensitive to when separation or alignment occurs along a shared time axis. Spell-order divergence/convergence strands ask whether a measure is sensitive to where separation or alignment occurs in the ordered sequence of spells. This distinction is central for prefix- and suffix-based measures, because LCP-type measures directly target shared beginnings, whereas RLCP-type measures target shared endings. The extended strand design therefore allows us to evaluate not only whether a measure responds to a given source of variation, but also whether it captures the location of divergence or convergence at the comparison scale relevant to the research question.

For data generation, each sequence is constructed in DSS–duration form. The distinct successive states (DSS) specify the ordered list of distinct states without repetition of consecutive identical states (Gabadinho et al. 2011), while the associated durations specify the length of each spell. For example, DSS = $[a, b, c]$ with durations $[5, 8, 7]$ corresponds to the spell sequence $((a, 5), (b, 8), (c, 7))$ with total length $T = 20$. In this representation, each spell is an ordered pair (a_k, d_k) consisting of a state and its duration.

This spell-based construction is used only as a generative device, not as a restriction on the dissimilarity measures being evaluated. It provides a parsimonious way to control spell

order, spell duration, and transition timing while keeping the total sequence length fixed. After generation, the same underlying trajectories can be represented either as time-expanded state sequences for position-wise measures such as Hamming, OM, LCP, and RLCP, or as spell sequences for spell-based measures such as OMspell, OMspellRS, LCPspell, and RLCPspell. Thus, position-wise and spell-based measures are evaluated on the same simulated trajectories; what differs is the representation on which each measure operates.

This construction also preserves timing information. Calendar-time strands are implemented by expanding the generated spell sequences onto a shared time axis and manipulating where separation or alignment occurs in time. Spell-order strands instead use the DSS structure directly and manipulate where separation or alignment occurs in the ordered sequence of spells, irrespective of exact calendar alignment.

The simulation framework proceeds as follows, and the following subsections provide more details in each step:

1. Specify global parameters: total sequence length T ($T = 20$), number of distinct states (five states a–e), number of sequences per group (2000), and number of independent replications (30).

2. Generate two groups of sequences according to strand-specific design rules, ensuring that only the focal source of variation differs systematically between groups.

3. Compute pairwise distance matrices using multiple distance measures, including LCP, RLCP, (R)LCPspell, OM, OMspell, OMspellRS, and Hamming. For brevity, spell-based OMspellRS variants are labelled `OMspellRS` in the simulation figures.

4. Evaluate how well each distance measure separates groups using a sensitivity metric.

For each strand, the full procedure is repeated across 30 independent replications with different random seeds. In each replication, we generate a new pair of groups, compute all pairwise distance matrices, and calculate the sensitivity metric for each method. Results are reported as means across replications, with standard deviations used to summarize simulation variability.

3.3 Distance Computation and Evaluation

For all OM-type measures, we use the same base cost setting: an indel cost of 1 and a constant substitution cost of 2. The methods include OM, OMspell, and OMspellRS. For OMspellRS, LCPspell, and RLCPspell, the reference duration scale is fixed at `duration_ref`$= T$, the total observation window, so that duration differences are expressed as proportions of the full sequence length. Original OMspell is evaluated using duration terms in raw time units and therefore does not use the `duration_ref` denominator. Sensitivity analysis is conducted over the duration-weight parameter λ (implemented as `expcost`), evaluated at 0,

0.10, 0.25, 0.50, 0.75, 1.00, 1.50, and 2.00. This reflects the distinction between scale and weight: `duration_ref` fixes the scale on which duration differences are measured, whereas `expcost` controls how strongly those scaled differences enter the distance. We further include LCP and RLCP (position-wise prefix/suffix), LCPmst and RLCPmst (DSS-based prefix/suffix measures with minimal shared time), and LCPspell and RLCPspell (spell-based prefix/suffix measures with optional reference-scaled duration penalties), with LCPspell and RLCPspell evaluated over the same `expcost` grid. Finally, Hamming distance is included as a simple position-wise benchmark, since it compares states only at identical time positions and thus provides a useful reference for the timing-sensitive end of the comparison spectrum.

To evaluate how well each distance method distinguishes between groups, we compute a chance-corrected pseudo-R^2 metric. Let D be the $n \times n$ distance matrix, where n is the total number of sequences from both groups. The total discrepancy is defined as the sum of all pairwise distances in the upper triangle,

$$\text{total} = \sum_{i<j} D_{ij}, \tag{29}$$

and the between-group discrepancy as

$$\text{between} = \sum_{i \in G_1, j \in G_2} D_{ij}. \tag{30}$$

The standard pseudo-R^2 used in Studer et al. (2016) is defined as

$$R^2 = \frac{\text{between}}{\text{total}}, \tag{31}$$

that is, the share of total pairwise discrepancy contributed by pairs drawn from different groups. However, this raw share has a non-zero baseline even when group labels are unrelated to the distance structure. By random labeling, we mean assigning group labels independently of the sequence distances while preserving the same group sizes. Under such random labeling, some pairs will still be between-group pairs simply because the sample is partitioned into two groups. For example, when the two groups are of equal size, approximately one half of all unordered pairs are between-group pairs. The raw ratio between/total therefore does not by itself indicate separation beyond chance.

We therefore apply a chance correction analogous to the logic of Cohen's kappa, which rescales an observed quantity by subtracting its chance-expected value and dividing by the maximum possible improvement beyond chance. Let the expected proportion of between-group pairs under random labeling be

$$p_{\text{exp}} = \frac{n_0 n_1}{\binom{n}{2}}, \tag{32}$$

where n_0 and n_1 denote group sizes. The chance-corrected pseudo-R^2 is then defined as

$$\text{pseudo-}R^2 = \frac{\frac{\text{between}}{\text{total}} - p_{\text{exp}}}{1 - p_{\text{exp}}}. \tag{33}$$

This correction rescales the raw between-group share relative to its chance baseline. A value of zero indicates no discrimination beyond what would be expected from the group sizes alone, negative values indicate worse-than-random separation, and a value of one corresponds to perfect group separation. The metric is therefore normalized to represent improvement over chance rather than the raw between-group share, making it more interpretable for comparative evaluation of distance measures.

Pseudo-R^2 values are interpreted comparatively within each simulation strand and are not compared across strands, since baseline discrepancies differ by construction. This ensures that results reflect relative sensitivity patterns rather than strand-specific absolute magnitudes.

3.4 Methodological Refinements

Compared to the simulation framework of Studer et al. (2016), we retain the core strand-based logic but adapt it to the aims of this study. In particular, our comparison is deliberately focused on OM-based and LCP-based families, because the objective is to evaluate the proposed (R)LCPspell framework and its relation to OM-style alignment methods. We therefore do not re-benchmark the full set of dissimilarity measures considered in prior reviews; instead, we provide a targeted comparison of methods that are most directly relevant to spell-based prefix and suffix comparison. The main refinements are summarized in Table 4 and briefly described below.

Unified normalization. While Studer et al. (2016) compare measures using raw distances, we use the reference-based normalization proposed by Elzinga et al. (2019) in the main analysis. The purpose of this step is practical: the methods compared in this paper operate on different representations and produce distances on different raw scales. A common normalization makes the sensitivity profiles easier to compare across methods, while the interpretation of each measure remains tied to its own comparison logic.

Elzinga et al. (2019) define a normalized distance with respect to a fixed reference object r as

$$D_r(x, y) = \frac{d(x, y)}{\big(d(x, y) + d(x, r) + d(y, r)\big)/2}. \tag{34}$$

This normalization maps metric distances to a dimensionless scale and is most directly justified when the underlying distance satisfies the triangle inequality. For this reason, Appendix D discusses the metric status of the measures used in the paper. LCP, RLCP, and the pure spell-order versions of LCPspell and RLCPspell satisfy the relevant metric conditions. By contrast, duration-augmented LCPspell and RLCPspell should be interpreted more cautiously

as reference-scaled dissimilarities, because their duration penalty is evaluated only over the matched prefix or suffix.

In the simulations, the normalized values are therefore used for comparative sensitivity analysis rather than for claims that every parameterized measure is a metric. The reference object is used only to place distances on a common scale and does not enter the group-generating process. We choose the first sequence in each replication as a deterministic reference object so that the same normalization rule is applied consistently across methods, strands, and replications. As a robustness check, Appendix E.1 reports results using method-specific built-in normalizations; the main sensitivity patterns remain substantively unchanged.

Focused parameter grids.

For OM-type measures, we use a common baseline cost setting with an indel cost of 1 and a constant substitution cost of 2. Under this setting, a substitution is equivalent to one insertion followed by one deletion, giving a standard neutral baseline for OM-style comparisons (Studer et al. 2016). We then focus parameter exploration on the duration-weight parameter (expcost) that is central to the spell-based methods. This design differs from Studer et al. (2016) in emphasis. Their review paper compares a broad set of dissimilarity measures and selected parameter configurations, whereas our study uses a narrower method set but a more detailed parameter grid for the duration-weight mechanism most relevant to (R)LCPspell and OMspell-type measures. This allows us to examine how sensitivity changes as duration receives increasing weight, rather than treating each parameter setting only as a separate method configuration.

3.5 Details of each strand

Together, the seven strands cover two levels of comparison. Strands 1–3 isolate the three core sources of sequence variation: timing, sequencing, and duration. Strands 4–7 operationalize divergence and convergence as configurations of separation and alignment, defined either in calendar time or in spell order. Table 5 summarizes the design of each strand, including the focal contrast, the group-level manipulation, and intuitive example trajectories. The examples in Table 5 are schematic illustrations of the group-level contrasts. In the actual simulations, sequences are generated stochastically: the focal manipulation is fixed by the strand design, while non-target dimensions are allowed to vary as background noise. Thus, the table should be read as describing the intended contrast between groups rather than as listing the exact sequences used in the simulations.

Two design choices are worth noting. First, the sequencing strand includes several variants because sensitivity to ordering may depend on both the scale and the location of the ordering change. Complete reversal creates a strong global contrast, whereas local permutations and early or late swaps introduce more localized contrasts. Second, the spell-order divergence and convergence strands manipulate where separation or alignment occurs in the sequence

of spells while keeping duration outside the focal manipulation. This preserves the distinction between duration as spell length, tested in Strand 3, and divergence/convergence as the location of separation or alignment, tested in Strands 6–7.

Table 4: Comparison of simulation design: This study versus Studer & Ritschard (2016)

Aspect	Studer & Ritschard (2016)	This study	Rationale for differences
Core framework	Strand-based sensitivity analysis	**Same**: strand-based sensitivity analysis	—
Core strands	Three strands: timing, sequencing, duration	**Same**: timing, sequencing, duration strands retained	—
Design approach	Mixed perturbation: non-target sources vary randomly	**Same**: mixed-perturbation logic retained	—
Sequence length	$T = 20$	**Same**: $T = 20$	—
Sequence generation	Spell-based (DSS + durations)	**Same**: spell-based (DSS + durations)	—
Timing strand	Focal state covers $t = 7$ vs. $t \in \{9, \ldots, 15\}$ (2–8 unit differences)	**Same**: focal-state timing contrast retained	—
Sequencing strand	Multiple DSS patterns tested	**Different**: expanded to four explicit patterns (complete reversal, local permutation, early swap, late swap)	Sequencing differences can vary in both scope and location; these variants distinguish broad order reversals from localized swaps and test whether sensitivity depends on where the ordering change occurs.
Duration strand	Focal state duration controlled, DSS randomly selected	**Same**: focal-duration contrast retained	—
Evaluation metric	Standard pseudo-R^2: $\frac{\text{between}}{\text{total}}$	**Different**: chance-corrected pseudo-R^2	Accounts for baseline expectations; enables interpretation of zero and negative values
Sequences per group	1 million (unique sequences weighted by counts)	**Different**: 2,000 sequences per group	Reduces the computational cost of repeated pairwise distance calculations for the expanded method set; stability is assessed through 30 independent replications
Replications	Single run (unique sequences)	**Different**: 30 independent replications with different random seeds	Enables uncertainty quantification (mean and std); assesses robustness across random seeds
Additional strands	None	**Different**: divergence and convergence strands added at two comparison scales: calendar time and spell order	Test LCP/RLCP-specific features; distinguish calendar-time from spell-order definitions of divergence/convergence

Continued on next page

Table 4: Comparison of simulation design: This study versus Studer & Ritschard (2016) (continued)

Aspect	Studer & Ritschard (2016)	This study	Rationale for differences
Parameter exploration	Broad method set with selected discrete parameter configurations	**Different**: focused parameter grid for spell-based duration weights (0, 0.10, 0.25, 0.50, 0.75, 1.00, 1.50, and 2.00), with OM-type base costs fixed at indel $= 1$ and substitution $= 2$	Examine how sensitivity changes as duration weight increases while keeping the baseline alignment costs comparable across OM-type methods
Normalization	No normalization (raw distances used)	**Different**: unified normalization (Elzinga et al. 2019)	Enables direct comparison across methods; ensures interpretable absolute values; addresses limitations of other normalization methods identified by Elzinga et al. (2019); complementary normalization checks are reported in Appendix E.1
Comparability extension checks	Included event-based and small-perturbation evaluations	**Same**: event-based and small-perturbation checks are retained, shown in Appendix E; **Different in reporting scope**: presented as a dedicated comparability-extension appendix tailored to the OM/LCP-focused method set	Demonstrate that core ranking patterns are not confined to a single spell/state simulation mechanism while preserving the paper's focused method comparison

Table 5: Design and intuitive examples for all seven simulation strands

Strand	Dimension tested	Group difference	Example trajectories
1	Timing (calendar position)	Same focal state occurs at different time points	Group 1: $a\,a\,\mathbf{b}\,b\,c\,c\,d\,d$ (state b at $t = 3$); Group 2: $a\,a\,a\,a\,\mathbf{b}\,b\,c\,c$ (state b appears later in time).
Strand 2: Sequencing sensitivity (same states, different order; timing and durations vary randomly)			
2a	Complete reversal	Reverse the full order of distinct successive states (DSS)	Group 1: $a\,b\,c\,d\,e$; Group 2: $e\,d\,c\,b\,a$.
2b	Local permutation	Swap two neighbouring states in the middle	Group 1: $a\,\mathbf{b}\,\mathbf{c}\,a$; Group 2: $a\,\mathbf{c}\,\mathbf{b}\,a$.
2c	Early swap	Swap the first two states	Group 1: $\mathbf{a}\,\mathbf{b}\,c\,d\,e$; Group 2: $\mathbf{b}\,\mathbf{a}\,c\,d\,e$.
2d	Late swap	Swap the last two states	Group 1: $a\,b\,c\,d\,\mathbf{e}$; Group 2: $a\,b\,c\,\mathbf{e}\,d$.
3	Duration (spell length)	Same spell order, different spell lengths	Group 1: $(a, 2)(\mathbf{b}, \mathbf{4})(c, 2)$; Group 2: $(a, 2)(\mathbf{b}, \mathbf{12})(c, 2)$.
4	Divergence (calendar time)	Trajectories diverge early vs. late along the timeline	**Early calendar-time divergence:** $a\,a\,\mathbf{b}\,\mathbf{b}\,\mathbf{c}\,\mathbf{c}\,\mathbf{d}\,\mathbf{d}$ vs. $a\,a\,\mathbf{e}\,\mathbf{e}\,\mathbf{f}\,\mathbf{f}\,\mathbf{g}\,\mathbf{g}$ **Late calendar-time divergence:** $a\,a\,b\,b\,c\,c\,\mathbf{d}\,\mathbf{d}$ vs. $a\,a\,b\,b\,c\,c\,\mathbf{e}\,\mathbf{e}$.
5	Convergence (calendar time)	Trajectories align early vs. late along the timeline	**Early calendar-time convergence:** $a\,b\,\mathbf{c}\,\mathbf{c}\,\mathbf{d}\,\mathbf{d}\,\mathbf{e}\,\mathbf{e}$ vs. $f\,g\,\mathbf{c}\,\mathbf{c}\,\mathbf{d}\,\mathbf{d}\,\mathbf{e}\,\mathbf{e}$ **Late calendar-time convergence:** $a\,b\,c\,e\,f\,g\,\mathbf{d}\,\mathbf{d}$ vs. $h\,i\,j\,k\,l\,m\,\mathbf{d}\,\mathbf{d}$.
6	Divergence (spell order)	Trajectories diverge at different spell positions	**Early spell divergence:** $\mathbf{a}\,b\,c\,d$ vs. $\mathbf{e}\,b\,c\,d$ **Late spell divergence:** $a\,b\,\mathbf{c}\,d$ vs. $a\,b\,\mathbf{e}\,d$
7	Convergence (spell order)	Different number of shared terminal spells	**Share last 1 spell:** $a\,b\,c\,\mathbf{d}$ vs. $e\,f\,g\,\mathbf{d}$ **Share last 3 spells:** $a\,\mathbf{c}\,\mathbf{d}\,\mathbf{e}$ vs. $b\,\mathbf{c}\,\mathbf{d}\,\mathbf{e}$

4 Results of Sensitivity Patterns Across Distance Measures

The simulation experiments reveal distinct sensitivity patterns across distance measures, providing guidance on which methods are most appropriate for different research questions. Figure 2 summarizes the findings across all experimental strands using the unified reference-based normalization proposed by Elzinga et al. (2019), which places measures on a common scale within each strand. Taller bars indicate higher sensitivity to the focal source of variation; the chance-corrected pseudo-R^2 ranges from 0 (no group separation beyond chance) to 1 (perfect separation), with negative values indicating performance below chance. In each panel, the x-axis lists the distance measures and their parameter settings. OM refers to the standard configuration with indel cost 1 and substitution cost 2. For spell-based measures, numeric labels indicate the value of the expansion-cost parameter `expcost`; for example, `LCPs 0.00` denotes LCPspell with `expcost` equal to 0, while larger values give greater weight to duration differences.

The figure should be interpreted primarily in terms of relative sensitivity within each panel rather than the absolute magnitude of the bars. Each strand is generated under a different contrast and therefore has a different baseline level of separability. For this reason, the most informative comparison is which measures are more or less sensitive than others under the same experimental condition. To keep small but substantively meaningful differences visible, the y-axis is scaled to the data range in panels where values are small.

Moreover, Studer et al. (2016) summarize their simulation results by reducing several strand-specific sensitivity scores to two composite axes: one contrasting sequencing with temporality, and the other contrasting duration with timing. This projection is useful for providing an overall map of many dissimilarity measures in a review paper, but it is less suitable for the purpose of the present study. Our design does not only compare sensitivity to timing, sequencing, and duration; it also introduces divergence and convergence strands in both calendar-time and spell-order formulations. For prefix- and suffix-based measures, the relevant information is precisely strand-specific: whether divergence occurs early or late, whether convergence is assessed from the beginning or the end, and whether the comparison is made in calendar time or in spell order. Collapsing these conditions into two composite axes would obscure these distinctions. We therefore report strand-specific bar plots, which display the pseudo-R^2 values directly for each experimental condition and allow methods to be compared within the same strand.

Appendix E reports three robustness checks: (1) a normalization check (built-in method-specific normalization vs. unified reference-based normalization), (2) a scoring check (chance-corrected pseudo-R^2 vs. raw pseudo-R^2 on all strands), and (3) a design check (comparability-extension strands based on event-based and small-perturbation generators, reported under both corrected and raw scoring). Across these checks, the broad substantive patterns remain visible, although absolute values and some relative rankings change when different normalization

conventions are used. The robustness checks should therefore be interpreted as evidence that the main qualitative conclusions do not depend on a single scaling convention or simulation generator, rather than as evidence that every cross-method ranking is invariant.

4.1 Main Findings

Timing (Panel A). When sequences differ only in when a state occurs, all methods show very low sensitivity. Even the best-performing measure, Hamming distance, yields only marginal separation. This indicates that, under the mixed-perturbation design used here, pure timing shifts are difficult to detect once sequencing and duration vary stochastically in the background. Timing sensitivity is therefore not simply a property of comparing sequences on a common calendar axis; it also depends on whether the timing signal remains distinguishable from variation in order and spell length.

Sequencing (Panels B–E). When the focal difference concerns the order of states, spell-based measures with 0 expcost achieve the strongest separation. In this setting, duration-related terms vanish, so OMspell, OMspellRS, LCPspell, and RLCPspell reduce to comparisons of the ordered sequence of spell states. This removes duration as a source of within-group noise and makes the sequencing signal especially clear.

As expcost increases, sensitivity to sequencing declines. This does not mean that larger expcost values are intrinsically worse; rather, in the sequencing strands, duration is deliberately introduced as background variation. Increasing expcost gives this non-target duration variation more influence in the distance calculation, thereby increasing within-group distances and diluting the between-group sequencing contrast. The response to sequencing also depends on where the ordering change occurs: LCP and RLCP show asymmetric behavior for early and late swaps, while OM-based measures show more moderate but less location-specific sensitivity.

Duration (Panel F). The duration strand clarifies the role of expcost. When expcost is 0, spell-based measures ignore duration differences and show near-zero sensitivity to the duration manipulation. As expcost increases, duration differences contribute more strongly to the distance, and some spell-based measures become more sensitive to variation in spell length, although the magnitude of this response differs between OMspell, OMspellRS, and LCPspell because normalization changes the scale on which duration penalties enter the final distance. At the same time, duration sensitivity is not limited to spell-based methods. Position-wise methods such as Hamming distance and standard OM can also respond to duration differences because longer spells generate repeated observations on the time-expanded sequence. The difference is that, in position-wise methods, duration sensitivity arises through repetition, whereas in spell-based methods it is controlled explicitly through the duration-weight parameter.

Divergence and convergence in calendar time (Panels G–H). When divergence and convergence are defined along the calendar-time axis, methods that aggregate differences over time, especially OM-based measures, tend to perform well. This reflects their ability to capture

accumulated differences across the full trajectory rather than relying only on a single prefix or suffix breakpoint. The convergence strand also shows that suffix-oriented measures become more relevant when the signal concerns terminal alignment. In particular, RLCP and RLCPmst respond more strongly than their prefix-oriented counterparts. RLCPspell also becomes more sensitive as duration receives greater weight, although whole-trajectory OM-based measures remain stronger under this calendar-time design.

Divergence and convergence in spell order (Panels I–L). When divergence and convergence are defined by the position of differences in the ordered sequence of spells, prefix- and suffix-based spell measures provide a more direct directional interpretation. LCPspell responds more strongly when the relevant separation concerns the beginning of the spell sequence, whereas RLCPspell responds more strongly when the signal concerns the terminal spell structure. OMspell-type measures may still achieve high overall separation in some panels because they summarize dissimilarity across the complete spell sequence. The distinctive contribution of LCPspell and RLCPspell is therefore not universal superiority, but their ability to encode explicitly whether similarity breaks down or reappears at the beginning or the end of a pathway.

OMspell versus OMspellRS. OMspell and OMspellRS use the same spell-level alignment logic and the same OMspell cost convention of Studer et al. (2016). They differ only in how duration terms are scaled. Original OMspell expresses duration costs in raw time units, so the effect of `expcost` depends on the unit of measurement and on the distribution of spell lengths. OMspellRS divides duration terms by a fixed reference duration τ, such as the total observation window or the maximum possible spell duration implied by the study design. The two methods often display similar qualitative sensitivity patterns, but OMspellRS provides a more stable interpretation of `expcost` across datasets and time scales.

4.2 Method Selection Guidance

These results show that method choice should be guided by the substantive definition of similarity. No single measure is uniformly best across all strands. Instead, each measure becomes informative when its comparison logic matches the type of sequence variation that the researcher aims to capture.

If the research question concerns the order of states, and duration differences are not substantively central, spell-based measures with 0 expcost or low expcost values are appropriate. In this setting, the comparison focuses primarily on the ordered sequence of spell states, while reducing the influence of random variation in spell length. This is particularly useful when trajectories are understood as pathways, and when the timing or duration of spells is secondary to the order in which states occur.

If duration is substantively central, expcost should be set above zero so that spell-length differences enter the distance explicitly. Larger values of expcost increase duration sensitivity,

but their interpretation should remain relative to the scale of the other cost components and to the empirical distribution of spell lengths. OMspellRS is preferable when the researcher wants the duration weight to be interpretable on a common reference scale across applications.

If the research question concerns divergence or convergence in calendar time, position-wise and OM-based measures are generally more appropriate because they compare trajectories along a common time axis and aggregate differences across the full sequence. If, however, divergence or convergence is defined in terms of pathway structure, such as how many spells two trajectories share at the beginning or at the end, LCPspell and RLCPspell are more appropriate. LCPspell directly captures shared beginnings in spell space, while RLCPspell captures shared endings.

It is also important to distinguish general sensitivity from direct operationalization. OM-based measures may show sensitivity to divergence and convergence in calendar time because these patterns contribute to overall trajectory dissimilarity. In this sense, they remain useful general-purpose measures, especially when the researcher does not have a strong prior focus on a particular source of variation, or when the relevant source of variation is not pronounced in the sequence data. However, their sensitivity does not directly indicate where similarity breaks down or reappears; it reflects the accumulated dissimilarity over the aligned trajectory. When the substantive question concerns divergence or convergence, prefix- and suffix-based measures are more interpretable because they are designed to measure shared beginnings and endings of sequences directly. Their spell-based versions are especially useful when researchers want to separate shared spell structure from exact calendar timing and to decide explicitly whether duration differences should matter. The empirical demonstration below illustrates how these distinctions affect clustering outcomes in a substantive application.

This point can be stated in terms of directionality. OM and related whole-trajectory measures are symmetric summaries of overall dissimilarity. They can indicate that two sequences are different, but they do not, by themselves, distinguish whether the difference reflects early divergence, late convergence, or a temporally reversed version of the same pattern. Hamming distance also aggregates mismatches over aligned positions and therefore does not directly encode whether mismatches occur at the beginning, middle, or end of the trajectory. By contrast, LCP and RLCP introduce direction into the comparison: LCP measures shared beginnings, whereas RLCP measures shared endings. LCPspell and RLCPspell extend this directional logic to ordered spells, making it possible to distinguish calendar-time alignment from spell-order similarity.

In this sense, the choice of a distance measure should be linked to the theoretical object of comparison: cumulative exposure suggests whole-trajectory measures; critical or sensitive periods suggest calendar-aligned measures; pathway order suggests spell-based measures; and divergence or convergence suggests directional prefix- or suffix-based measures.

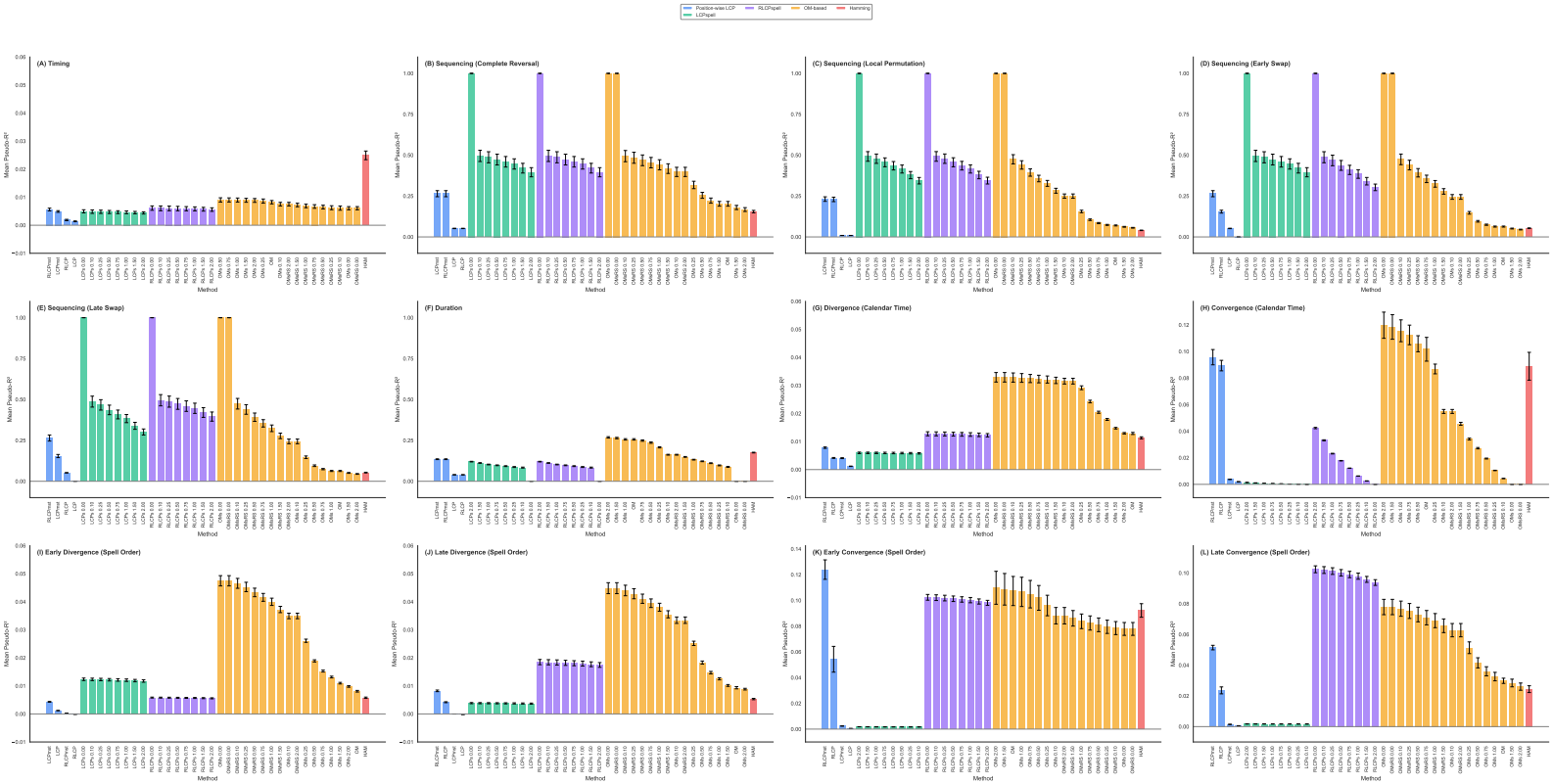


Figure 2: Sensitivity profiles across all methods and experimental strands using unified reference-based normalization (Elzinga et al. 2019).

5 Empirical Demonstration

In this section, we provide an empirical comparison of OM, OMspell, OMspellRS, LCP, and LCPspell. We use the pairfam dataset (Brüderl et al. 2020) prepared by Raab et al. (2022) containing 1,027 family formation trajectories of German individuals. Family formation trajectories provide a suitable setting, as they often exhibit divergence from a common initial state (single) into distinct pathways such as cohabitation and marriage (Cherlin 2004; Fasang 2014; Raab 2017), which aligns well with the type of divergence captured by prefix-based measures.

Based on this dataset, we conduct dissimilarity-based clustering, a widely used sequence analysis approach in empirical research. We first compute pairwise dissimilarities between all trajectories, then apply clustering based on the resulting dissimilarity matrix (using Ward's hierarchical clustering), and finally visualize the clustering solutions. To assess how different dissimilarity measures shape clustering outcomes, we compare Hamming distance, OM, OMspell, OMspellRS, LCP, and LCPspell. For OM, we use a standard parameterization with an indel cost of 1 and a substitution cost of 2 (`sm` in the Figure) with reasons explained in Section 3.4. Hamming distance is included as a simple position-wise benchmark, since it compares states only at identical time positions and provides a reference for the timing-sensitive end of the comparison spectrum. For spell-based measures, we vary the expansion-cost parameter, using values of 0.5 and 2. This parameter scales the duration-related terms in the spell-level cost function: increasing it makes the distance more sensitive to differences in spell duration, whereas decreasing it reduces the relative influence of duration and makes spell-state alignment more prominent. We use these two values as illustrative lower- and higher-duration-sensitivity settings.

Figure 3 presents the resulting clustering solutions for the pairfam dataset. Here, we retain a four-cluster solution across all methods to ensure comparability. The number of clusters is determined based on the clustering quality plot of each measure, where the average silhouette width (ASW) indicates an optimal solution at four clusters. Each row in Figure 3 corresponds to a specific dissimilarity measure and parameter configuration. Within each row, the panels display index plots for the resulting clusters. The labels above each panel indicate the cluster identifier, the number of sequences assigned to the cluster, and the corresponding sample weight, as the pairfam data are weighted. The key objective of the figure is to assess whether different dissimilarity measures produce distinct cluster structures, or similar clusters with internal variations. Such differences provide insight into how alternative measures emphasize different dimensions of sequence similarity.

Overall, different dissimilarity measures produce broadly similar clusters, while still yielding distinct variations in how trajectories are grouped, reflecting their different emphases on sequence similarity. Within the spell-based measures, however, the differences between parameter settings are relatively modest for OMspell and OMspellRS. In particular, varying the expcost parameter of OMspellRS from 0.5 to 2 does not substantially alter the overall cluster

structures. The Hamming solution is also visually close to the OM solution with indel cost 1 and substitution cost 2, suggesting that, in this empirical example, the additional alignment flexibility of OM does not substantially alter the main clustering pattern relative to a strict position-wise comparison.

The most visible contrast is therefore not between Hamming and OM, but between whole-trajectory measures and prefix-based measures. Hamming, OM, OMspell, and OMspellRS summarize dissimilarity across the full observation window, either by comparing identical time positions, by allowing alignment through indels, or by aligning spells based on the edit operations. Their clustering solutions tend to group individuals according to broad trajectory patterns over the entire sequence. By contrast, LCP-type measures place greater emphasis on the beginning of trajectories: LCP compares how long two sequences remain identical from the start, while LCPspell compares how long they share the same initial spell structure. The resulting clusters therefore reveal different aspects of family formation trajectories. Whole-trajectory measures are more informative about overall trajectory profiles, whereas LCP-type measures are more informative about how initially similar trajectories begin to diverge.

This difference is reflected in the cluster solutions. Under the whole-trajectory measures, one cluster is characterized by a strong concentration in the terminal state "married with two or more children" (gray color). This pattern is captured because these measures aggregate similarity and dissimilarity across the entire sequence, including later-life states. Under LCP, however, such a cluster does not appear in the same form, because the measure primarily evaluates divergence from the beginning of the sequence. Instead, all four LCP clusters eventually contain substantial shares of individuals who move into "married with two or more children", but they differ in how they diverge from the initial state. For example, the first three LCP clusters all involve transitions from "single without children" to "cohabiting without children", but the timing of this transition differs. In Cluster 1, the initial single state is relatively short and individuals move into cohabitation early. In Cluster 2, individuals remain single without children for at least the first 59 months before gradually entering cohabitation. Cluster 3 follows a similar pathway but with a shorter initial single period than Cluster 2. Cluster 4 differs more sharply, as it contains trajectories that begin directly in cohabitation without a preceding spell of being single without children.

The comparison between LCP and LCPspell further clarifies the role of data representation. Because LCP compares sequences position by position, it is especially sensitive to the exact timing of early divergence. This helps explain why LCP more clearly isolates the cluster in which individuals remain "single without children" for a long initial period before gradually entering "cohabiting without children". LCPspell, by contrast, compares trajectories at the spell level and is less sensitive to the exact calendar timing of transitions. As a result, it does not isolate this timing-specific pattern as clearly.

However, LCPspell captures a different feature that is less visible under LCP. Across both expcost settings, LCPspell identifies clusters containing recurrent spell-order patterns involving

a return to “single without children”. In particular, one LCPspell cluster contains trajectories that move from “single without children” to “cohabiting without children” and then back to “single without children”. A related pattern appears in another cluster, where individuals begin in “cohabiting without children” and subsequently move into “single without children”. These recurrent spell-order patterns are not separated as clearly by LCP; they appear only partially, for example among a small subset of sequences in one LCP cluster. This illustrates the main empirical implication of the proposed distinction: position-wise LCP is more sensitive to exact early timing, whereas LCPspell is better suited to identifying divergence in the ordered structure of spells, even when the exact timing of transitions varies.

6 Discussion and Conclusions

This study develops a spell-based perspective on sequence comparison through three main steps. First, it introduces a conceptual framework that links sequence representations to dissimilarity measures, distinguishing between position-wise and spell-based formulations. In this way, we highlight that sequence dissimilarity is shaped not only by the distance formula, but also by the data representation on which the formula operates. Second, it extends OMspell by rescaling duration terms with a fixed reference duration, yielding OMspellRS. This extension preserves the OMspell alignment logic of Studer et al. (2016) while making the duration-weight parameter easier to compare across time scales. Third, it proposes spell-based prefix and suffix distances, LCPspell and RLCPspell, which capture divergence and convergence at the level of spells rather than through exact calendar-time alignment. By separating timing, sequencing, and duration as core sources of sequence variation, and treating divergence and convergence as configurations of separation and alignment, the proposed framework clarifies how each method emphasizes particular aspects of trajectories and provides a more principled basis for method selection.

The first contribution is conceptual. While prior work has distinguished timing, sequencing, and duration as three important life-course dimensions (Liao et al. 2022; Studer et al. 2016), we conceptualize them as key sources of sequence variation to provide a more nuanced understanding of sequence comparisons with examples. Moreover, existing literature has shown that different dissimilarity measures highlight different aspects of sequence similarity (Studer et al. 2016); this paper adds that these differences are rooted not only in distance formulas, but also in the representation of the sequence object itself. In position-wise measures, the comparison is anchored to a shared time scale, and duration enters indirectly through repeated observations. In spell-based measures, spells become the units of comparison, and duration can be incorporated explicitly. This distinction matters because the same pair of trajectories can be interpreted differently depending on whether similarity is defined in calendar time or in spell structure. OMspellRS illustrates the methodological value of this perspective by making duration sensitivity explicit, reference-scaled, and compatible with bounded normalization.

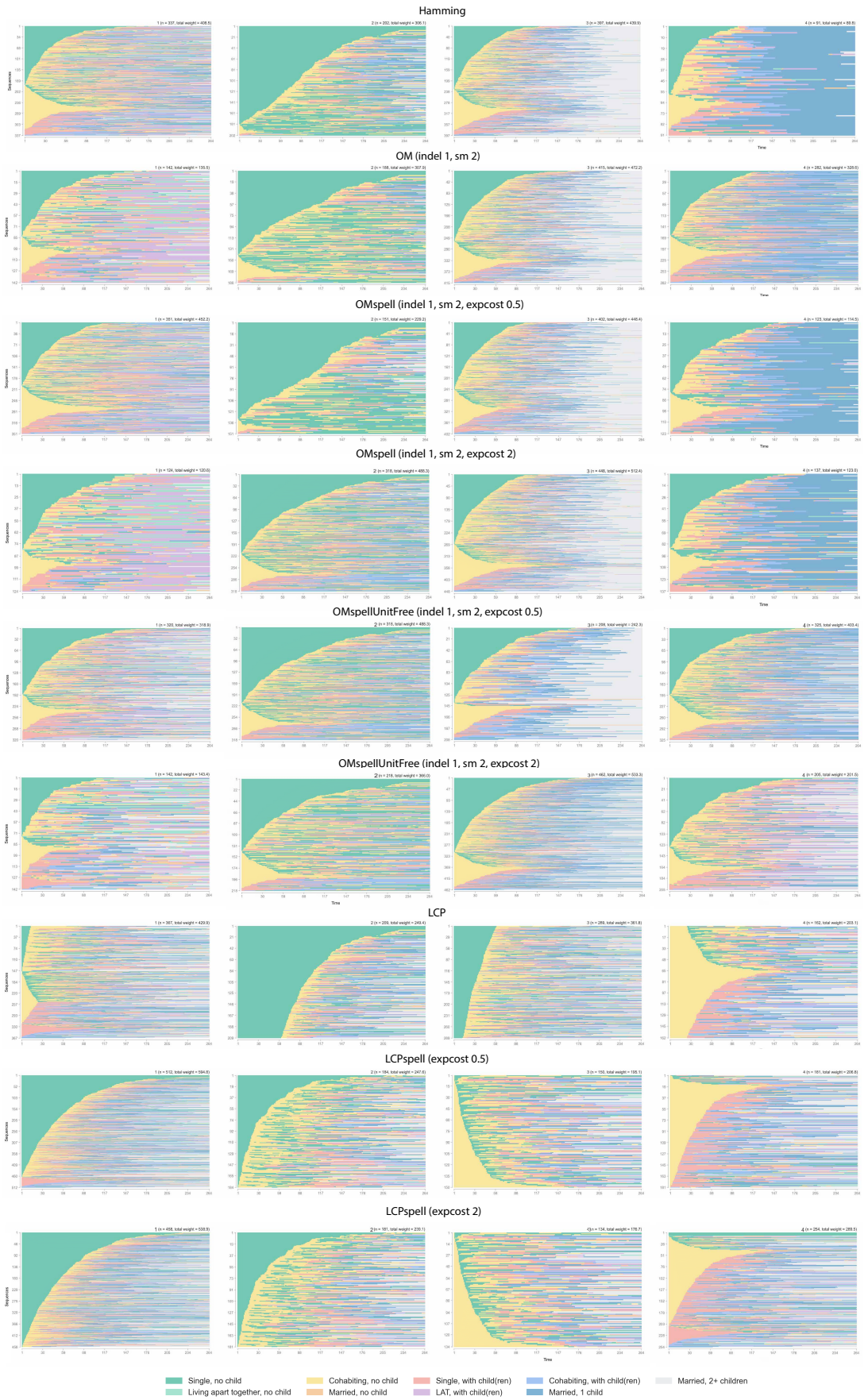


Figure 3: Clustering results for the pairfam dataset

This perspective also reframes the role of optimal matching in applied social research. OM remains useful, but it should be understood as one particular way of defining sequence similarity rather than as a default measure for all sequence comparison problems. As an edit-distance measure applied to time-expanded sequences, OM summarizes dissimilarity over the whole trajectory, but this broad summary can make it difficult to identify whether differences are driven mainly by timing, sequencing, or duration (with more limitations mentioned in Introduction). This makes OM-type measures useful when the research question concerns overall trajectory profiles, but less direct when the substantive interest lies in where trajectories begin to separate, where they later become similar, or whether apparent divergence is driven by exact transition timing rather than by differences in spell structure. Prefix- and suffix-based measures provide a more direct operationalization for these latter questions. The empirical demonstration illustrates this distinction: whole-trajectory measures tend to recover clusters defined by broad family-formation profiles, whereas LCP-type measures reveal how initially similar trajectories diverge from the start. Within the LCP family, position-wise LCP remains closely tied to exact calendar timing, while LCPspell shifts the comparison to shared spell-level prefixes or suffixes.

These distinctions also have direct implications for applied life-course research. Many substantive theories are not only about whether trajectories differ, but about the temporal form of that difference. If the research question concerns cumulative or aggregate pathway differences, whole-trajectory measures such as OM remain appropriate because they summarize dissimilarity across the full sequence. If the research question concerns exposure during a fixed age or calendar window, as in critical- or sensitive-period models, measures that preserve calendar-time alignment, such as Hamming distance or OM with a high indel cost, may be preferable because time warping can obscure the substantive meaning of the window. If the question concerns the order of transitions or the persistence of states, spell-based measures are more appropriate because they compare trajectories through spells and durations rather than through repeated time positions. Finally, if the question concerns where initially similar trajectories separate, or where initially different trajectories become similar, prefix- and suffix-based measures provide a more direct operationalization of divergence and convergence.

The broader implication is that method choice in sequence analysis should be guided by the substantive feature of interest rather than by default conventions, especially the default use of edit distances. Researchers should ask whether their question concerns overall trajectory profiles, exact timing, spell duration, spell order, or the location of divergence and convergence. Different dissimilarity measures answer different versions of this question. Edit distances, attribute-based measures (e.g., longest common subsequence, prefix- and suffix-based measures), and distribution-based measures each encode different assumptions about what makes trajectories similar; see Studer et al. (2016) for a broader review. Sequence analysis should therefore not be understood as the search for a universally optimal distance, but as the selection of a representation and dissimilarity measure that make the relevant form of similarity visible.

The study is not without limitations. The proposed reference-scaled duration terms in OM-

spellRS represent one possible way to make duration differences comparable across time units, but they reflect a specific modeling choice. By scaling durations relative to a fixed reference duration, such as the total observation window or the maximum possible spell duration implied by the study design, the approach emphasizes relative rather than absolute differences in duration. This scaling deliberately avoids using the maximum observed spell length in the dataset, because that choice would make the distance between the same two sequences depend on the other sequences included in the sample. While this improves comparability across datasets, it may be less appropriate in settings where absolute time differences carry substantive meaning, for example when policy thresholds or institutional time scales are of interest. Alternative normalization strategies, or approaches that retain absolute duration information, could therefore be explored in future work. The choice between spell-based and position-wise approaches should therefore be guided by the substantive interpretation of time in the data.

At the same time, prefix- and suffix-based measures are not designed to capture all forms of local sequence structure. They are most interpretable when the substantive question concerns shared beginnings or shared endings. They are less sensitive to differences that occur in the middle of otherwise similar trajectories, such as temporary interruptions, returns to a previous state, or repeated episodes between common initial and terminal states. In such cases, researchers may need complementary tools, such as subsequence-based, sliding-window, or local-pattern methods. The contribution of LCPspell is therefore not to replace whole-trajectory or subsequence-based approaches, but to fill a specific gap: the direct measurement of divergence and convergence in spell order.

In addition, prefix-based measures such as LCP and RLCP focus on shared initial or terminal segments of trajectories. While this provides a clear operationalization of divergence from a shared beginning and convergence toward a shared ending, it also implies reduced sensitivity to differences occurring in the middle of sequences or to more complex patterns involving multiple divergence and convergence phases. Extending prefix-based approaches to capture such patterns is beyond the scope of the current paper and remains an open methodological question.

Finally, although the simulation design incorporates mixed perturbations to reflect the interdependence of timing, sequencing, and duration, it remains a controlled setting. As such, the results should be interpreted as revealing systematic tendencies of different measures rather than as providing definitive performance rankings in fully empirical contexts. Taken together, the analysis shows that sequence comparison is not only a technical problem of choosing a distance formula, but also a representational choice about what aspects of trajectories should be made visible.

Acknowledgements

We would like to express our deep gratitude to Professor Gilbert Ritschard for his very insightful comments and suggestions on this paper. We are grateful to Professor Tim Liao and the PhD

students who participated in the 2025 Sequence Analysis Seminar at the Department of Sociology, University of Illinois Urbana-Champaign, for their valuable feedback on earlier versions of this paper. We also thank Professor Charles Rahal and participants in the 2026 Metrics and Models Seminar Series at the University of Oxford for their helpful comments and suggestions. We are grateful to Professor Ridhi Kashyap and Professor Man-Yee Kan for their feedback on a previous version of the manuscript. Finally, we thank Heyi Zhang and Yuting Liang for their support.

Disclosure statement

Not applicable.

Funding

Not applicable.

References

Abbott, Andrew and John Forrest (1986). "Optimal matching methods for historical sequences". In: *The Journal of Interdisciplinary History* 16.3, pp. 471–494.

Abbott, Andrew and Angela Tsay (2000). "Sequence analysis and optimal matching methods in sociology: Review and prospect". In: *Sociological methods & research* 29.1, pp. 3–33.

Aisenbrey, Silke and Anette Fasang (2017). "The interplay of work and family trajectories over the life course: Germany and the United States in comparison". In: *American Journal of Sociology* 122.5, pp. 1448–1484.

Barban, Nicola and Maria Sironi (2018). "Sequence analysis as a tool for family demography". In: *Analytical family demography*. Springer, pp. 101–123.

Biemann, Torsten and Deepak K Datta (2014). "Analyzing sequence data: Optimal matching in management research". In: *Organizational Research Methods* 17.1, pp. 51–76.

Brüderl, Josef et al. (2020). "The German family panel (pairfam)". In: *GESIS data archive, Cologne. ZA5678 Data file Version* 11.0, pp. 10–4232.

Cherlin, Andrew J (2004). "The deinstitutionalization of American marriage". In: *Journal of marriage and family* 66.4, pp. 848–861.

Elzinga, Cees H (2007). "Sequence analysis: Metric representations of categorical time series". In: *Manuscript, Dept of Social Science Research Methods, Vrije Universiteit, Amsterdam*.

— (2010). "Complexity of categorical time series". In: *Sociological methods & research* 38.3, pp. 463–481.

Elzinga, Cees H and Matthias Studer (2015). “Spell sequences, state proximities and distance metrics”. In: *Sociological Methods & Research* 44.1, pp. 3–47.

— (2019). “Normalization of distance and similarity in sequence analysis”. In: *Sociological Methods & Research* 48.4, pp. 877–904.

Fasang, Anette Eva (2014). “New perspectives on family formation: What can we learn from sequence analysis?” In: *Advances in sequence analysis: Theory, method, applications*, pp. 107–128.

Flothow, Amelie, Anna Novelli, and Leonie Sundmacher (2023). “Analytical methods for identifying sequences of utilization in health data: a scoping review”. In: *BMC Medical Research Methodology* 23.1, p. 212.

Gabadinho, Alexis et al. (2011). “Analyzing and visualizing state sequences in R with TraMineR”. In: *Journal of statistical software* 40, pp. 1–37.

Graham, Wendy et al. (2016). “Diversity and divergence: the dynamic burden of poor maternal health”. In: *The Lancet* 388.10056, pp. 2164–2175.

Grusky, David B and Robert M Hauser (1984). “Comparative social mobility revisited: models of convergence and divergence in 16 countries”. In: *American Sociological Review*, pp. 19–38.

Halpin, Brendan (2010). “Optimal matching analysis and life-course data: The importance of duration”. In: *Sociological Methods & Research* 38.3, pp. 365–388.

— (2014). “Three narratives of sequence analysis”. In: *Advances in sequence analysis: Theory, method, applications*. Springer, pp. 75–103.

Hembram, Sulekha and Sushil Kr Haldar (2020). “Is India experiencing health convergence? An empirical analysis”. In: *Economic Change and Restructuring* 53.4, pp. 591–618.

Hollister, Matissa (2009). “Is optimal matching suboptimal?” In: *Sociological methods & research* 38.2, pp. 235–264.

Kiess, Johannes et al. (2017). “Path dependency and convergence of three worlds of welfare policy during the Great Recession: UK, Germany and Sweden”. In: *Journal of International and Comparative Social Policy* 33.1, pp. 1–17.

Lesnard, Laurent (2010). “Setting cost in optimal matching to uncover contemporaneous socio-temporal patterns”. In: *Sociological methods & research* 38.3, pp. 389–419.

Levine, Joel H (2000). “But what have you done for us lately? Commentary on Abbott and Tsay”. In: *Sociological methods & research* 29.1, pp. 34–40.

Liang, Yuqi et al. (2025). *Documentation Website for Sequenzo and Social Sequence Analysis*. Zenodo. DOI: `10.5281/zenodo.17698151`. URL: `https://doi.org/10.5281/zenodo.17698151`.

Liao, Tim F. (2021). “Using sequence analysis to quantify how strongly life courses are linked”. In: *Sociological Science* 8, pp. 48–72.

Liao, Tim F. and Anette Eva Fasang (2021). “Comparing groups of life-course sequences using the Bayesian information criterion and the likelihood-ratio test”. In: *Sociological Methodology* 51.1, pp. 44–85.

Liao, Tim F. and Rebecca Yiqing Gan (2020). “Filipino and Indonesian migrant domestic workers in Hong Kong: Their life courses in migration”. In: *American Behavioral Scientist* 64.6, pp. 740–764.

Liao, Tim F. and Ting-Syuan Lin (2026). “Using Sequence Analysis in Social Science Research”. In: *Handbook on Data Modelling and Data Analysis*. Ed. by David Weakliem. Northampton, MA: Edward Elgar.

Liao, Tim F. et al. (2022). “Sequence analysis: Its past, present, and future”. In: *Social science research* 107, p. 102772.

Malo, Miguel A and Fernando Muñoz–Bullón (2003). “Employment status mobility from a life-cycle perspective: A sequence analysis of work-histories in the BHPS”. In: *Demographic Research* 9, pp. 119–162.

McVicar, Duncan and Michael Anyadike-Danes (2002). “Predicting successful and unsuccessful transitions from school to work by using sequence methods”. In: *Journal of the Royal Statistical Society Series A: Statistics in Society* 165.2, pp. 317–334.

Meyer, John W, John Boli-Bennett, and Christopher Chase-Dunn (1975). “Convergence and divergence in development”. In: *Annual review of sociology* 1, pp. 223–246.

Mortimer, Jeylan T and Michael J Shanahan (2007). *Handbook of the life course*. Springer Science & Business Media.

Pollock, Gary (2007). “Holistic trajectories: a study of combined employment, housing and family careers by using multiple-sequence analysis”. In: *Journal of the Royal Statistical Society Series A: Statistics in Society* 170.1, pp. 167–183.

Raab, Marcel (2017). “Childhood family structure and early family formation in East and West Germany”. In: *Journal of Marriage and Family* 79.1, pp. 110–130.

Raab, Marcel and Emanuela Struffolino (2022). *Sequence Analysis*. SAGE Publications, Incorporated.

Rehnberg, Johan, Stefan Fors, and Johan Fritzell (2019). “Divergence and convergence: How do income inequalities in mortality change over the life course?” In: *Gerontology* 65.3, pp. 313–322.

Ritschard, Gilbert et al. (2021). “Package ‘TraMineRextras’”. In: *Comprehensive R Archive Network (CRAN)*.

Rousset, Patrick, Jean-François Giret, and Yvette Grelet (2012). “Typologies de parcours et dynamique longitudinale”. In: *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique* 114.1, pp. 5–34.

Rubin, Ashley T (2023). “The promises and pitfalls of path dependence frameworks for analyzing penal change”. In: *Punishment & Society* 25.1, pp. 264–284.

Stevens, Fred (2010). “The convergence and divergence of modern health care systems”. In: *The new Blackwell companion to medical sociology*, pp. 434–454.

Studer, Matthias (2012). “Étude des inégalités de genre en début de carrière académique à l’aide de méthodes innovatrices d’analyse de données séquentielles”. PhD thesis. éditeur non identifié.

Studer, Matthias and Gilbert Ritschard (2016). “What matters in differences between life trajectories: A comparative review of sequence dissimilarity measures”. In: *Journal of the Royal Statistical Society Series A: Statistics in Society* 179.2, pp. 481–511.

Studer, Matthias et al. (2011). “Discrepancy analysis of state sequences”. In: *Sociological methods & research* 40.3, pp. 471–510.

Wu, Lawrence L (2000). “Some comments on “Sequence analysis and optimal matching methods in sociology: Review and prospect””. In: *Sociological methods & research* 29.1, pp. 41–64.

Yamaguchi, Kazuo (1991). *Event history analysis*. Sage Publications.

Appendix

Measuring Divergence and Convergence in Sequence Analysis: A Spell-Based Extension of Longest Common Prefixes

Yuqi Liang and Jan H. E. Meyerhoff-Liang

Department of Sociology, University of Oxford; Institute for New Economic Thinking, University of Oxford; Blavatnik School of Government, University of Oxford

A Index plots of sequence datasets that have different tendencies of variations in timing, order, duration, divergence and convergence

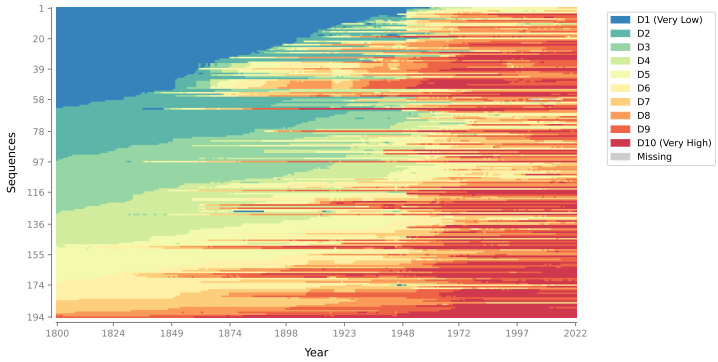
This section presents index plots for a set of empirical sequence datasets that exhibit different tendencies of variation along key sources of sequence variation, namely timing, order, duration, divergence, and convergence. An index plot is a standard visualization tool in sequence analysis that displays a collection of sequences as horizontal lines, where each line represents one unit (e.g. an individual or a country) and colors indicate categorical states over time. By aligning sequences along a common time axis, index plots provide a direct visual summary of how states evolve across units. All index plots are generated using the Python sequence analysis package `Sequenzo`. With the exception of Figure A.2, all datasets are included as built-in example datasets in the package to facilitate reproducibility and accessibility. In this section, we will only briefly describe the datasets and their sources, and for more details please refer to the documentation website of the `Sequenzo` package (Liang et al. 2025) at `https://sequenzo.yuqi-liang.tech/`.

Figure A.1 illustrates variation primarily in timing. Panel (a) shows country-level carbon emissions per capita trajectories, derived from Gapminder (2026). The original continuous values are transformed into categorical states using global deciles computed across all countries and years, allowing consistent comparisons over time. Panel (b) presents development aid trajectories based on political science aid datasets, capturing temporal variation in aid fragmentation patterns across countries, obtained from Gutting et al. (2017).

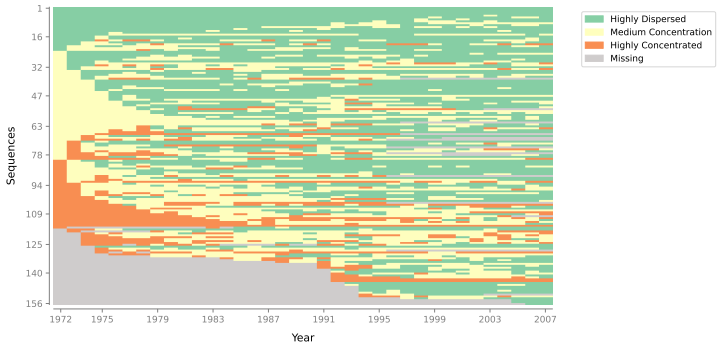
Figure A.2 presents regional trajectories of revealed comparative advantage (RCA), a widely used measure in economics of relative specialization in particular activities (French 2017); these data are not included in the `Sequenzo` package. Figure A.3 then illustrates divergence and convergence patterns. Panel (a) shows family formation trajectories of German individuals based on the pairfam dataset (Brüderl et al. 2020) collected by Raab et al. (2022), where sequences diverge substantially over time as individuals follow different life-course

pathways. Panel (b) shows activity trajectories of individuals in Northern Ireland based on the `mvad` dataset (McVicar et al. 2002) from the `TraMineR` package (Gabadinho et al. 2011), where sequences with heterogeneous starting points tend to converge toward similar later-life states.

The index plot from Liao et al. (2020) is not reproduced here due to data availability constraints, but the remaining plots together provide a comprehensive visual illustration of how different types of variation manifest in real-world sequence data.



(a) Carbon emissions per capita trajectories



(b) Development aid trajectories

Figure A.1: Index plots illustrating different tendencies of timing variation

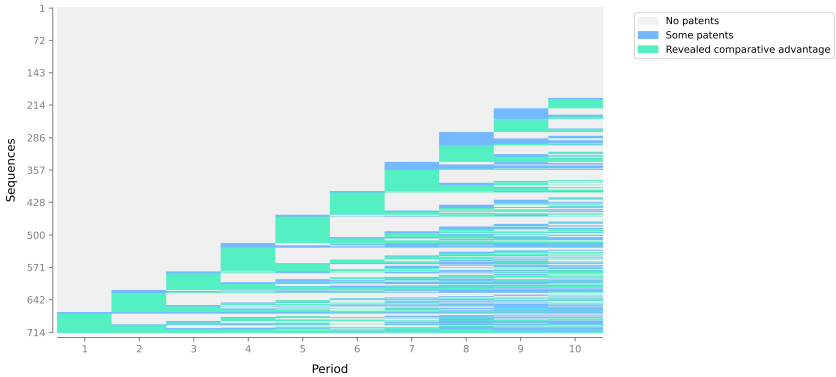
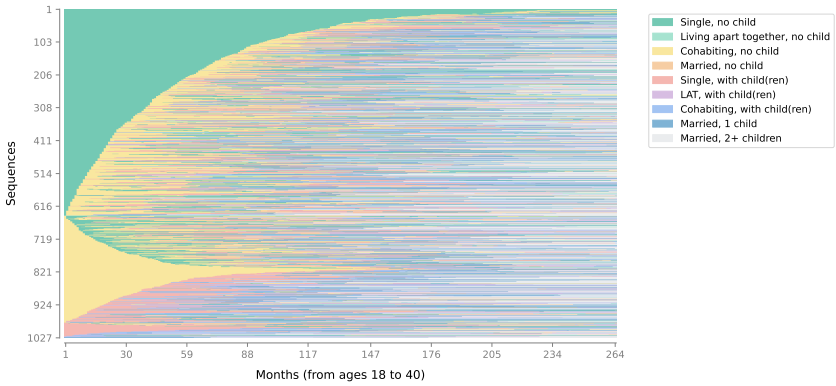
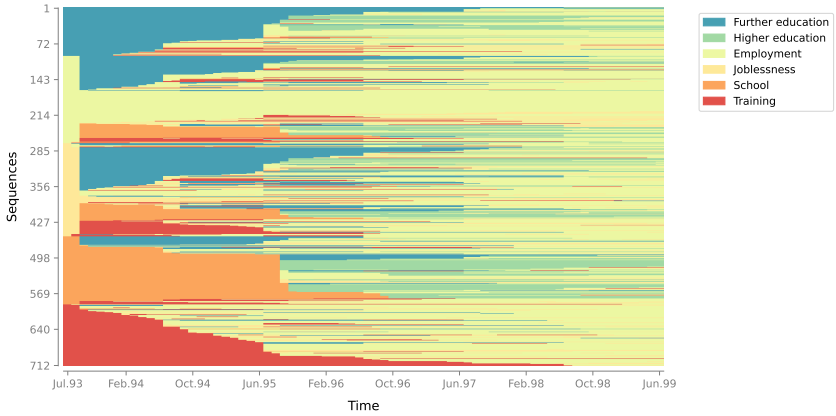


Figure A.2: Index plot of regional trajectories of revealed comparative advantage illustrating variations in duration



(a) Divergence: Family formation trajectories of German individuals



(b) Convergence: Transition from school to work trajectories of individuals in Northern Ireland

Figure A.3: Index plots illustrating divergence and convergence

B Illustration: OM, OMspell, and reference-scaled OMspell distances

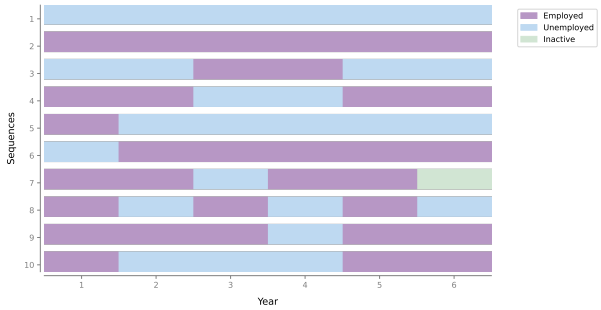


Figure B.1: Ten sequences for illustration

This appendix illustrates how position-wise OM, OMspell, and reference-scaled OMspell compute distances in two simple cases. The first case compares Sequences 9 and 10 in Figure B.1. These sequences share the same ordered spell states but differ in spell durations. This example illustrates same-state spell substitution. The second case compares Sequences 1 and 2. These sequences consist of one spell each, but the spell states differ. This example illustrates different-state spell substitution and clarifies why OMspell uses $d_i + d_j - 2$, rather than $d_i + d_j$, in the duration component of different-state substitutions.

Throughout this appendix, we use the same cost setting as in the main text:

$$c_{\text{indel}} = 1, \qquad \sigma(\mathtt{E}, \mathtt{U}) = 2,$$

and we set $\lambda = 1$ unless otherwise stated. For reference-scaled OMspell, we use $\tau = 6$, corresponding to the total observation window in this illustrative example.

Before presenting the calculations, it is useful to clarify how different values of `expcost` affect the distance. Following Studer et al. (2016), when `expcost` $= 0$, duration differences are not penalized: OMspell reduces to a pure spell-order comparison. In this case, two sequences with identical ordered spell states but different spell durations are treated as identical in terms of spell order. When `expcost` is positive, duration differences contribute to the distance. Larger values give greater weight to duration differences relative to state substitution and insertion/deletion costs. Choosing `expcost` is therefore a modelling decision: it determines whether and how much duration matters relative to spell-state alignment.

B.1 Same spell states with different durations: Sequences 9 and 10

We first consider Sequences 9 and 10. In time-expanded form, both sequences have length 6:

$$\text{Seq. 9: } \mathtt{EEEUEE}, \qquad \text{Seq. 10: } \mathtt{EUUUEE}.$$

In spell form, they are

$$\mathbf{s}_9 = \big((\mathtt{E},3),(\mathtt{U},1),(\mathtt{E},2)\big), \qquad \mathbf{s}_{10} = \big((\mathtt{E},1),(\mathtt{U},3),(\mathtt{E},2)\big).$$

They therefore share the same spell order,

$$\mathtt{E} \to \mathtt{U} \to \mathtt{E},$$

but differ in the durations of the first two spells.

B.1.1 Position-wise OM

Under standard OM, the sequences are compared in their time-expanded representation. An optimal alignment matches positions 1, 4, 5, and 6 exactly, and substitutes $\mathtt{E}$ with $\mathtt{U}$ at positions 2 and 3. Each such substitution costs

$$\sigma(\mathtt{E},\mathtt{U}) = 2.$$

Hence

$$d_{\mathrm{OM}} = 0 + 2 + 2 + 0 + 0 + 0 = 4.$$

This value arises from repeated position-level mismatches. Duration is not explicitly compared; it matters only because the same mismatch appears at consecutive time positions.

B.1.2 OMspell

Under OMspell, the same sequences are compared as spell sequences. With $\lambda = 1$, the optimal alignment matches all three spells:

$$(\mathtt{E},3) \leftrightarrow (\mathtt{E},1), \qquad (\mathtt{U},1) \leftrightarrow (\mathtt{U},3), \qquad (\mathtt{E},2) \leftrightarrow (\mathtt{E},2).$$

Because the spell states match in all three aligned pairs, only the same-state substitution cost applies:

$$c(o) = \lambda|d_i - d_j|.$$

The three aligned spell costs are therefore

$$\begin{aligned}
(\mathtt{E},3) \leftrightarrow (\mathtt{E},1): &\quad \lambda|3-1| = 2,\\
(\mathtt{U},1) \leftrightarrow (\mathtt{U},3): &\quad \lambda|1-3| = 2,\\
(\mathtt{E},2) \leftrightarrow (\mathtt{E},2): &\quad \lambda|2-2| = 0.
\end{aligned}$$

Thus,

$$d_{\mathrm{OMspell}} = 2 + 2 + 0 = 4.$$

Although this value equals the position-wise OM distance in this example, the interpretation is different. The OM distance arises from two repeated position-level mismatches, whereas the OMspell distance arises from duration differences within matched same-state spells.

If $\lambda = 0.5$, the same alignment gives

$$d_{\text{OMspell}} = 0.5 \times 2 + 0.5 \times 2 + 0 = 2.$$

This illustrates the role of λ: reducing `expcost` reduces the contribution of spell-duration differences while leaving the spell-state alignment unchanged.

B.1.3 Reference-scaled OMspell

Reference-scaled OMspell, implemented as `OMspellRS`, retains the same spell-level alignment logic but divides duration terms by a fixed reference duration τ. With $\tau = 6$ and $\lambda = 1$, the same-state substitution cost becomes

$$c(o) = \lambda \frac{|d_i - d_j|}{\tau}.$$

The same alignment therefore gives

$$\frac{|3-1|}{6} + \frac{|1-3|}{6} + \frac{|2-2|}{6} = \frac{2}{6} + \frac{2}{6} + 0 = \frac{2}{3}.$$

Thus,

$$d_{\text{OMspellRS}} = \frac{2}{3}.$$

If the same trajectories are measured in months rather than years, the spell sequences become

$$\mathbf{s}_9 = \big((\mathtt{E}, 36), (\mathtt{U}, 12), (\mathtt{E}, 24)\big), \qquad \mathbf{s}_{10} = \big((\mathtt{E}, 12), (\mathtt{U}, 36), (\mathtt{E}, 24)\big),$$

and the corresponding reference duration is $\tau = 72$. The reference-scaled duration component is then

$$\frac{|36-12|}{72} + \frac{|12-36|}{72} + \frac{|24-24|}{72} = \frac{24}{72} + \frac{24}{72} + 0 = \frac{2}{3}.$$

The raw reference-scaled distance is therefore unchanged. This illustrates the reference-scaled interpretation: duration is expressed as a proportion of a fixed reference duration, rather than as a cost per raw time unit.

If a normalized distance is required, normalization is a separate post-processing step. Using the OM-style structural upper bound discussed in Section **??**, and taking $n_s = m_s = 3$, $\max(c_{\text{indel}}) = 1$, and $\max(\sigma) = 2$, we obtain

$$d_{\text{struct}} = |3-3| \cdot 1 + \max(3,3) \cdot 2 = 6.$$

The normalized reference-scaled OMspell distance in this example is therefore

$$\frac{d_{\text{OMspellRS}}}{d_{\text{struct}}} = \frac{2/3}{6} = \frac{1}{9} \approx 0.111.$$

This normalization calculation is not part of the definition of `OMspellRS`; it is included only to show how the raw reference-scaled distance can be reported on a bounded scale when desired.

B.2 Different spell states: Sequences 1 and 2

We now consider Sequences 1 and 2, which illustrate different-state substitution. In time-expanded form, they are

$$\text{Seq. 1: } \texttt{U U U U U U}, \qquad \text{Seq. 2: } \texttt{E E E E E E}.$$

In spell form, they are simply

$$\mathbf{s}_1 = \big((\texttt{U}, 6)\big), \qquad \mathbf{s}_2 = \big((\texttt{E}, 6)\big).$$

Thus, the two sequences each contain one spell, but the spell states differ.

B.2.1 Position-wise OM

Under standard OM, the two sequences differ at every time position. One admissible optimal alignment substitutes `U` with `E` at all six positions. Since each substitution costs

$$\sigma(\texttt{U}, \texttt{E}) = 2,$$

the distance is

$$d_{\text{OM}} = 6 \times 2 = 12.$$

Equivalently, deleting all six positions from one sequence and inserting all six positions of the other sequence would also cost 12 when $c_{\text{indel}} = 1$. Thus, under OM$(1, 2)$, substitution and deletion-plus-insertion are tied in this example.

B.2.2 OMspell

Under OMspell, the comparison is between the two spells

$$(\texttt{U}, 6) \leftrightarrow (\texttt{E}, 6).$$

Because the states differ, the different-state substitution cost applies:

$$c(o) = \sigma(a, b) + \lambda(d_a + d_b - 2).$$

With $\sigma(\mathtt{U}, \mathtt{E}) = 2$, $\lambda = 1$, and $d_a = d_b = 6$, this gives

$$c(o) = 2 + (6 + 6 - 2) = 12.$$

Therefore,

$$d_{\text{OMspell}} = 12.$$

This calculation clarifies the role of the terms $d_a - 1$ and $d_b - 1$. Different-state substitution can be understood as first reducing each spell to its one-unit base, and then substituting the two base states:

$$(\mathtt{U}, 6) \to (\mathtt{U}, 1) \to (\mathtt{E}, 1) \to (\mathtt{E}, 6).$$

The two reductions contribute

$$\lambda(6 - 1) + \lambda(6 - 1) = 10,$$

and the state substitution contributes

$$\sigma(\mathtt{U}, \mathtt{E}) = 2.$$

The total is therefore

$$10 + 2 = 12.$$

This is why the duration term is $d_a + d_b - 2$, not $d_a + d_b$. A one-unit spell is already the base spell unit and should not incur an additional expansion cost.

B.2.3 Reference-scaled OMspell

Under reference-scaled OMspell, the different-state substitution cost becomes

$$c(o) = \sigma(a, b) + \lambda \frac{d_a + d_b - 2}{\tau}.$$

With $\tau = 6$, $\lambda = 1$, and $d_a = d_b = 6$, we obtain

$$c(o) = 2 + \frac{6 + 6 - 2}{6} = 2 + \frac{10}{6} = \frac{11}{3} \approx 3.667.$$

Thus,

$$d_{\text{OMspellRS}} = \frac{11}{3}.$$

The interpretation is different from OMspell. The state difference still contributes the substitution cost 2, but the duration-expansion component is now expressed relative to the six-period observation window. If the same trajectories were measured in months, the spell durations would be 72 and 72, and the reference duration would be $\tau = 72$. The duration component

would remain

$$\frac{72 + 72 - 2}{72}$$

only if the one-unit base is also defined at the monthly scale. This illustrates an important modelling point: reference-scaled OMspell is interpretable on a common scale when the time grid and the one-unit spell base are defined consistently. In applications, τ should therefore be chosen according to the observation design and reported explicitly.

This second example shows why different-state spell substitution is not merely a state substitution. When the two spells have different states, OMspell accounts for both the state change and the amount of duration that must be compressed or expanded around the one-unit spell base. Reference-scaled OMspell preserves this logic while expressing the duration component relative to a fixed reference duration.

C Illustration: LCP variants for Sequences 9–10 and Sequences 1–2

This appendix illustrates the four LCP variants in two settings. The first setting uses Sequences 9 and 10, which share the same ordered spell states but differ in duration. The second setting uses Sequences 1 and 2, which differ from the first state and from the first spell. These two cases show how LCP-type measures behave when trajectories differ only in spell duration within a shared spell order, and when they differ in spell state from the beginning.

C.1 Same spell states with different durations: Sequences 9 and 10

We first consider Sequences 9 and 10. In position-wise form,

$$\text{Seq. 9: } \mathtt{E\,E\,E\,U\,E\,E}, \qquad \text{Seq. 10: } \mathtt{E\,U\,U\,U\,E\,E}.$$

Both have length 6. In spell form,

$$\mathbf{s}_9 = \big((\mathtt{E}, 3), (\mathtt{U}, 1), (\mathtt{E}, 2)\big), \qquad \mathbf{s}_{10} = \big((\mathtt{E}, 1), (\mathtt{U}, 3), (\mathtt{E}, 2)\big).$$

They share the same spell order,

$$\mathtt{E} \to \mathtt{U} \to \mathtt{E},$$

but differ in the durations of the first two spells.

C.1.1 LCP

The position-wise LCP compares the two sequences from the beginning on the shared time axis. Position 1 matches:

$$\mathtt{E} = \mathtt{E}.$$

At position 2, the sequences differ:

$$\mathtt{E} \neq \mathtt{U}.$$

Thus the common prefix length is

$$L = 1.$$

By Eq. 15,

$$d_{\mathrm{LCP}} = n + m - 2L = 6 + 6 - 2 \times 1 = 10.$$

The large distance reflects the fact that the two sequences diverge at the second calendar-time position. Position-wise LCP does not recognize that the two trajectories nevertheless share the same spell order.

C.1.2 LCPmst

LCPmst computes the common prefix over the sequence of distinct successive states (DSS), not over repeated calendar-time positions. For Sequences 9 and 10, the DSS representations are identical:

$$\text{Seq. 9 DSS: } \mathtt{E} \to \mathtt{U} \to \mathtt{E}, \qquad \text{Seq. 10 DSS: } \mathtt{E} \to \mathtt{U} \to \mathtt{E}.$$

Hence the common DSS-prefix length is

$$L = 3.$$

The corresponding spell durations are

$$(3, 1, 2) \qquad \text{and} \qquad (1, 3, 2).$$

With $T_x = T_y = 6$, the minimal shared time over the common DSS prefix is

$$A_{\mathrm{mst}} = \min(3, 1) + \min(1, 3) + \min(2, 2) = 1 + 1 + 2 = 4.$$

By Eq. 20,

$$d_{\mathrm{LCPmst}} = T_x + T_y - 2A_{\mathrm{mst}} = 6 + 6 - 2 \times 4 = 4.$$

Thus, unlike position-wise LCP, LCPmst recognizes the shared ordered spell-state structure, but it still penalizes the duration differences through minimal shared time.

C.1.3 LCPprod

LCPprod uses the same common DSS prefix, so again

$$L = 3.$$

However, it weights matched prefix elements by the product of their durations:

$$A_{\text{prod}} = 3 \times 1 + 1 \times 3 + 2 \times 2 = 3 + 3 + 4 = 10.$$

By Eq. 22,

$$d_{\text{LCPprod}} = T_x + T_y - 2A_{\text{prod}} = 6 + 6 - 2 \times 10 = -8.$$

This negative value illustrates the interpretive difficulty of the product-based variant. Since products of durations are not bounded by the smaller duration, the accumulated product term can exceed the total duration scale. This is why LCPprod is less stable as a dissimilarity measure and is mainly of theoretical rather than practical interest.

C.1.4 LCPspell

LCPspell compares the ordered spell states. For Sequences 9 and 10, all three spell states match:

$$(\mathtt{E}, 3) \leftrightarrow (\mathtt{E}, 1), \qquad (\mathtt{U}, 1) \leftrightarrow (\mathtt{U}, 3), \qquad (\mathtt{E}, 2) \leftrightarrow (\mathtt{E}, 2).$$

Hence the common spell-prefix length is

$$L = 3,$$

and the spell-count component is

$$n_s + m_s - 2L = 3 + 3 - 2 \times 3 = 0.$$

When $\lambda = 0$, LCPspell ignores duration differences and yields

$$d_{\text{LCPspell}} = 0.$$

The two trajectories are then treated as identical in spell-prefix structure.

When $\lambda > 0$, duration differences among the matched prefix spells enter through the reference-scaled duration penalty. With $\tau = 6$ and $\lambda = 1$,

$$A_{\text{dur}} = \frac{|3-1|}{6} + \frac{|1-3|}{6} + \frac{|2-2|}{6} = \frac{2}{6} + \frac{2}{6} + 0 = \frac{2}{3}.$$

Therefore,

$$d_{\text{LCPspell}} = (n_s + m_s - 2L) + \lambda A_{\text{dur}} = 0 + 1 \times \frac{2}{3} = \frac{2}{3}.$$

If a bounded normalized LCPspell distance is desired, the upper bound must include both the spell-count component and the duration component. With $n_s = m_s = 3$, $\lambda = 1$, and $\min(n_s, m_s) = 3$,

$$d_{\text{max}}^{\text{LCPspell}} = (n_s + m_s) + \lambda \min(n_s, m_s) = 6 + 3 = 9.$$

The normalized LCPspell distance is therefore

$$\frac{d_{\text{LCPspell}}}{d_{\text{max}}^{\text{LCPspell}}} = \frac{2/3}{9} = \frac{2}{27} \approx 0.074.$$

C.2 Different spell states from the beginning: Sequences 1 and 2

We now consider Sequences 1 and 2. In position-wise form,

$$\text{Seq. 1: } \texttt{U U U U U U}, \qquad \text{Seq. 2: } \texttt{E E E E E E}.$$

In spell form,

$$\mathbf{s}_1 = \big((\texttt{U}, 6)\big), \qquad \mathbf{s}_2 = \big((\texttt{E}, 6)\big).$$

Unlike Sequences 9 and 10, these two sequences do not share the same first state or first spell state.

C.2.1 LCP

The position-wise LCP checks whether the sequences match from the first time position. At position 1,

$$\texttt{U} \neq \texttt{E}.$$

Thus,

$$L = 0.$$

Since both sequences have length 6,

$$d_{\text{LCP}} = n + m - 2L = 6 + 6 - 0 = 12.$$

This is the maximum possible raw LCP distance for two sequences of length 6: the trajectories diverge immediately on the calendar-time axis.

C.2.2 LCPmst

For LCPmst, the DSS representations are

$$\text{Seq. 1 DSS: } \texttt{U}, \qquad \text{Seq. 2 DSS: } \texttt{E}.$$

The first DSS element differs, so the common DSS-prefix length is

$$L = 0.$$

There is therefore no minimally shared time in the prefix:

$$A_{\text{mst}} = 0.$$

With $T_x = T_y = 6$,

$$d_{\text{LCPmst}} = T_x + T_y - 2A_{\text{mst}} = 6 + 6 - 0 = 12.$$

Thus, LCPmst also records immediate divergence, because the first spell states differ.

C.2.3 LCPprod

LCPprod also has no common DSS prefix in this case:

$$L = 0.$$

Hence the product-duration accumulation term is

$$A_{\text{prod}} = 0.$$

The distance is therefore

$$d_{\text{LCPprod}} = T_x + T_y - 2A_{\text{prod}} = 6 + 6 - 0 = 12.$$

Unlike the previous example, no negative value arises, because there are no matched DSS-prefix elements whose product durations could exceed the total duration scale.

C.2.4 LCPspell

For LCPspell, the spell sequences are

$$(\texttt{U}, 6) \qquad \text{and} \qquad (\texttt{E}, 6).$$

The first spell states differ, so the common spell-prefix length is

$$L = 0.$$

The spell-count component is therefore

$$n_s + m_s - 2L = 1 + 1 - 0 = 2.$$

Since there are no matched prefix spells, the duration component is also zero:

$$A_{\text{dur}} = 0.$$

Thus,

$$d_{\text{LCPspell}} = 2.$$

This value is smaller than the position-wise LCP distance of 12 because LCPspell operates on spells rather than time positions. The interpretation is not that the two trajectories are more similar in substantive terms, but that the unit of comparison has changed. Position-wise LCP counts six unmatched time positions in each sequence; LCPspell counts one unmatched spell in each sequence. This example therefore illustrates the core representational distinction: LCP measures immediate divergence on the calendar-time axis, whereas LCPspell measures immediate divergence in the ordered sequence of spells.

C.3 Summary of the two examples

The two examples highlight complementary cases. Sequences 9 and 10 have identical spell order but different spell durations. Position-wise LCP treats them as diverging early because they differ at the second time position, while LCPspell treats them as identical when $\lambda = 0$ and only partially different when duration is weighted. Sequences 1 and 2 differ from the first state and from the first spell. In this case, both position-wise and spell-wise prefix measures detect immediate divergence, but they express it on different scales: time positions for LCP, and spell units for LCPspell.

Together, these examples clarify why the proposed spell-based prefix approach is useful. It does not simply make distances smaller or larger. Rather, it changes the object being compared: from synchronized calendar-time positions to ordered spell structures, with duration differences incorporated only when explicitly weighted.

D Metric Conditions for LCP, RLCP, LCPspell, RLCPspell, and OMspellRS

A function $d : X \times X \to \mathbb{R}_{\geq 0}$ is a distance metric if it satisfies the following axioms for all $x, y, z \in X$:

D1. $d(x, x) = 0$ (identity);

D2. $d(x, y) > 0$ whenever $x \neq y$ (positivity);

D3. $d(x, y) = d(y, x)$ (symmetry);

D4. $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality).

This appendix clarifies the metric status of the distance and dissimilarity measures used in the paper. LCP and RLCP are metrics because they are induced by common-prefix and common-suffix similarities. OMspellRS is an edit-distance-type measure and inherits metric properties under the usual conditions on the underlying edit-cost structure. LCPspell and RLCPspell require more care. Their pure spell-order versions are metric, because they reduce to LCP or RLCP applied to the sequence of spell states. However, when the duration-penalty parameter is positive, the resulting duration-augmented versions are not claimed to be metrics unconditionally. They should instead be interpreted as reference-scaled prefix or suffix dissimilarities unless the relevant metric conditions are verified for the chosen parameter range.

D.1 LCP and RLCP

Let $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_m)$ denote position-wise state sequences. The *longest common prefix* length is

$$L(\mathbf{x}, \mathbf{y}) = \max\big\{k \geq 0 : x_i = y_i \text{ for all } i \in \{1, \ldots, k\}\big\},$$

with the convention that $L = \min(n, m)$ when the prefix matches up to the shorter sequence. The raw LCP distance is

$$d_{\mathrm{LCP}}(\mathbf{x}, \mathbf{y}) = n + m - 2L(\mathbf{x}, \mathbf{y}). \tag{D.1}$$

Define the similarity $s(\mathbf{x}, \mathbf{y}) = L(\mathbf{x}, \mathbf{y})$. Then

$$d_{\mathrm{LCP}}(\mathbf{x}, \mathbf{y}) = s(\mathbf{x}, \mathbf{x}) + s(\mathbf{y}, \mathbf{y}) - 2s(\mathbf{x}, \mathbf{y}).$$

Chen et al. (2009) show that whenever s satisfies the standard similarity axioms, including non-negativity, boundedness by self-similarity, symmetry, and the covering inequality, the induced function

$$d(x, y) = s(x, x) + s(y, y) - 2s(x, y)$$

defines a metric. The common-prefix length satisfies these requirements: it is nonnegative, bounded above by the length of each sequence, symmetric, and satisfies the covering condition for prefix structures (Elzinga et al. 2019). Hence D1–D4 hold for LCP.

RLCP applies the same construction from the end of the sequence. Let $L_{\mathrm{rev}}(\mathbf{x}, \mathbf{y})$ denote the longest common suffix length, that is, the number of consecutive matching positions from the last position backward. The raw RLCP distance is

$$d_{\mathrm{RLCP}}(\mathbf{x}, \mathbf{y}) = n + m - 2L_{\mathrm{rev}}(\mathbf{x}, \mathbf{y}). \tag{D.2}$$

Because this has the same structural form as (D.1), the same argument applies. The longest common suffix defines a similarity satisfying the same axioms, and therefore RLCP is also a metric.

D.2 LCPspell and RLCPspell

LCPspell operates on spell sequences rather than time-expanded state sequences. Let

$$\mathbf{s} = ((a_1, d_1), \ldots, (a_{n_s}, d_{n_s})), \qquad \mathbf{r} = ((b_1, e_1), \ldots, (b_{m_s}, e_{m_s})),$$

where a_k and b_ℓ are states, d_k and e_ℓ are spell durations, and n_s and m_s are the numbers of spells in the two sequences. The common spell-prefix length L is the number of consecutive spells from the beginning for which the states match:

$$a_k = b_k \qquad \text{for } k = 1, \ldots, L.$$

The raw LCPspell dissimilarity is

$$d_{\mathrm{LCPspell}}(\mathbf{s}, \mathbf{r}) = (n_s + m_s - 2L) + w \sum_{k=1}^{L} \frac{|d_k - e_k|}{\tau}, \tag{D.3}$$

where $w \geq 0$ is the duration-weight parameter and $\tau > 0$ is a fixed reference duration scale chosen before distance computation. In the main text, w corresponds to λ or `expcost`, and τ is set through `duration_ref`. The default reference scale is the total observation window T. It is not estimated from the maximum observed spell duration in the dataset, because that would make the distance between the same two sequences depend on which other sequences are included.

The first component,

$$n_s + m_s - 2L,$$

is exactly the LCP distance applied to the sequence of spell states, or equivalently to the DSS representation. It therefore satisfies the metric axioms by the same argument as position-wise

LCP. When $w = 0$, LCPspell reduces to this spell-count component and is a metric on spell-state sequences. The same applies to RLCPspell, which reduces to RLCP applied to the reversed sequence of spell states.

When $w > 0$, the second term adds a non-negative and symmetric reference-scaled duration penalty over the matched prefix or suffix. This term is substantively useful because it distinguishes trajectories that share the same ordered spell states but differ in the durations of those matched spells. However, because the duration penalty is evaluated only over the common prefix or suffix, the matched part itself can change across pairwise comparisons. The triangle inequality is therefore not guaranteed for all choices of w and τ.

A simple example illustrates the issue. Consider three one-spell sequences

$$x = ((A, 100)), \qquad y = ((A, 1)), \qquad z = ((B, 100)).$$

For x and y, the spell state matches, so the duration penalty enters:

$$d_{\mathrm{LCPspell}}(x, y) = w\frac{|100 - 1|}{\tau}.$$

If $\tau = 100$ and $w = 10$, then

$$d_{\mathrm{LCPspell}}(x, y) = 9.9.$$

By contrast, x and z do not share the first spell state, so $L = 0$ and the duration term does not enter:

$$d_{\mathrm{LCPspell}}(x, z) = 2.$$

Similarly,

$$d_{\mathrm{LCPspell}}(z, y) = 2.$$

Thus,

$$d_{\mathrm{LCPspell}}(x, y) = 9.9 > 2 + 2 = d_{\mathrm{LCPspell}}(x, z) + d_{\mathrm{LCPspell}}(z, y),$$

which violates the triangle inequality. This example shows why the duration-augmented LCPspell should not be described as a metric without additional restrictions.

Accordingly, the metric claim for LCPspell and RLCPspell is limited to the pure spell-order case $w = 0$. For $w > 0$, the measures should be interpreted as prefix- or suffix-based dissimilarities with an interpretable reference-scaled duration penalty. In the simulations, w is used as a controlled sensitivity parameter. Results based on unified reference normalization are therefore reported together with built-in normalization robustness checks, and the substantive conclusions are interpreted in terms of relative sensitivity patterns rather than as depending on strict metric claims for every duration-augmented parameter configuration.

The role of τ in (D.3) is to place duration differences on a reference-scaled and design-comparable scale. Replacing τ with the total observation window or another fixed design-implied duration bound changes the scale of the duration component but does not make the

distance dataset-dependent or pair-dependent. This is why the paper avoids using the maximum observed spell duration as the scaling factor.

D.3 OMspellRS

OMspellRS computes the minimum-cost edit path between two spell sequences using insertions, deletions, and substitutions. It follows the OMspell cost convention of Studer et al. (2016), in which expansion costs apply only to duration beyond the one-unit spell base. The reference-scaled version preserves the same spell-level edit logic but divides duration-expansion terms by a fixed reference duration τ.

Let the base substitution matrix be $\sigma(a, b)$ and let $c_{\text{indel}}(a)$ denote the insertion/deletion cost for state a. OMspellRS uses the following operation costs:

$$\text{Insertion/Deletion of } (a, d): \quad c_{\text{indel}}(a) + \lambda \frac{d-1}{\tau}, \tag{D.4}$$

$$\text{Substitution } (a, d_a) \leftrightarrow (b, d_b),\ a = b: \quad \lambda \frac{|d_a - d_b|}{\tau}, \tag{D.5}$$

$$\text{Substitution } (a, d_a) \leftrightarrow (b, d_b),\ a \neq b: \quad \sigma(a, b) + \lambda \frac{d_a + d_b - 2}{\tau}. \tag{D.6}$$

These formulas are the reference-scaled counterparts of the OMspell costs in Studer et al. (2016). The terms $d-1$ and d_a+d_b-2 reflect the fact that a one-unit spell carries no additional expansion cost beyond the base insertion/deletion or substitution cost. The parameter λ controls the weight of duration expansion, and τ fixes the duration scale before distance computation.

The reference scaling should be understood relative to a fixed observation design. Same-state duration differences are invariant to a change of time unit when both durations and τ are rescaled consistently. For operation costs involving the one-unit spell base, such as $d - 1$ and $d_a + d_b - 2$, comparability also requires that the one-unit base be interpreted consistently with the temporal grid. Thus, OMspellRS is best understood as a reference-scaled formulation that improves comparability across time units, rather than as a claim that all OMspell costs are invariant under arbitrary changes of temporal granularity.

The metric property of OMspellRS depends on the underlying edit-cost structure. Yujian et al. (2007) show that an optimal matching edit distance is a metric when the extended cost matrix, obtained by treating insertion/deletion as substitution with an empty state, defines a metric on the extended state space. In the present setting, this requires the base substitution and indel costs to satisfy the usual nonnegativity, identity, symmetry, and triangle-inequality conditions. Under this implementation and cost convention, OMspellRS is a minimum-cost alignment distance over spell-level objects.

The duration terms in (D.4)–(D.6) are nonnegative when $\lambda \geq 0$ and $\tau > 0$. Same-state substitution uses the scaled absolute difference $|d_a - d_b|/\tau$, which satisfies the triangle inequality in the duration coordinate. Different-state substitution and insertion/deletion use expansion

terms based on the additional duration beyond the one-unit spell base. These terms preserve the intended spell-level edit interpretation, provided that the one-unit base and the reference duration are defined on the same temporal grid.

This argument concerns the raw OMspellRS distance under the specified cost convention. Normalizing the resulting distance matrix is a separate post-processing step. In the main simulations, we use the reference-based normalization of Elzinga et al. (2019). Because this normalization is most directly justified for metric distances, duration-augmented prefix and suffix measures are interpreted cautiously as normalized dissimilarities, and the main conclusions are checked against method-specific built-in normalization in Appendix E.1.

E Robustness Checks of Simulation Results

E.1 Sensitivity Patterns Using Built-in Normalization

Figure E.1 presents the sensitivity patterns obtained when each method is evaluated using its built-in normalization, that is, the default normalization selected by the software implementation when `norm = auto`. This robustness check serves a different purpose from the unified reference-based normalization used in the main analysis. The main analysis applies the same reference-based normalization to all eligible methods in order to place distances on a common scale. By contrast, the built-in-normalization check asks whether the main sensitivity patterns remain visible when each method is reported under the normalization convention that is most commonly associated with that method in applied use.

Table E.1 summarizes the automatic normalization rules used in this robustness check. Let $d(x, y)$ denote the raw distance between sequences x and y. Let u_x and u_y denote the sequence-specific scale terms passed to the normalization routine, and let $d_{\max}(x, y)$ denote the method-specific upper bound used for max-distance normalization. For state-sequence measures, u_x and u_y usually correspond to time-expanded sequence lengths or their cost-weighted analogues. For spell-based measures, they are defined on the spell representation when the method itself operates on spells.

The corresponding normalization formulas are summarized in Table E.2. The table is intended to clarify the scale transformations used in the robustness check, rather than to define new distance measures.

For OMspell-type distances, it is useful to distinguish the automatic normalization used in this robustness check from the optional max-distance normalization. Under `norm = auto`, OMspell and OMspellRS use the Yujian–Bo normalization reported in Table E.1. If max-distance normalization is explicitly requested instead, a structural upper bound can be defined from the spell counts and the largest base insertion/deletion and substitution costs:

$$d_{\max}^{\mathrm{OMspell}} = |n_s - m_s| \max(c_{\mathrm{indel}}) + \max(n_s, m_s) \max_{a \neq b} \sigma(a, b), \tag{E.1}$$

Table E.1: Automatic built-in normalization rules used in the robustness check

Method family	**Auto normalization**	**Rationale**
OM, Hamming, DHD	`maxlength`	These are position-wise or time-expanded measures, so normalization is based on the larger sequence scale.
LCS, LCP, RLCP	`gmean`	These measures are based on common attributes or common prefixes/suffixes; geometric-mean normalization rescales the shared component relative to both sequence scales.
LCPmst, RLCPmst	`maxdist`	Minimal shared time gives a natural upper bound based on total durations.
LCPprod, RLCPprod	`none`	Product-duration weighting may produce negative raw distances; automatic normalization is therefore avoided.
LCPspell, RLCPspell	`maxdist`	The raw distance has an explicit spell-count component and an optional duration penalty, for which a method-specific upper bound is defined.
OMloc, OMslen, OMspell, OMspellRS, OMtspell, OMstran, TWED	`YujianBo`	These are edit-distance-type measures; Yujian–Bo normalization rescales an edit distance using the distance to the empty sequence.
NMS, NMSMST, SVRspell	`YujianBo`	These measures are normalized using the package's default distance-scaling convention for subsequence-based distances.
CHI2, EUCLID	method-specific `auto`	These distribution-based measures use their own internal normalization convention rather than the generic edit-distance normalizations.

Table E.2: Generic formulas for built-in normalization options

Normalization	**Formula**	**Interpretation**
`none`	$D(x,y) = d(x,y)$	No rescaling is applied.
`maxlength`	$D(x,y) = \dfrac{d(x,y)}{\max(u_x, u_y)}$	Rescales the raw distance by the larger sequence-specific scale.
`gmean`	$D(x,y) = 1 - \dfrac{d_{\max}(x,y) - d(x,y)}{2\sqrt{u_x u_y}}$	Rescales the unmatched part of the distance relative to the geometric mean of the two sequence scales.
`maxdist`	$D(x,y) = \dfrac{d(x,y)}{d_{\max}(x,y)}$	Divides the raw distance by a method-specific upper bound.
`YujianBo`	$D(x,y) = \dfrac{2d(x,y)}{u_x + u_y + d(x,y)}$	Normalizes an edit distance using the distances of each sequence to the empty sequence, following the Yujian–Bo normalization logic.

where n_s and m_s are the numbers of spells in the two sequences. Duration expansion costs are not added separately to this upper bound, because OMspell is a minimum-cost alignment: if an expansion path were more costly than an available substitution or insertion/deletion path, the optimal alignment would choose the cheaper alternative.

For LCPspell and RLCPspell, the automatic built-in normalization is different. Their duration penalty is part of the prefix or suffix distance definition rather than an optional edit path that can be avoided by the dynamic programming algorithm. Consequently, their `maxdist` upper bound includes both the spell-count component and the duration component:

$$d_{\max}^{\mathrm{LCPspell}} = (n_s + m_s) + \lambda \min(n_s, m_s), \tag{E.2}$$

provided that the fixed reference duration τ is chosen as an upper bound on possible spell duration, so that $|d_k - e_k|/\tau \leq 1$ for each matched spell.

Figure E.1 shows that the absolute values differ between the unified reference-based normalization in the main text and the built-in normalization rules reported here. This is expected, because built-in normalization rules are method-specific and do not place all measures on exactly the same scale. Nevertheless, the broad sensitivity patterns remain visible rather than identical. In particular, spell-order measures with `expcost`=0 remain among the leading methods in the sequencing strands. For the duration strand, the built-in normalization changes the relative ordering of some methods, especially OM, Hamming, and spell-based OM configurations, but the substantive conclusion remains the same: methods that either accumulate repeated time positions or explicitly weight duration differences are the ones that respond most strongly to duration contrasts. The divergence and convergence strands likewise retain the expected directional distinction between prefix- and suffix-oriented measures, although near-tied configurations may change order.

This robustness check therefore supports the main conclusion: the substantive findings do not depend on a single normalization convention. At the same time, the unified reference-based normalization remains preferable for the main comparison because it applies a common scaling logic across methods. Built-in normalization is useful for comparison with common software defaults, whereas unified normalization is better suited for interpreting cross-method sensitivity patterns in a controlled simulation design.

E.2 Raw Pseudo-R^2 Robustness Check on All Strands

Figure E.2 shows the Studer et al. (2016) raw pseudo-R^2 (ratio of between-group to total distance sums) across the full strand set used in this paper. The raw pseudo-R^2 results are broadly consistent with the chance-corrected results: the broad sensitivity patterns remain substantively similar, although the absolute scale and some relative rankings may differ.

E.3 Comparability-Extension Checks: Event-Based and Small-Perturbation Strands

To assess whether the main conclusions depend on the particular spell/state sequence generator used in the main simulations, we conduct two additional sets of checks inspired by the simulation designs of Studer et al. (2016). The first set uses event-based sequence generation. Instead of directly generating state sequences from DSS patterns and spell durations, we first generate the occurrence times of underlying events and then derive the resulting state sequences from these events. We consider three event-based strands: event order, event timing, and inter-event duration. These correspond to whether groups differ in the order in which events occur, the timing of a focal event, or the elapsed time between two events.

The second set examines small perturbations. These checks ask whether a distance measure reacts strongly when two otherwise similar sequences differ only by a very small local change. We use two small-perturbation strands: a single-token perturbation, where one state observation is changed, and a boundary-shift perturbation, where the change occurs at or near the boundary between two adjacent spells. These designs are deliberately subtle: the two groups are expected to be only weakly separated because the systematic difference between them is small.

The same set of distance measures and the same chance-corrected pseudo-R^2 evaluation are used as in the main analysis. Figure E.3 summarizes the results. Three patterns are most relevant. First, in the event-order strand, OMspell-family configurations show the strongest sensitivity, while LCPspell and RLCPspell variants also remain among the better-performing methods. Second, in the event-timing and inter-event-duration strands, many duration-aware or positive-`expcost` configurations form a high-performing group with nearly identical pseudo-R^2 values. LCPspell and RLCPspell variants belong to this group, but they should not be interpreted as uniquely superior in these two checks. The result instead shows that the spell-based duration component behaves consistently under the event-based generator. Third, both small-perturbation strands produce pseudo-R^2 values close to zero overall. This is expected because the group difference is intentionally limited to a small local change.

Some chance-corrected pseudo-R^2 values are negative. This does not indicate a computational problem. It means that, for those methods, between-group distances are not larger than would be expected under random group labels. Such values are especially plausible in the small-perturbation checks, where the systematic group difference is intentionally weak.

As a further check, Figure E.4 reports the same extension checks using the raw pseudo-R^2 measure used by Studer et al. (2016), defined as between-group discrepancy divided by total discrepancy. The broad sensitivity patterns remain substantively similar, although the absolute scale and some relative rankings may differ.

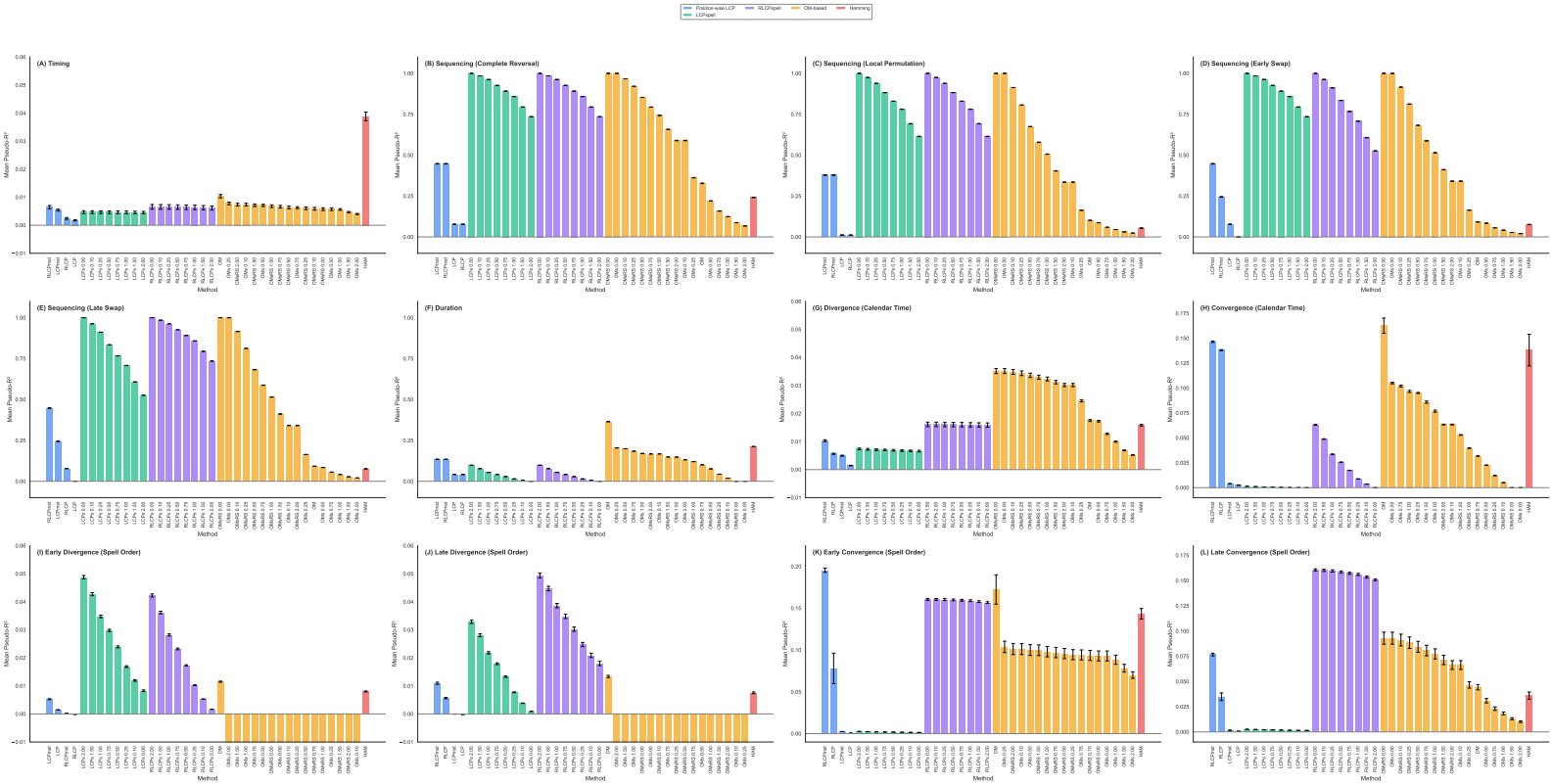


Figure E.1: Sensitivity profiles across all methods and experimental strands using built-in normalization (each method's default normalization)

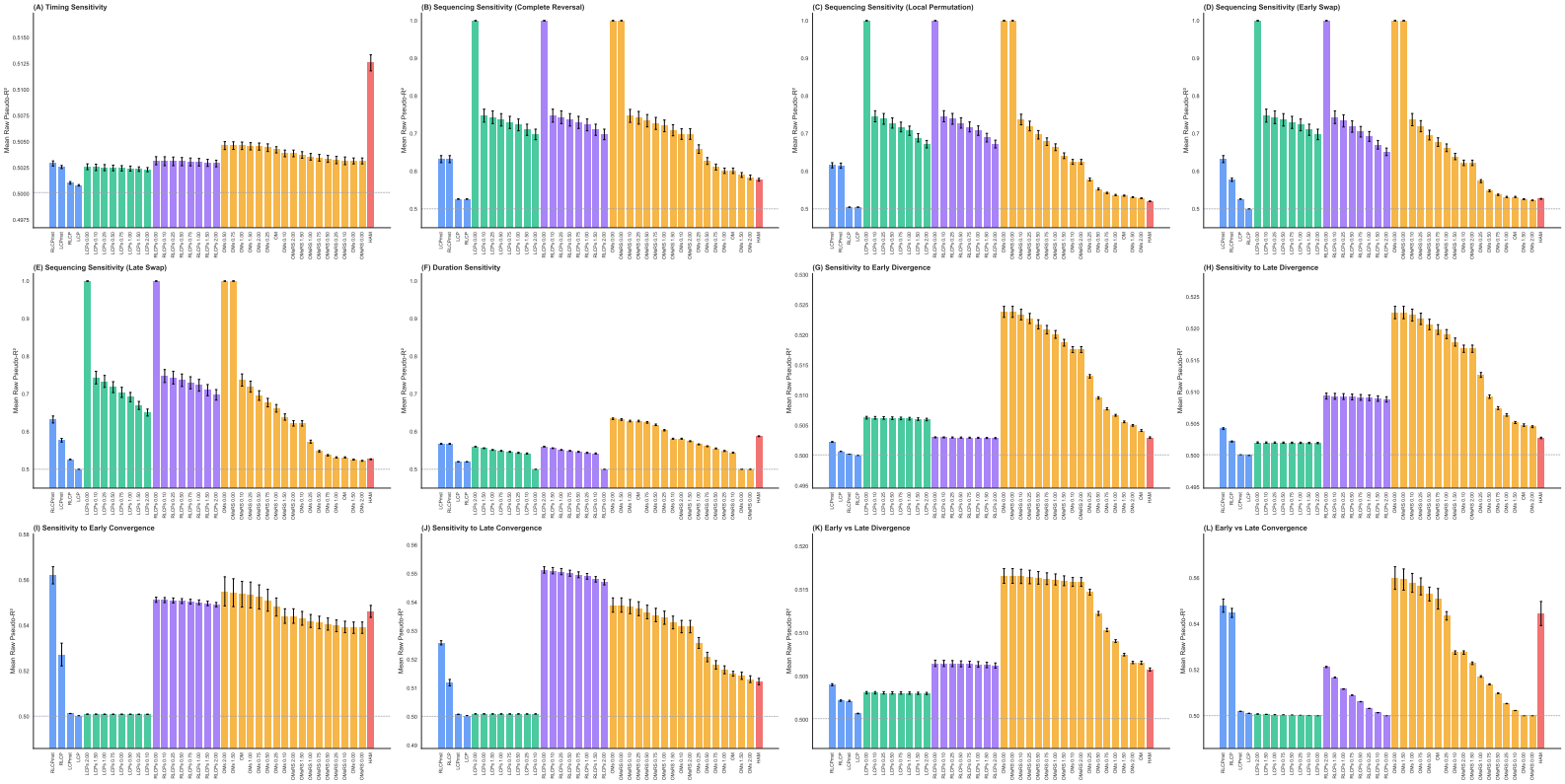


Figure E.2: All-strand sensitivity patterns using raw (uncorrected) pseudo-R^2 (Studer et al. 2016).

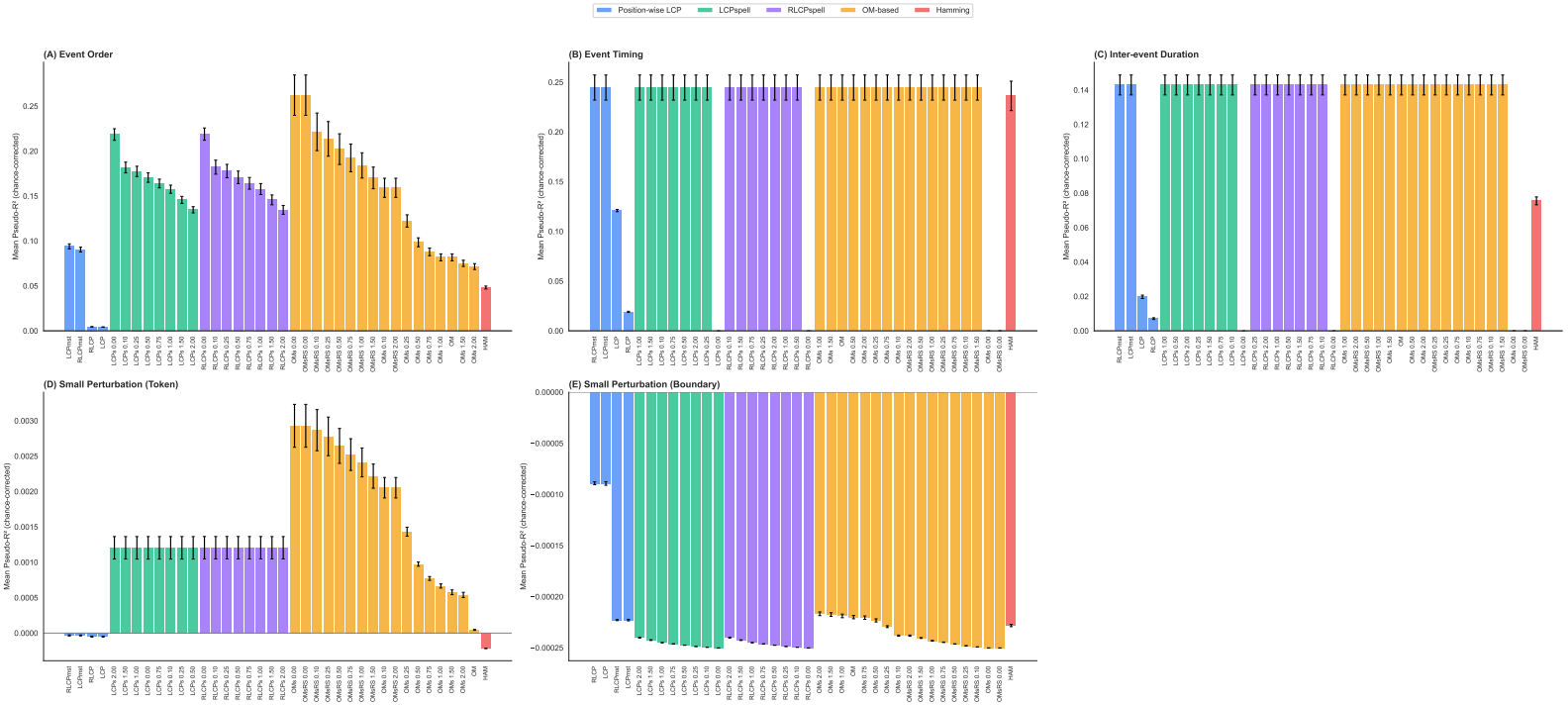


Figure E.3: Sensitivity profiles for comparability-extension checks (event-based and small-perturbation strands).

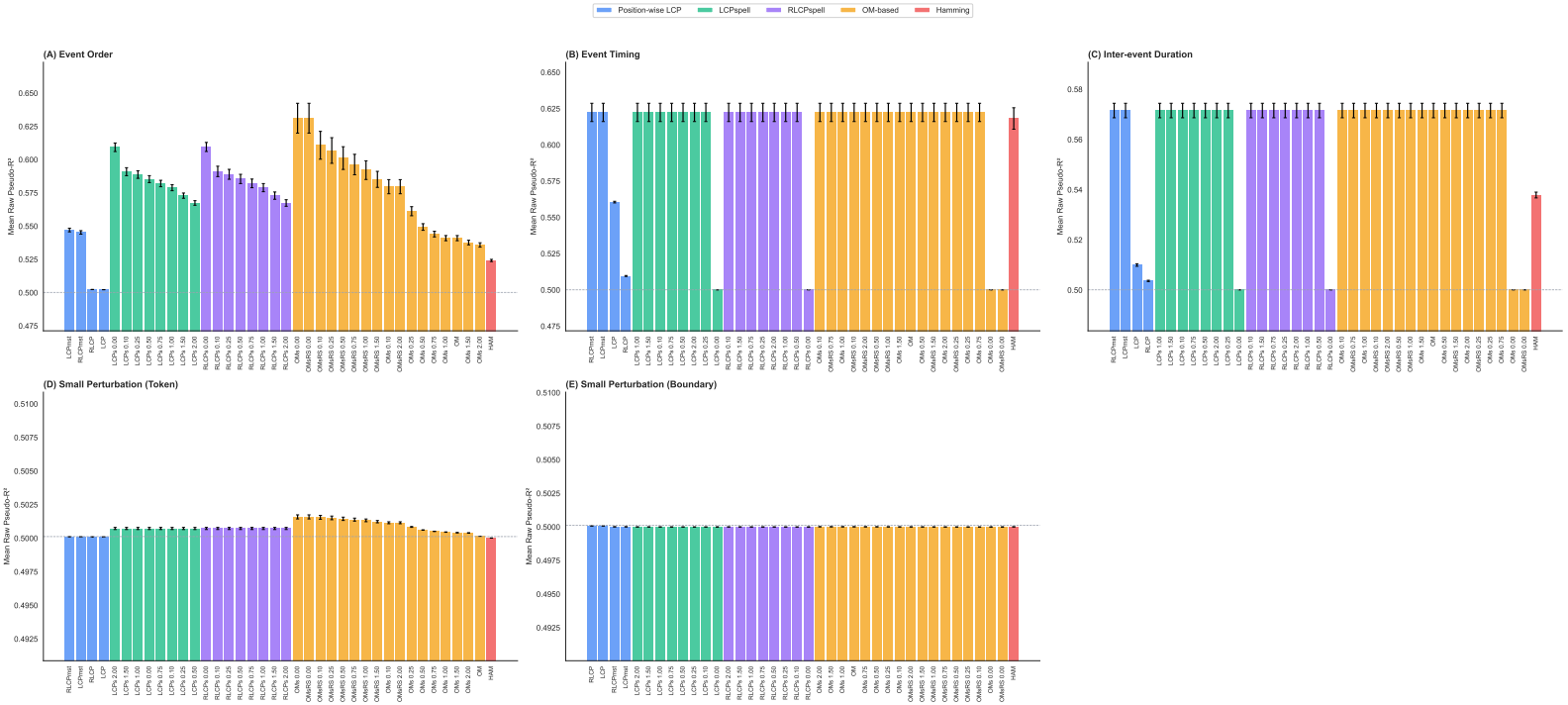


Figure E.4: Sensitivity profiles for comparability-extension checks using raw (uncorrected) pseudo-R^2.

References

Brüderl, Josef et al. (2020). "The German family panel (pairfam)". In: *GESIS data archive, Cologne. ZA5678 Data file Version* 11.0, pp. 10–4232.

Chen, Shihyen, Bin Ma, and Kaizhong Zhang (2009). "On the similarity metric and the distance metric". In: *Theoretical Computer Science* 410.24, pp. 2365–2376.

Elzinga, Cees H and Matthias Studer (2019). "Normalization of distance and similarity in sequence analysis". In: *Sociological Methods & Research* 48.4, pp. 877–904.

French, Scott (2017). "Revealed comparative advantage: What is it good for?" In: *Journal of International Economics* 106, pp. 83–103.

Gabadinho, Alexis et al. (2011). "Analyzing and visualizing state sequences in R with TraMineR". In: *Journal of statistical software* 40, pp. 1–37.

Gapminder (2026). *Gapminder Data Documentation and Sources*. URL: https://www.gapminder.org/data/documentation/ (visited on 04/27/2026).

Gutting, Raynee and Martin C Steinwand (2017). "Donor fragmentation, aid shocks, and violent political conflict". In: *Journal of Conflict Resolution* 61.3, pp. 643–670.

Liang, Yuqi et al. (2025). *Documentation Website for Sequenzo and Social Sequence Analysis*. Zenodo. DOI: 10.5281/zenodo.17698151. URL: https://doi.org/10.5281/zenodo.17698151.

Liao, Tim F. and Rebecca Yiqing Gan (2020). "Filipino and Indonesian migrant domestic workers in Hong Kong: Their life courses in migration". In: *American Behavioral Scientist* 64.6, pp. 740–764.

McVicar, Duncan and Michael Anyadike-Danes (2002). "Predicting successful and unsuccessful transitions from school to work by using sequence methods". In: *Journal of the Royal Statistical Society Series A: Statistics in Society* 165.2, pp. 317–334.

Raab, Marcel and Emanuela Struffolino (2022). *Sequence Analysis*. SAGE Publications, Incorporated.

Studer, Matthias and Gilbert Ritschard (2016). "What matters in differences between life trajectories: A comparative review of sequence dissimilarity measures". In: *Journal of the Royal Statistical Society Series A: Statistics in Society* 179.2, pp. 481–511.

Yujian, Li and Liu Bo (2007). "A normalized Levenshtein distance metric". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6, pp. 1091–1095.