TISMIR

RESEARCH ARTICLE

ubiquity press

Time to Align! Modelling Musical Timelines for Music Information Retrieval and Digital Musicology

JOHANNES HENTSCHEL
AXEL BERNDT
CARLOS CANCINO-CHACÓN
SIMON DIXON
ANNE FOO
MARK GOTHAM
PATRICIA HU
MAIK KÖSTER
FELIPE D. MARTINS
DAVIDE A. MAURO
MEINARD MÜLLER
MARKUS NEUWIRTH
ALEXANDER PACHA
KEVIN R. PAGE
SILVAN PETER
EGOR POLYAKOV
LAURENT PUGIN
DAVID M. WEIGL
CHRISTOF WEIß
GERHARD WIDMER

*Author affiliations can be found in the back matter of this article

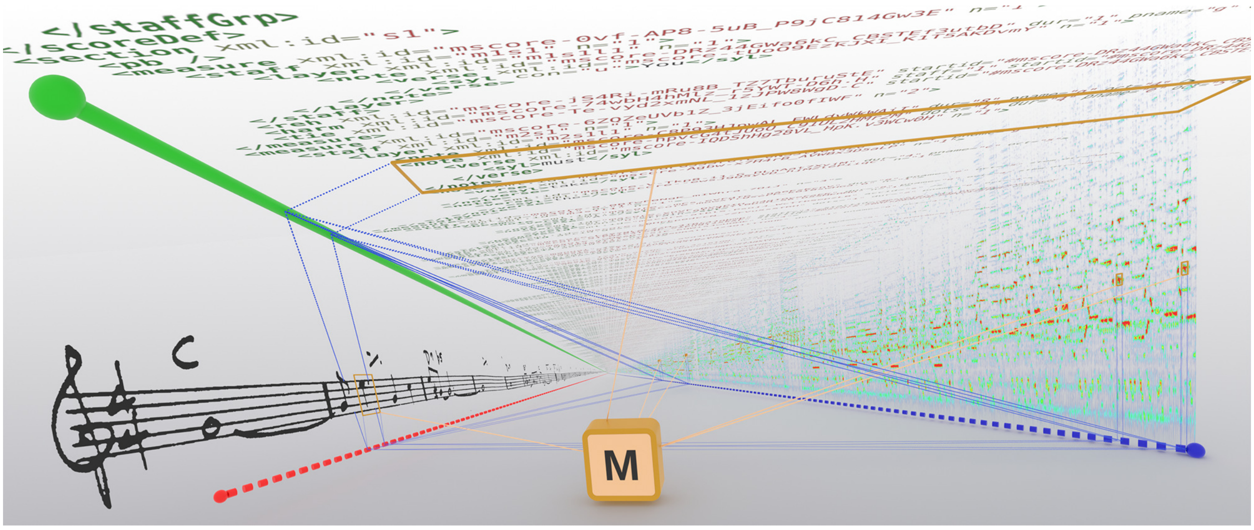
CORRESPONDING AUTHOR:
Johannes Hentschel
Institute for Theory and History, Anton Bruckner University, Linz, Austria
johannes.hentschel@bruckneruni.at

ABSTRACT

Aligning musical data – matching corresponding events across different representations of the same music – is central to Music Information Retrieval and Digital Musicology. Datasets span graphical representations (images, scans), logical encodings (scores, MIDI) and physical recordings (audio, video, sensor data), each imposing its own coordinate system for musical time. Integrating them requires costly alignment, but results are typically stored in *ad hoc* formats that are poorly documented and difficult to share and reuse. To address this, we present *TimeToAlign!*, a cross-domain model for musical timelines and their alignments that organises time-related data into continuous and discrete variants across three temporal domains – graphical, logical and physical – connected by typed conversion maps and explicit match claims. In addition to describing the model, we demonstrate it through five concrete use cases of increasing complexity: (1) aligning a historical piano roll with a digital score, (2) modelling flow control in a school song arrangement, (3) transferring annotations between graphical analyses, (4) propagating harmonic labels across hundreds of multimodal timelines of a Beethoven string quartet and (5) encoding the conceptual and temporal relationships in the genesis of a rock song. The model is implemented as timetoalign, an open-source Python library with typed timelines, loaders for common formats (such as audio formats, MusicXML or CSV) and conversion maps. Runnable tutorial and how-to notebooks, a documentation homepage and a collection of real-world data from diverse repertoires accompany the library.

KEYWORDS:
musical timelines, alignment, synchronisation, temporal domains, exchange format, Python library

TO CITE THIS ARTICLE:
Hentschel, J., Berndt, A., Cancino-Chacón, C., Dixon, S., Foo, A., Gotham, M., Hu, P., Köster, M., Martins, F.D., Mauro, D.A., Müller, M., Neuwirth, M., Pacha, A., Page, K.R., Peter, S., Polyakov, E., Pugin, L., Weigl, D.M., Weiß, C., & Widmer, G. (2026). Time to Align! Modelling Musical Timelines for Music Information Retrieval and Digital Musicology. *Transactions of the International Society for Music Information Retrieval*, 9(1), 384–404.
DOI: https://doi.org/10.5334/tismir.296



1 INTRODUCTION

Music research relies on heterogeneous time-related data: scores, recordings, annotations, images, symbolic encodings, sensor data and more. Music Information Retrieval (MIR) and Digital Musicology (DM) both depend on such data, although they approach it from different angles – MIR emphasising technical extraction and processing, and DM focusing on hermeneutic, historical and analytical enquiry. Much of these data can be conceptualised as forming a *musical timeline*: a structured representation of how musical events unfold over time. Timelines manifest across a wide range of data types, from sheet music and audio recordings to symbolic representations (e.g. MIDI), performance data (e.g. audio recordings) and graphical analyses: the landscape of music-related research data is deeply fragmented. Datasets differ in repertoire, completeness, analytical detail, encoding formats and quality control. Combining them, i.e. matching corresponding events across different representations of the same music, requires *alignment*, a process that is both central to these disciplines and expensive to carry out. Unfortunately, the resulting alignment data are rarely practical for reuse: they tend to be stored in *ad hoc* formats that are poorly documented and tightly coupled to the specific datasets they connect.

These are not abstract concerns. In practice, undocumented or incompatible time encoding leads to concrete failures at every stage of the research workflow. Perhaps the most pervasive problem is that datasets rarely specify how musical time was encoded. Measure numbers, in particular, are a frequent source of ambiguity: different score editions count first and second endings, anacrusis measures or inserted *ad lib* bars differently such that 'measure 32' in one dataset may refer to a different passage in another. Beat positions are similarly treacherous: the unit of a beat (e.g. an eighth or a quarter), the convention for numbering beats within a measure and the handling of partial beats at phrase boundaries can all vary between datasets and are often undocumented. These discrepancies mean that annotations painstakingly created for one score representation cannot be transferred to another without manual reconciliation – a potentially costly endeavour requiring costly expert labour. This challenge even exists within the physical domain: timestamps from multiple digital copies of the same CD recording may diverge due to the different handling of silence at track boundaries, leading to a mismatch of extracted features. A further downstream consequence is that applications relying on measure numbers or beat positions for synchronisation (for instance, score-display tools that couple a cursor to audio playback) may exhibit erratic behaviour when the same measure number occurs multiple times in a score. Each of these problems stems from the same root cause: the absence of an explicit, shareable model for musical timelines and their alignments.

We present *TimeToAlign!*, a generalised cross-domain model for musical timelines and their alignments. Building on previous work (e.g. Selfridge-Field, 1997), we organise time-related data into three temporal domains – graphical, logical and physical – each in continuous and discrete variants (see Figure 1; for a detailed explanation, see Section 2.1). Timelines are connected by typed *ConversionMaps* (within a domain or across domains) and explicit *MatchClaims* (claims of equivalence between events on different timelines, with metadata recording the [human or machine] agent, method and certainty of the alignment). This article makes two contributions:

1. A **conceptual model** for cross-domain musical timelines and their alignments, demonstrated through five real-world use cases of increasing complexity, drawn from historical piano rolls, score-based flow control, acousmatic music, multimodal recordings and rock music genesis research.

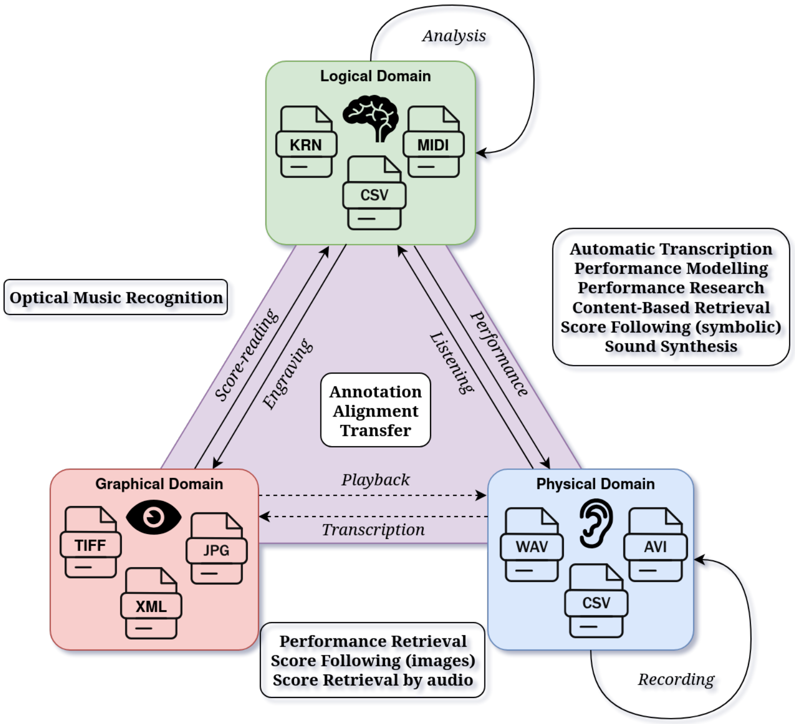


Figure 1 Three temporal domains, each associated with a human organ (brain, eye and ear) and typical file formats. Arrows with italic captions represent activities of transfer from one domain to another. White rectangles name select MIR and DM applications situated between two domains or (in the middle) potentially involving all three.

2. A **Python library** (`timetoalign`) that implements this model with typed timelines, loaders for common formats (such as audio formats, MusicXML or CSV) and conversion maps.

The library is accompanied by tutorial notebooks and a documentation homepage with rendered examples and specimen data from diverse repertoires.[1]

The remainder of this article is organised as follows. Section 2 situates our work within the landscape of time-related research data, reviewing the three temporal domains, common synchronisation tasks and the limitations of existing models and formats. Section 3 introduces the *TimeToAlign!* model through five concrete examples that progressively demonstrate its key concepts. Section 4 describes the `timetoalign` library, its tutorial notebooks and the documentation site. Section 5 discusses what the model and implementation achieve, their limitations and future directions. The full conceptual specification, tutorial walkthroughs and source code are available at the accompanying documentation site.[1]

2 TIME-RELATED RESEARCH DATA IN MIR AND DM

2.1 TEMPORAL DOMAINS IN MUSICAL DATA

We adopt a conceptualisation of musical time through three domains – physical, logical and graphical – each providing a distinct perspective on how time is represented in music-related data (Figure 1). The **physical** domain concerns data on physical, real-time manifestations of sound or sound production: audio recordings, synthesised renditions, sensor data and derived features such as spectrograms. Time in this domain is measured in seconds, samples or frames. The **logical** domain concerns symbolic, conceptual representations of musical structure: note events in MIDI or MusicXML, measure grids, beat positions and analytical annotations. Time is measured in beats, quarter notes, MIDI ticks or other abstract units. The **graphical** domain concerns spatial representations that *encode* timelines from the first two domains visually: sheet music scans, graphical analyses, piano-roll images and bounding boxes produced by an Optical Music Recognition (OMR) system. Images are not themselves temporal objects, but they map musical time – whether physical or logical – to spatial coordinates (pixels, millimetres), which is why we treat them as a temporal domain in their own right. Score-following videos and animated graphical analyses (e.g. the spectromorphological analysis from Thoresen (2010) available on YouTube[2]) straddle the physical and graphical domains: they are physical renderings of graphical timelines, and our model accommodates this by linking both.

Similar conceptual triads appear in prior work, often using different terminology that emphasises a particular modality, production technique, representation or operationalisation of the same three domains (Table 1). Selfridge-Field's 'sound or phonological context, [. . .] graphical context of notation, rational context'

Domain	Modality	Production techniques	Representation	Operationalisations
Graphical	Seeing (static, architectural), sense of touch, spatial thinking	Printing, sketching, drawing, encoding visually	Visual	Raster images (coordinate system), shapes
Logical	Conceptualising, understanding, reading	Encoding symbolically, representing, modelling	Symbolic	Musical objects, notational objects
Physical	Hearing, seeing (dynamic in time, directional)	Performing, editing/ producing, animating	Audio-visual, sound	Audio features, sound studies, acoustic modelling

Table 1 Disentangled terminology concerning the three domains.

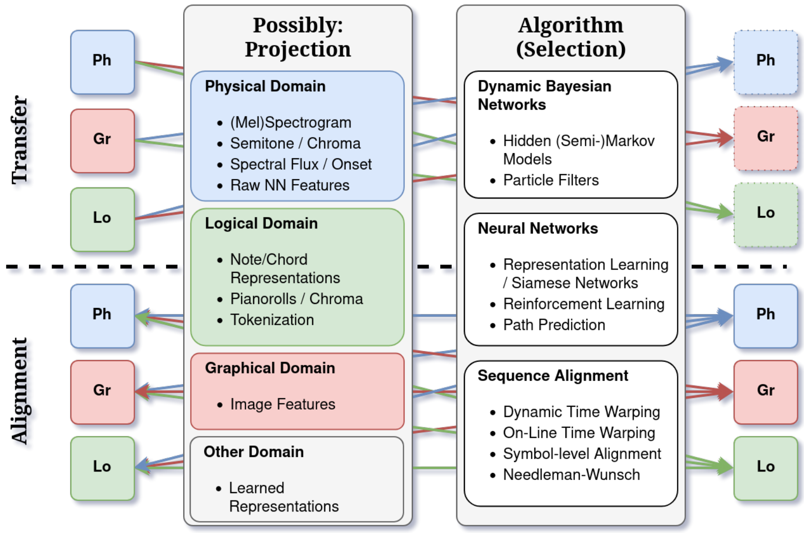


Figure 2 Schema showing transfer and alignment across domains. Transfer converts a timeline from one domain to another (left-to-right arrows); alignment identifies temporal equivalence between two timelines from the same or different domains (two-headed arrows). Both may use an intermediary representation ('projection') and one or several algorithms (rectangles).

(Selfridge-Field, 1997, pp. 7–8) corresponds to our physical, graphical and logical domains. Gotham et al. (2023a) propose an equivalent triad of 'performance', 'print' and 'logic', focused on production techniques. Wiggins et al. (2010) and Liem et al. (2011), following Babbitt (1965), distinguish an acoustic, an auditory/perceptual and a graphemic domain. Our Figure 1 adapts Figure 1 of Liem et al. (2011) while adjusting the transformations between domains. We depart from the Babbitt-derived triad in one respect: what they call the auditory/perceptual domain is, from a temporal standpoint, a subdomain of the physical; the cognitive processes they associate with it (score reading, listening, transcription) are relevant for engaging with *all three* domains rather than constituting a domain of their own.

A consequence of our triad is that music notation is *both* graphical and logical: a score is, in effect, an alignment between a graphical layout (where notes are placed on the page) and a logical structure (which events they represent and in what order). This does not entail that the transfer between the graphical and physical domains must always pass through a logical intermediary, as recent end-to-end approaches demonstrate (e.g. Jung et al., 2025). By clearly differentiating graphical from logical aspects, the three domains can add conceptual clarity to notation-related applications such as score following, score annotation and score retrieval, each of which may focus on one aspect or the other, if not both (cf. Gotham et al., 2025).

2.2 MUSIC SYNCHRONISATION TASKS

Navigating between corresponding timelines is central to both MIR and DM. As schematised in Figure 2, this takes two forms. *Domain transfer* converts a timeline from one domain to another (e.g. transcription converts audio to a symbolic score), producing two aligned timelines. *Alignment* identifies temporal equivalences between two existing timelines from the same or different domains. We subsume both under 'music synchronisation' (Arifi et al., 2003; Dannenberg and Raphael, 2006; Müller, 2021).

Music synchronisation underpins a wide range of applications: score following (Dorfer et al., 2016; Fremerey et al., 2008; Peter et al., 2025b), automatic music transcription (Benetos et al., 2019), score annotation (Ballester et al., 2025; Hentschel et al., 2025), expressive performance modelling (Berndt, 2021; Cancino-Chacón et al., 2018), automatic accompaniment (Cancino-Chacón et al., 2023; Dannenberg and Raphael, 2006), version mapping (Grünbacher and Neuwirth, 2024) and cross-format annotation transfer (Hu and Widmer, 2023). Within DM, there is growing interest in integrated data resources that connect diverse representations of the same music (e.g. Dreyfus et al., 2025).

Cross-domain alignment typically requires projecting data into a shared feature representation (for instance, chroma features derived from both audio and symbolic data (Gómez, 2006)) or using deep learning to map different modalities into a common embedding space (Dorfer et al., 2018; Müller et al., 2019; Weiß and Peeters, 2021). Synchronisation algorithms such as Dynamic Time Warping (DTW) then establish temporal correspondences (Müller, 2021). The process is complicated by feature discrepancies between domains (Thomas et al., 2012), expressive performance variations (Arzt and Widmer, 2015; Dannenberg and Mukaino, 1988) and structural differences such as repeated sections and jumps that require unfolding before alignment can proceed (Agrawal et al., 2021; Brazier and Widmer, 2021; Fremerey et al., 2010; Peter et al., 2025a).

The critical point for our purposes is that every one of these tasks produces alignment data – a set of correspondences between events on different timelines – yet these data are typically stored in task-specific, *ad hoc* formats (if stored at all), making them expensive to reproduce and difficult to build upon. No widely applicable standard for distributing these data in a reusable way yet exists.

2.3 EXISTING APPROACHES AND THEIR LIMITATIONS

Any music encoding format constitutes a model of musical time, at least implicitly. Research data that depend on music representations (such as score annotations, audio features and electroencephalography data aligned with auditory stimuli) typically adopt their time model or overlay their own, often equally implicit, model. Interoperability problems arise when these time models are not explicitly communicated.

Task-specific formal models. Formal mathematical models of musical time have been developed for specific use cases. For instance, Foscarin's (2020) model for automatic music transcription defines timelines in the physical and logical domains for contiguous events but does not generalise beyond this context.

Semantic Web approaches. The Timeline Ontology (Abdallah et al., 2006) and Segment Ontology (Fields et al., 2011) use Semantic Web technologies to model music events and temporal relations. The Timeline Ontology operates with two principal classes, ContinuousTimeLine and DiscreteTimeLine, each with respective time instants and intervals (adopted from the OWL-Time Ontology[3]), connected by TimeLineMaps (UniformSamplingMap, ShiftMap, etc.). The Segment Ontology extends this with SegmentLine, Segment and SegmentLineMap. Despite their potential generality, these ontologies have been applied principally to audio recordings and signal processing (Cannam et al., 2010), as exemplified by the CALMA dataset (Page et al., 2017) and, more recently, the Music Encoding and Linked Data (MELD) framework (Dreyfus et al., 2025; Weigl and Page, 2017).

Storage formats. The match file format (Foscarin et al., 2022) stores aligned note events for score–performance pairs. Music Performance Markup (MPM) (Berndt, 2021) provides a toolbox and exchange format for alignment and annotation data across the logical, graphical and physical domains. IEEE 1599 (Baggi and Haus, 2009, 2013) is an XML standard for multimodal alignment that references external sources from all three domains and represents synchrony by projecting events onto a single abstract timeline called the 'spine'.

De facto practices. Many datasets do not follow a fully specified format. Multimodal platforms such as Repovizz (Mayor et al., 2013) organise data streams along a shared physical timeline. In Optical Music Recognition, alignments between images and symbolic representations are typically encoded as *ad hoc* JSON or XML files enumerating bounding boxes (Calvo-Zaragoza et al., 2020).

Despite these contributions, several gaps remain:

1. **No cross-domain coverage.** No existing format or model covers all three temporal domains simultaneously. The Timeline and Segment ontologies focus on the physical domain, IEEE 1599 references all three but does not model their internal structure and the match file format is limited to scores and performances.
2. **No reusable alignment artefact.** There is no standard way to represent the alignment *itself* (the set of correspondences between events on different timelines) as a shareable, self-describing artefact independent of the datasets it connects.
3. **Ambiguous encoding conventions.** Measure numbers, beat sizes and coordinate systems vary between datasets and are rarely documented, making data reuse error-prone even within a single domain.
4. **No provenance or certainty metadata.** The agent, method and confidence level of alignment claims are almost never recorded, making it impossible to assess the reliability of existing alignments or to combine them systematically.
5. **No runnable toolchain.** No open-source library exists for creating, validating and exchanging cross-domain alignments with typed timelines and conversion maps.

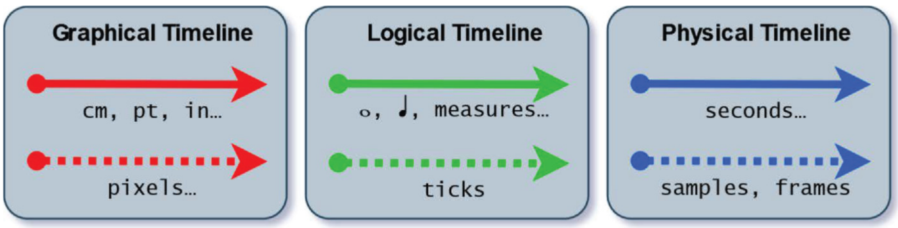
3 A CROSS-DOMAIN TIMELINE MODEL IN PRACTICE

The *TimeToAlign!* model organises musical timelines and their alignments as a hierarchy of typed containers connected by two distinct mechanisms. Each timeline is a positive coordinate axis – continuous (real numbers) or discrete (integers) – defined by an origin and a measuring unit. As depicted in Figure 3, we divide timelines into the three domains introduced previously: graphical (pixels, millimetres), logical (beats, ticks, quarter notes) and physical (seconds, samples), yielding six timeline types. Figure 3a shows the visual vocabulary used throughout the following examples: timelines are drawn as arrows whose colour indicates domain (red = graphical, green = logical, blue = physical) and whose style indicates modality (solid = continuous, dotted = discrete). *ConversionMaps* (Figure 3b) appear as thin, double lines, applying the same conventions according to the target unit; they are responsible for converting between time coordinates of the same or different units. Coordinates appear throughout the figures as triangular *wedges* pointing at timeline positions (Figure 3c). A timeline may nest *child timelines* that share its measuring unit, enabling hierarchical decomposition. As shown in Figure 3c, children typically terminate with hollow diamonds to indicate they are *locked* – their length fixed at embedding time – preventing unwanted side effects. Coordinates transfer between parent and child by exact offset arithmetic: a coordinate on the parent propagates to each child by subtracting the child's offset, a mechanism that recurses through any nesting depth.

Central to the model is the *ConversionMap* (C-map): a typed function attached to a timeline that maps each coordinate to exactly one value: usually another coordinate of a different measuring unit. C-maps are typically attached to a single timeline; connecting disparate timelines is the role of two further mechanisms.

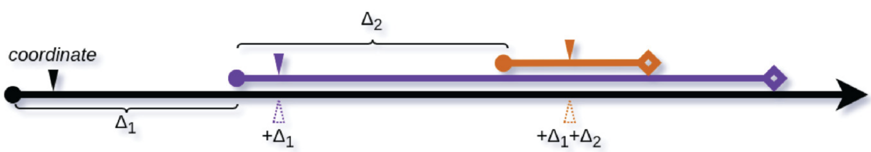
The first mechanism for connecting timelines is the *TimelineGroup*: it collects commensurable timelines and enables interpolation-based coordinate transfer between its members. The second mechanism is the *MatchClaim*: an explicit claim, issued by a human or algorithmic agent, that events on two different timelines are equivalent, carrying metadata on agent, method and certainty. MatchClaims produce *AlignmentAnchors* (coordinate pairs recording which points on two timelines correspond) from which ordered *MatchLines* and interpolation-based *WarpMaps* are derived for coordinate transfer across groups. An *AlignmentBundle* collects multiple groups and their inter-group MatchClaims into a single navigable structure.

Each connecting structure has an associated multi-coordinate accessor. A *TimeStamp* is a cross-section through a TimelineGroup at a chosen coordinate, capturing the coordinates of all member timelines plus the values converted by every attached C-map (a single timeline with its C-maps counts as a group of one and yields TimeStamps as well). A *MatchStamp* is the analogous accessor for an AlignmentBundle: the union of TimeStamps from groups synchronously connected by a MatchGraph. The reader may wish to consult Table 2 and



(a) Six types of timelines and typical measurement units. Each unit can be unambiguously attributed to either a continuous timeline (continuous lines) or a discrete timeline type (dotted lines).

(b) Six types of ConversionMaps, styled according to the timeline type associated with their target unit. For instance, a ConversionMap to WAV format would result in a discrete physical timeline and shown as a dotted blue double line.



(c) Child timelines and coordinate propagation. A coordinate (black wedge) on the parent timeline propagates to nested children via exact offset arithmetic. Child timelines terminate with hollow diamonds indicating they are length-locked.

Figure 3 Visual representation of our model's core components: timelines, ConversionMaps (C-maps) and parent–child relationships.

Term	Definition
AlignmentAnchor	A coordinate pair associating one coordinate on timeline A with one on timeline B; a neutral record with no claim semantics.
AlignmentBundle	A collection of TimelineGroups with inter-group match connections; provides coordinate transfer via offset arithmetic, interpolation and WarpMaps.
Break	A control event that voids contiguity at its instant; intervals cannot span a break.
Child timeline	A timeline nested within a parent, sharing its measuring unit but defining its own origin.
Commensurability	Two timelines are commensurable if connected by a chain of C-maps, membership in the same TimelineGroup or cross-group MatchClaims.
ConversionMap (C-map)	A typed function attached to a timeline that maps each coordinate to exactly one value (coordinate, label or filename); also permits inverse conversion back to the timeline's own unit.
Coordinate	A positive number representing distance from a timeline's origin.
Event	Anything associated with a timeline via instants (InstantEvent or TimeIntervalEvent).
FlowMap	A sequence of TimeIntervals specifying a particular path through a score's control structure (repeats, jumps); not itself a timeline. Used for unfolding.
Instant	A specific point on a timeline, associated with a coordinate.
Jump	A control event linking a 'jump-from' to a 'jump-to' instant, creating non-linear contiguity.
MatchClaim	A claim of equivalence between events on different timelines, with provenance metadata (agent, method, certainty). May be synchronous (producing AlignmentAnchors) or conceptual (structural equivalence without temporal commitment).
MatchGraph	An auxiliary structure formed by extending MatchClaims through shared coordinates; may be viewed as a hyperedge connecting multiple coordinates from different timelines.
MatchLine	An ordered sequence of AlignmentAnchors derived from MatchClaims, sorted by coordinate on a source timeline; input for WarpMap generation.
MatchStamp	The union of TimeStamps from groups synchronously connected by a MatchGraph.
NOMATCH	A sentinel value explicitly encoding the absence of a corresponding event.
Region	A named part of a timeline defined by a TimeInterval (e.g. 'Verse', 'Bridge').
RotationMap	A periodic ConversionMap that wraps coordinates modulo a pattern length (e.g. a looped ostinato).
Segment	A child timeline that is contiguous with its siblings within a parent timeline.
SegmentLine	A parent timeline containing only contiguous segments.
Timeline	A positive coordinate axis defined by an origin and a measuring unit; continuous (real) or discrete (integer).
TimeInterval	A left-inclusive, right-exclusive interval $[s, e)$ defined by a start and an end instant.
TimelineGroup	A container for timelines claimed to be commensurable; placing timelines in a group asserts commensurability regardless of underlying units. Coordinates transfer via interpolation.
TimeIntervalStamp	The interval analogue of a TimeStamp: a pair of TimeStamps marking the start and end of an interval, recording its extent in all connected coordinate systems.
TimeStamp	A cross-section through a timeline or group at a given coordinate, capturing positions in all connected systems plus values converted by all attached C-maps.
WarpMap	An interpolation-based coordinate mapping derived from a MatchLine; enables non-linear coordinate transfer between TimelineGroups.

Table 2 Glossary of key terms in the TimeToAlign! model. This table collects the terms introduced across Sections 3.1–3.5; the full glossary, including additional terms, is available on the documentation homepage.

the full glossary on the documentation homepage as a reference throughout.[1]

Rather than enumerating formal definitions, we demonstrate these concepts through five use cases of increasing complexity. Sections 3.1 and 3.2 introduce the core concepts behind our model of individual timelines and groups: the first through a piano-roll alignment scenario and the second through a school song with repeat structure. Section 3.3 demonstrates coordinate transfer and MatchClaim creation through

C-maps between graphical analyses. Section 3.4 illustrates a large multimodal alignment across all three domains at scale. Section 3.5 extends the use of alignments via *MatchGraphs* – which may be viewed as hyperedges connecting multiple coordinates from different timelines – and demonstrates more nuanced musicological application through conceptual (rather than synchronous) MatchClaims. Each of the five examples exists as a runnable notebook using the `timetoalign` Python implementation, available in the How-to section of the documentation.

3.1 ALIGNING A HISTORICAL PIANO ROLL WITH A DIGITAL SCORE

The Stanford University Piano Roll Archive (SUPRA) provides, for each historical Welte–Mignon roll, a high-resolution scan, a file of hole-punch positions, two MIDI files (one preserving the raw hole data, one with expressive dynamics) and a synthesised MP3 (Shi et al., 2019). Consider a researcher who wishes to build a player application that displays a digital score alongside the piano roll and plays the synthesised audio, while a cursor indicates the corresponding position in the scan (cf. Ballester et al., 2025). The challenge is that the five artefacts span all three temporal domains and use incommensurable coordinate systems: the scan measures 299,400 pixels in height; the raw MIDI file contains 30,092 events counted in ticks; the MP3 lasts approximately 573 seconds; and the annotated score counts 888 quarter-note beats across 222 measures, carrying 1,074 harmonic labels. No documented alignment between these representations exists.

Figure 4 shows how we model this scenario. Each artefact becomes a *timeline*: a positive coordinate axis defined by an origin and a measuring unit. The scanned roll is a *discrete graphical timeline* (DGT1, the red dotted arrow at the bottom of Figure 4), measured in pixels. Although the timeline is discrete according to the scan, its C-map (the red double line below DGT1) maps each coordinate to the material real-world position in inches, a first demonstration of the ConversionMap concept. The annotated score is a *continuous logical timeline* (CLT1, the green solid arrow at the top), measured in quarter-note coordinates that come as fractions for precision; a discrete logical C-map (the green double line) additionally converts beats to measure numbers. The synthesised audio is a *discrete physical timeline* (DPT1, the blue dotted arrow), measured in samples, with a C-map to seconds. The two MIDI files are *discrete logical timelines* (DLT1, DLT2, the green dotted arrows), measured in ticks (sometimes called pulses, an integer-valued unit suited to quantised performance data), each carrying a

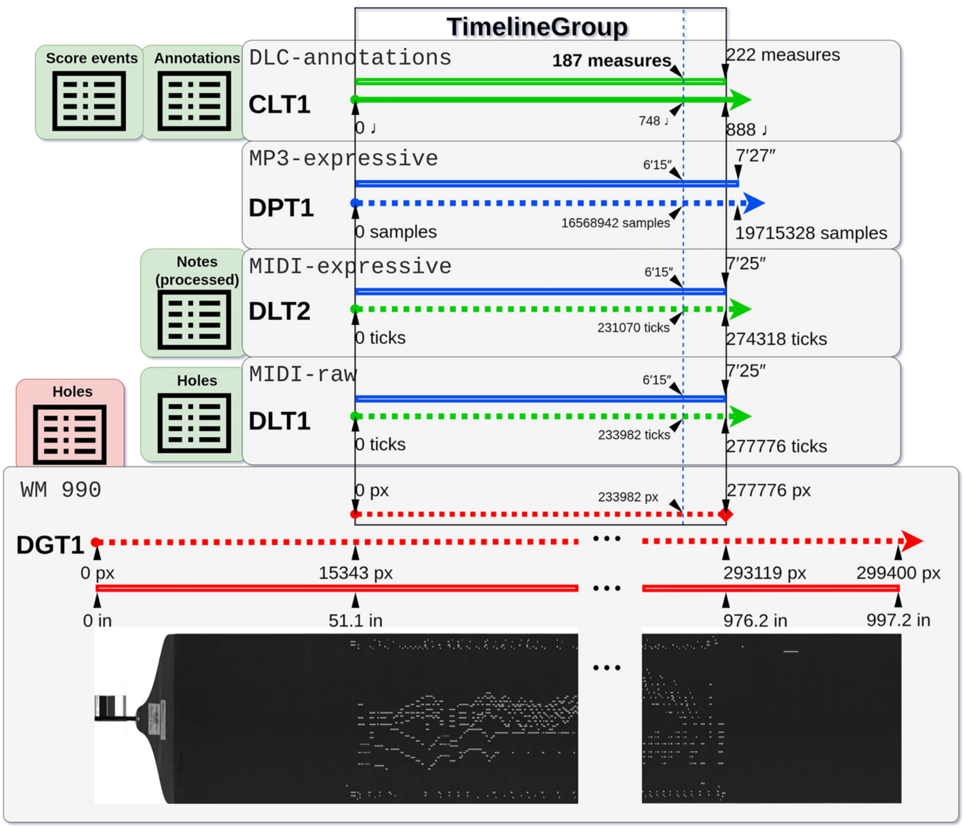


Figure 4 Modelling the alignment between a historical piano roll (DGT1, **red dotted arrow**), its region of musical content (DGT1 child), two MIDI files (DLT1, DLT2, **green dotted arrows**), a synthesised audio recording (DPT1, **blue dotted arrow**) and an annotated score (CLT1, **green solid arrow**). C-maps appear as thin, double lines coloured by their target domain. The **black rectangle** denotes the TimelineGroup; **the blue dashed vertical line** shows a TimeStamp cross-section at a chosen coordinate.

C-map to seconds. The distinction between discrete and continuous timelines within the same domain is not merely typographic: it determines the permissible arithmetic (integer vs real-valued interpolation) and prevents silent precision loss when coordinates are transferred. Each of these C-maps also permits the inverse conversion back to the timeline's own unit, which proves highly convenient for addressing time points and intervals in whichever coordinate system is most natural for the task or data at hand.

Because only a portion of the scanned roll contains musical content (pixel rows 15,343–293,119), we model this region as a *child timeline*: a nested coordinate system (the shorter red dotted arrow inside DGT1's box) that shares the parent's measuring unit but defines its own origin. In the child, the first hole punch sits at coordinate 0 rather than at absolute pixel 15,343, simplifying all downstream references while preserving the relationship to the parent via a known offset. The C-map on the parent reveals that the musical content spans 23.5 metres of paper, enabling data to be referenced not only in pixel coordinates but also in real-world positions on the material roll in the archive.

To make all five artefacts mutually navigable, we collect them in a *TimelineGroup* – the black rectangle in Figure 4. By placing timelines into a group, we claim that they are *commensurable* – regardless of their underlying units or the presence of C-maps. The group is an important tool that enables interpolation between timelines and allows one to obtain *TimeStamps*: cross-sections showing the coordinates of all member timelines plus the respective values converted by every attached C-map. Figure 4 shows a TimeStamp as blue dashed vertical lines through the group at a chosen coordinate, with interpolated values labelled on each timeline: a single query that returns positions in all five coordinate systems simultaneously. For our player application, a TimeStamp at each beat position provides exactly the data needed: the pixel row in the scan, the sample position in the audio and the tick offsets in both MIDI files. Note, however, that the timelines in this group are not genuinely *aligned*: the group provides interpolation-based coordinate transfer, which gives a rough estimate but not a musically verified correspondence. To create a genuine alignment, one would match the scanned holes to processed MIDI events to score notes – a straightforward task for this particular example, given the data available, and a workflow that the model supports through MatchClaims (Section 3.3).

In sum, this example has rendered five artefacts across three temporal domains mutually navigable through a handful of typed containers and maps – including, crucially, the distinction between interpolated group membership and verified alignment that the model makes explicit from the outset.

3.2 FLOW CONTROL IN A SCHOOL SONG ARRANGEMENT

Consider a digitised double page of a school song arrangement: *Un poquito cantas* from the *Chorissimo! Blue* songbook (Weigele and Brecht, 2016). It contains repeat signs, a *dal segno al coda* structure and a percussion ostinato notated on a separate staff (Figure 5). A score-following application must know which bars to play in what order and how the seven staff systems spread across two pages map onto the logical structure.

The graphical layout is a *SegmentLine* (DGT1, the red dotted arrow at the bottom of Figure 5): a sequence of contiguous *Segments* – that is, child timelines joined end to end. Each of the seven segments (seg1–seg7, visible as the red diamond connectors) corresponds to one staff system, with pixel widths ranging from 646 to 698 px. Each segment carries a single C-map (the green double line above): a quick fix that simply divides the segment's width into four equal parts for a rough mapping to measure numbers. A more precise mapping would be conceivable by creating an array of *x*-coordinates of all measure boundaries, which could either constitute the segment line or actually be embedded as children in their respective system segments.

The logical structure (CLT1, the green solid arrow in the centre of Figure 5), measured in quarter-note beats, introduces three new concepts. First, named *Regions* (curly braces above CLT1) partition the timeline into semantically meaningful parts without creating separate child timelines. Six Regions are visible: Intro, Core (repeated), Core (last time), Verse, Chorus and Coda; the Verse and Chorus Regions group further into the two different Core Regions. Second, *Breaks* (the red vertical lines crossing CLT1 and DGT1) void contiguity at specific coordinates: the break at the Coda mark prevents intervals from spanning the boundary between the main body and the Coda, correctly encoding that these sections are not temporally adjacent in performance. Third, *Jumps* (the dotted curved arrows below CLT1) create non-linear contiguity: the jump from the end of the Chorus back to the Verse encodes the repeat sign; the jump from the end of the second pass to the Coda encodes the *dal segno al coda*.

Breaks and jumps are *flow control events* – a feature of our model that determines all possible flows we can compute for a given timeline. One of these is the *Default FlowMap* shown in Figure 5, which our model implementation computes automatically from the notated structure. Apart from such automatically computed flows, users can define custom flows as any sequence of interval events, including Regions. Different performances may follow different paths: one choir may take all repeats, while another may skip a verse. A *FlowMap unfolds* the timeline by slicing it at section boundaries and concatenating the slices into a new SegmentLine – the mechanism used at scale in the Beethoven example (Section 3.4).

Figure 5 Modelling flow control in a school song arrangement. **DGT1** (bottom, **red dotted arrow**): Discrete graphical timeline representing two scanned pages decomposed into seven staff-system segments (red diamond connectors). **CLT1** (centre, **green solid arrow**): Continuous logical timeline representing the arrangement's 24 measures, with named Regions (curly braces: Intro, Verse, Chorus, Coda), Breaks (**red vertical lines**) and Jumps (**dotted curved arrows**) encoding the repeat signs and *dal segno al coda* structure. **CLT2** (right, **green solid arrow**): Continuous logical timeline representing a four-bar percussion ostinato mapped via a RotationMap (**purple box**). Two groups (**black rectangles**) enclose the main body and the ostinato, respectively. The **Default FlowMap** (purple box, left) shows the traversal path. **Blue dashed vertical lines** show a TimeStamp cross-section.

A second logical timeline (CLT2, the green solid arrow on the right of Figure 5) represents the four-bar percussion ostinato. Because the ostinato repeats cyclically throughout the song, it is associated with a *RotationMap* (the purple box labelled $f(n) \mapsto n \cdot$Ostinato): a periodic ConversionMap that wraps coordinates modulo the pattern length. The RotationMap acts like a generator, returning the ostinato as many times as the flow requires. Figure 5 shows two groups (the black rectangles): one enclosing CLT1 and DGT1 (the main body) and one enclosing CLT2 (the Ostinato). The blue dashed vertical lines in Figure 5 mark a TimeStamp at a chosen logical coordinate. FlowMaps may also be applied to groups: the idea here would be to unfold the ostinato (timeline or group) as many times as needed and to unfold the main body according to the appropriate flow, then either combine the timelines from the resulting groups into another group (if the lengths are known to match) or add them to each other as children (e.g. the unfolded CLT1 as a child to the unfolded CLT2, assuming the performance starts with the ostinato alone, and the unfolded DGT1 'body' as a child to the unfolded DGT1 'ostinato').

The flow-control apparatus demonstrated here – at the modest scale of a 24-measure school song – handles the full complexity of repeat signs, *dal segno* and cyclic ostinati without duplicating any underlying data. That the same mechanism applies without modification to a Classical string quartet movement is demonstrated in Section 3.4.

3.3 TRANSFERRING ANNOTATIONS BETWEEN GRAPHICAL ANALYSES

Consider a researcher studying Lasse Thoresen's spectromorphological analysis of Åke Parmerud's acousmatic composition *Les objets obscurs*, composed in 1991. The same analysis was published in two versions: as a single raster image in Thoresen (2009) and as five separate images, each extended with a form-analytical layer beneath, in Thoresen (2010). Both depict the same 150-second audio excerpt, but with different pixel dimensions. An annotation placed on one image – say, the bounding box of sound object H – cannot simply be copied to the other: the pixel coordinates differ, the images are laid out differently and no documented mapping between them exists.

We model each published version as a *discrete graphical timeline* (DGT1 and DGT2, the red dotted arrows in Figure 6). Each image's multi-line layout is decomposed into a *SegmentLine* (the concept introduced in Section 3.2) with one Segment per visual system. DGT1 (the 2009 version) has five segments of equal length (975 pixels each, totalling 4,875 pixels); DGT2 (the 2010 version) has five segments of slightly varying length (864–867 pixels each, totalling 4,328 pixels). A C-map attached to each timeline (the blue solid line labelled 'SamplesToSeconds' in Figure 6) maps its pixel coordinates to seconds on a shared continuous physical timeline representing the audio recording. For DGT1, a single linear C-map maps 4,875 pixels to 150 seconds – again, a pragmatic decision

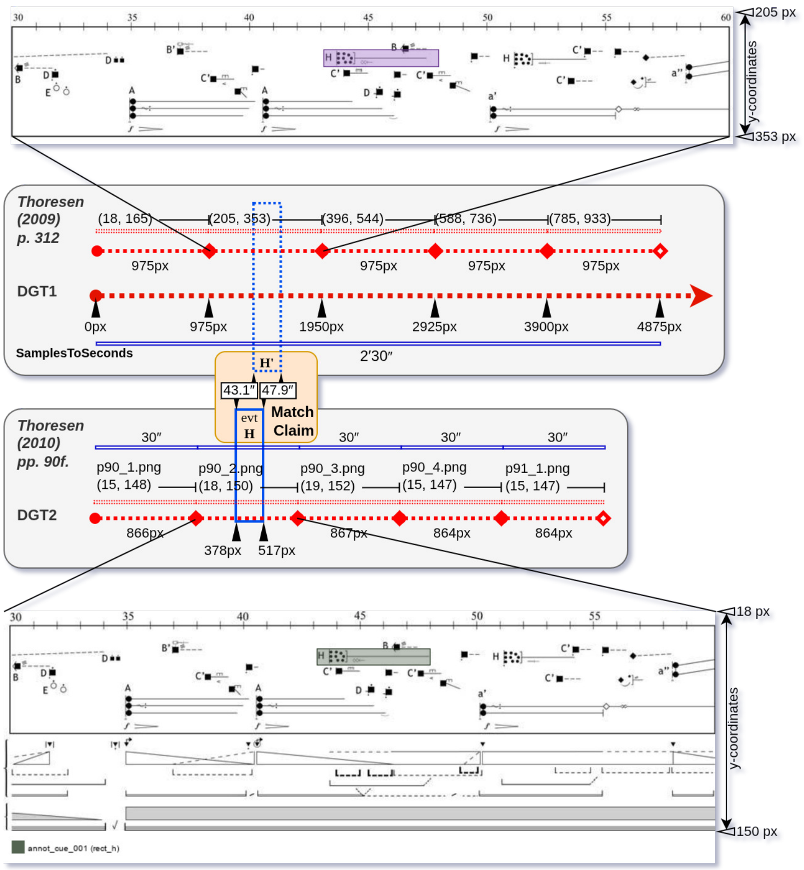


Figure 6 Transferring a graphically defined event H from one image to another. DGT1 and DGT2 (**red dotted arrows**) are discrete graphical timelines with five segments each (red diamond connectors), representing two published versions of the same spectromorphological analysis. C-maps (**blue solid lines**) convert pixel coordinates to seconds on a shared continuous physical timeline. The peach box labelled 'Match Claim' represents the MatchClaim connecting event H across the two images; **blue vertical lines** through the box show the TimeIntervalStamps of the event's temporal boundaries.

that could be modelled more precisely if circumstances required it; for DGT2, five separate linear C-maps each map one segment's pixel length to 30 seconds.

For event H, whose pixel coordinates in DGT2's second segment span 378 to 517 px, the associated *TimeIntervalStamp* (effectively a pair of TimeStamps marking start and end) records this pixel range together with the physical time in seconds (43.1"–47.9") derived from the attached C-map; DGT1 produces its own TimeIntervalStamp for event H' (both visible as the coordinate tuples in Figure 6).

To assert that event H in DGT2 and event H in DGT1 represent the same sound object, we create a *MatchClaim*: an explicit claim connecting events across timelines (the peach box labelled 'Match Claim' in Figure 6). The MatchClaim carries structured metadata: the agent who made the claim, the decision criteria and the certainty level. In the simple version, the transfer is purely coordinate-based: H's pixel *x*-coordinates in DGT2 are converted to seconds via the C-map and then back to pixels in DGT1 via the *inverse* of DGT1's C-map. In a more elaborate version, one can also transfer the *y-coordinates*, which represent event properties (the vertical extent of the sound object in the image). By defining a custom C-map that interpolates between the *y*-segments of DGT1 and DGT2 (visible as the vertical arrows and hollow coordinate wedges in Figure 6), the vertical positions are warped from one image to the other.

The same principle extends to any image that encodes time spatially (for instance, piano-roll scans, sonograms or even the spiral notations of George Crumb's *Makrokosmos*) by defining the appropriate mapping from pixel coordinates to physical or logical time coordinates. What appears to be a tricky problem – transferring information between two heterogeneous versions of the same graphical analysis – turns out to be conceptually straightforward thanks to the model's core machinery: SegmentLines, C-maps with inversion and MatchClaims with structured provenance.

3.4 MULTIMODAL ALIGNMENT AT SCALE

This example, the most elaborate in our collection, demonstrates how the model scales to a complex, real-world research scenario spanning all three temporal domains (Figure 7). A performance researcher has access to three multimodal recordings of the fourth movement of Beethoven's *String Quartet no. 4*, op. 18/4, captured in a controlled environment with six microphones, motion-capture sensors and derived audio features at varying sampling rates (Marchini et al., 2014). The underlying dataset catalogues well over a hundred signals per recording: audio, motion capture, bowing gestures and Essentia audio descriptors. Two digital score encodings are available: one from the Annotated Beethoven Corpus (ABC), version 2.6, which carries harmony and phrase analyses (Neuwirth et al., 2018), and one from the OpenScore String Quartet corpus (Gotham et al., 2023b), which encodes all four movements in a single file. OMR ground-truth data provide pixel-level bounding boxes for 3,190 note heads across 22 score pages (Shatri and Fazekas, 2021). The researcher's goal: inter-relate the harmony and phrase annotations with the performances and their sensor data while correctly handling the movement's repeat structure.

Figure 7 is subdivided into three zones. The upper zone (white background, right) shows the *Folded Score Group*: CLT1 (the green solid arrow), the ABC v2.6 encoding with 878.5 quarter-note beats across 13 atomic sections (A–M), and DGT1 (the red dotted arrow), the OMR ground-truth data. The score contains repeats and alternative endings (the same kind of flow-control structure modelled via Breaks and Jumps in Section 3.2), meaning that notated measures do not represent the performance order. The *Default FlowMap* (the lettered blocks at the right of the score group) specifies the traversal according to conventional unfolding logic. In the upper-left corner sits the *Multimodal Recording Group* (Marchini et al., 2014), shown as a black rectangle. We show only a representative subset: five discrete physical timelines (blue dotted arrows) for the main modalities – recording, tonal features, low-level features, rhythm features and motion capture – each with children (visible as stacked segments inside the labelled boxes) representing individual musicians. Every timeline carries a C-map to seconds. The dataset contains two additional recording groups ('Mechanical' and 'Exaggerated') with a similar structure, which we omit from the figure for clarity.

The black rectangle in the middle shows the *Unfolded Score Group*: CLT1_unfolded (the green solid arrow, 1,116 quarter-note beats) and DGT1_unfolded (the red dotted arrow, 135,198 pixels). The unfolded section sequence (A, A, B, B, C, D, E, D, . . ., M) now runs linearly, expanding the 878.5 quarter-note beats to 1,116 quarters. The data stores (Notes, Labels) are carried along and multiplied by the unfolding operation.

The lilac background represents the first *AlignmentBundle*: a structure that combines the Multimodal Recording Group with the Unfolded Score Group via inter-group MatchClaims. The orange *MatchClaims* box at the boundary between the two zones represents these claims, each asserting that a note event in the recording corresponds to one in the unfolded score. These event pairs resolve to *AlignmentAnchors*: coordinate pairs recording which point on the score corresponds to which point on the recording, according to the respective events' start and end coordinates. At this point, harmony and phrase annotations can be transferred from the score to the motion-sensor data and instrument-pickup recordings: Those coinciding with a note can be transferred directly; the remaining ones would necessitate the creation of a WarpMap (see further below).

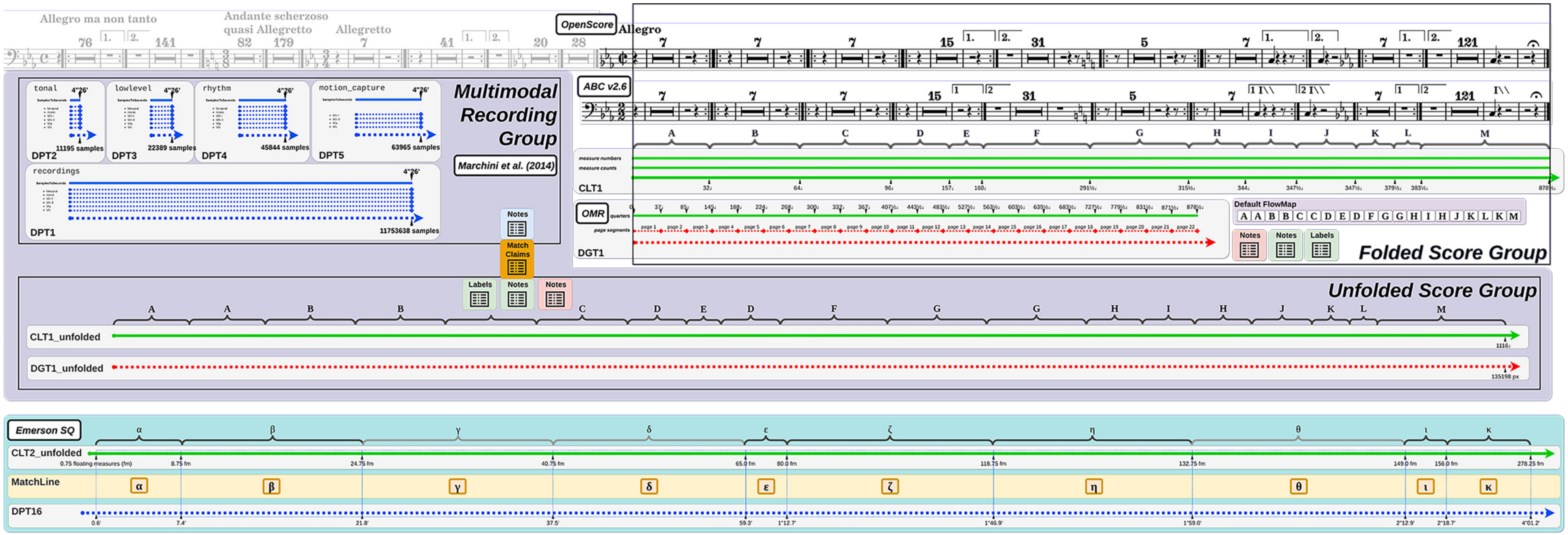


Figure 7 Multimodal alignment scenario for the fourth movement of Beethoven's *String Quartet no. 4*, Op. 18/4. The figure is divided into three zones. **Upper zone** (white background): the *Folded Score Group* (right, containing CLT1 and DGT1) and the *Multimodal Recording Group* (left, containing five **blue dotted arrows** for recording and feature timelines, each with stacked children). The **Default FlowMap** (lettered blocks **A, A, B, B, C . . . M**) shows the traversal path. **Middle zone** (lilac background): the *Unfolded Score Group* (CLT1_unfolded, DGT1_unfolded), representing AlignmentBundle #1. The orange **MatchClaims** box connects the Recording Group to the unfolded score. **Lower zone** (cyan background): AlignmentBundle #2 with CLT2_unfolded (**green solid arrow**), the *MatchLine* (orange boxes α–κ in **dotted blue** AlignmentAnchor rectangles) and DPT16 (the Emerson String Quartet recording, **blue dotted arrow**).

The lower zone (cyan background) represents a second, independent *AlignmentBundle*. It contains CLT2_unfolded (the green solid arrow, an earlier ABC v1.0 edition measured in floating-point measures) and DPT16 (the blue dotted arrow, the Emerson String Quartet recording, 4'01.2"). Between them sits a *MatchLine*: an ordered sequence of MatchClaims shown as orange boxes (α–κ), each within a dotted blue box representing its AlignmentAnchor. Each Greek letter corresponds to a Region of the unfolded score; the MatchLine thus communicates a coarse-grained score–audio alignment encoded as a Flow CSV in the dataset.

Now, the key insight: the two AlignmentBundles are currently independent. However, CLT1_unfolded and CLT2_unfolded encode the same music – merely from different editions. By adding CLT2_unfolded to the *Unfolded Score Group*, we bridge the two bundles: all transferable data from the Marchini recordings becomes available to the Emerson timeline and vice versa. This cascading alignment – connecting independent datasets by aligning one timeline each – is the central payoff of the AlignmentBundle architecture. The practical consequence is that a single additional group membership – adding an independent score edition to an existing group – retroactively enriches every timeline already present in both bundles, without modifying any of the original data or alignment claims.

3.5 ENCODING CONCEPTUAL AND TEMPORAL RELATIONSHIPS IN SONG GENESIS

The previous examples dealt with alignment as a fundamentally direct temporal relationship: events on different timelines correspond because they occur at equivalent points in time. Musicological research, however, often requires a broader notion of temporal correspondence. Consider a study of the genesis of Jimi Hendrix's *1983 . . . (A Merman I Should Turn to Be)* that compares three versions: a solo demo (CPT3), a band demo with Mitch Mitchell on drums (CPT2) and the studio recording on *Electric Ladyland* (CPT1). The three versions share a recognisable form – all have an Intro, several Verses and a Bridge – but sections are added, removed and restructured across versions. The instrumental middle section present in the studio version and the band demo is entirely absent from the solo demo. Some correspondences are merely conceptual ('both have a Bridge'), while others are also temporal ('these Bridge subsections can be aligned beat-by-beat'). How can these heterogeneous relationships be encoded in a single, queryable structure?

As introduced in Section 3.3, a *MatchClaim* is an explicit claim, issued by a human or algorithmic agent, positing that events from different timelines are equivalent or synchronous, carrying structured metadata on the agent, decision criteria and certainty. MatchClaims can be automatically extended into *MatchGraphs* through a coordinate that they share with one or multiple other MatchClaims. A MatchGraph is an auxiliary construct for creating a MatchStamp, which is the union of TimeStamps from the groups synchronously connected by a MatchGraph. However, in other cases, we may want to view MatchGraphs as *hyperedges* connecting multiple coordinates from different timelines. Critically, a MatchClaim may be either *synchronous* (asserting both structural and temporal equivalence, thereby producing AlignmentAnchors) or *conceptual* (asserting structural equivalence without temporal commitment). One can create MatchGraphs to include synchronous claims, conceptual claims or both; however, *MatchStamps* (like those shown as dotted blue lines in the figure) can only ever include TimeStamps from those groups where at least one MatchClaim is synchronous. Figure 8 shows 15 section-level MatchGraphs (M1–M15), each formed from one or more MatchClaims. M1 (Intro) is purely conceptual: all three versions have an introduction, but they differ harmonically and temporally. M3 (Verse A1) is synchronous across all three versions, enabling beat-level alignment. M12 (Instrumental Part) explicitly records a *NOMATCH* sentinel for the solo demo: a positive assertion that CPT3 has no corresponding section, rather than a mere absence of data. These sentinel values make structural omissions first-class objects in the alignment structure, preventing ambiguity about whether a gap represents missing data or a genuine musical absence.

The conceptual/synchronous distinction proves its worth in the Bridge section. M5 asserts a conceptual MatchClaim between the Bridge sections of all three versions, but, because the sections differ in length (the solo demo has an extra opening measure; the studio version has additional iterations), no global synchronous alignment is warranted. However, four subsection-level MatchClaims (M6–M9) *are* synchronous, yielding AlignmentAnchors that enable a fine-grained chord comparison (excerpt shown in the bottom-right rectangle of Figure 8). This reveals Hendrix's harmonic evolution: the solo demo mostly uses a three-chord progression (D–A–C), matching the fourth chord (G) only in the final iteration; the band demo introduces passing chords ($G^{7\#9}$) absent from the other versions; and the studio version settles on a clean four-chord cycle but extends the section with additional instrumental iterations. A *MatchLine* – an ordered sequence of AlignmentAnchors derived from these MatchClaims, sorted by coordinate on a chosen source timeline – serves as the input for WarpMap generation and enables systematic comparison across versions.

Throughout, the three primary timelines (TiLiA annotations of the audio recordings) remain immutable.[4] The MatchClaims form an overlay that adds regions, sentinels and certainty metadata without modifying the source data – a separation of concerns that is essential when multiple analysts may produce competing alignment claims for the same material. The Hendrix

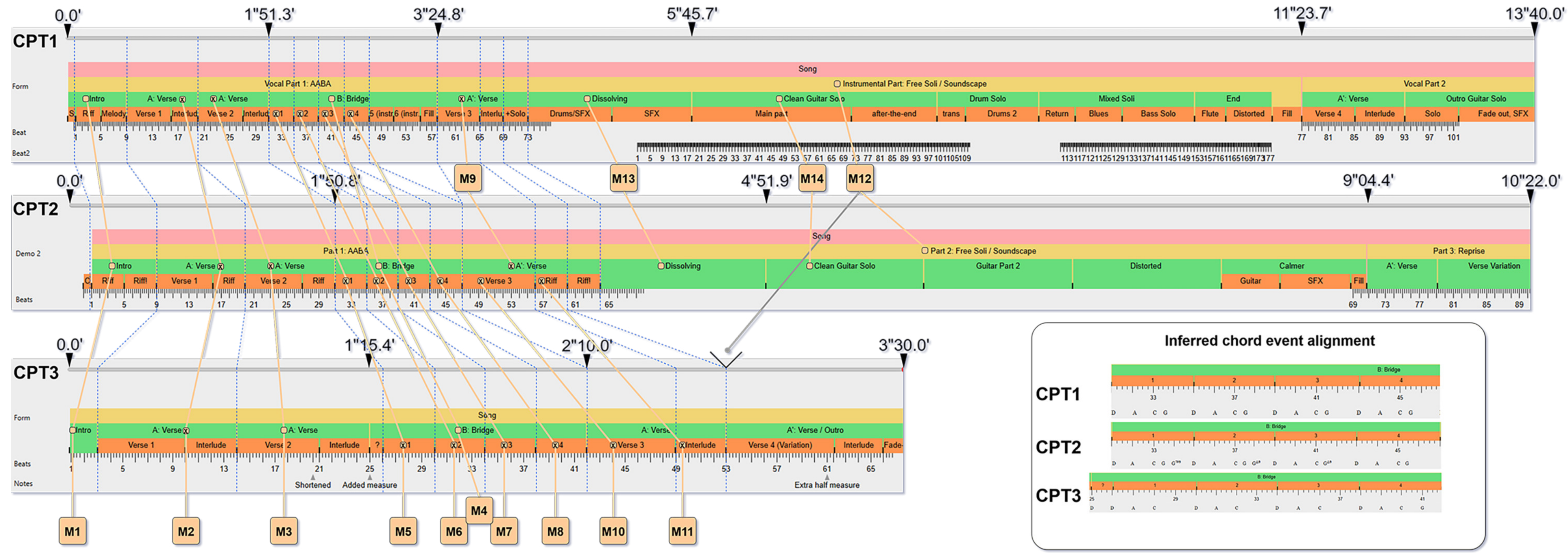


Figure 8 Conceptual and temporal alignment of formal analyses for three versions of Hendrix's *1983 . . . (A Merman I Should Turn to Be)*: a solo demo (CPT3), a band demo (CPT2) and the studio recording (CPT1). Peach rectangles represent MatchGraphs (M1–M15), each formed from one or more MatchClaims; those marked with **X** are synchronous as well as conceptual. Dashed blue lines indicate MatchStamps derived from synchronous MatchClaims. The lower-right inset shows how synchronous alignment of Bridge subsections (M6–M9) enables fine-grained chord comparison.

example is the most interpretively rich of the five: it shows that the same machinery used for temporal alignment serves equally for encoding musicological judgements about structural correspondence, absence, and evolution across versions of a work.

4 IMPLEMENTATION AND REPRODUCIBILITY

To turn the conceptual model of Section 3 into a working tool, we provide `timetoalign`[5], an open-source Python library released under the MIT licence, whose design is guided by four goals: *precision*, *scalability*, *interoperability* and *reproducibility*. The library is currently in beta; we present its architectural decisions at a level of abstraction that will remain valid as the API matures.

Precision. A recurring source of errors in alignment research is silent precision loss when fractional beat positions are stored as floating-point numbers (Section 1). To address this, every continuous coordinate in the library carries both a floating-point value and an exact fraction (numerator and denominator) so that a note at beat $2\frac{1}{3}$ is stored losslessly while remaining available for fast numerical operations. This dual representation is the foundation on which typed timelines, ConversionMaps and MatchClaims are built: the six timeline classes from Figure 3a enforce domain and continuity constraints at construction time, ensuring that a discrete graphical timeline measured in pixels cannot be silently confused with a continuous logical timeline measured in quarter-note beats.

Scalability. Musical datasets routinely contain tens of thousands of events. Events are therefore stored as tabular data rather than as individual objects, enabling bulk operations without per-event overhead. Coordinate transfer operates at three levels – exact offset arithmetic for parent–child relationships, linear interpolation within TimelineGroups and WarpMaps (derived from MatchLines) for cross-group alignment – each chosen for the appropriate trade-off between precision and generality.

Interoperability. The library can only be useful if it connects to data as they already exist. Specialised loader plugins parse common formats – MIDI files, MusicXML and MEI scores, audio files, IIIF image manifests and generic JSON, XML and CSV formats, among others – into typed timelines with appropriate ConversionMaps already attached. A MIDI loader, for instance, derives a ticks-to-seconds map from the tempo events; a score loader constructs a measure-beat grid with metric positions. A family of ConversionMap types covers the principal conversion patterns: linear mappings, lookup tables with interpolation, sequential composition, region-based dispatch, periodic maps (for cyclic structures such as ostinati) and metric-aware maps. Maps that produce coordinates (as opposed to labels or filenames) support inversion and composition so that complex coordinate chains can be built incrementally.

Reproducibility. A Parquet-based exchange format for publishing alignment data independently of the datasets it connects is under development. Because a timeline's identity depends only on its origin, unit and length, the alignment structure can be constructed from dimensional metadata alone, allowing one to share the full alignment topology – groups, MatchClaims, ConversionMaps provenance – even when the original data are large, proprietary or subject to copyright restrictions. This complements rather than competes with RDF/JSON-LD approaches: Table-based big-data paradigms are optimised for efficient bulk access, while Linked Data excels at semantic interoperability and federated queries.

The library ships with a set of tutorial notebooks, a documentation homepage with rendered examples and the full conceptual specification and a specimen collection with real-world data from the repertoires discussed in Section 3. The library, documentation and test data have been archived on Zenodo[6].

5 DISCUSSION AND FUTURE WORK

5.1 ACHIEVEMENTS

The TimeToAlign! model unifies the three temporal domains – graphical, logical and physical – under a single framework of typed timelines, ConversionMaps and MatchClaims. The five examples in Section 3 demonstrate that this framework handles scenarios ranging from simple two-image annotation transfer to a large-scale multimodal alignment spanning all three domains and that the same concepts apply uniformly across all of them.

A recurring theme across the examples is the role of *nesting*. Child timelines, Segments and SegmentLines allow complex structures to be composed from simple parts, while the coordinate arithmetic that accompanies nesting (offset transfer between parent and child, interpolation within groups, WarpMaps across groups) is handled by the framework rather than by the user. The Beethoven example illustrates this most directly: each of the recording group's physical timelines carries between four and six children, yet a single TimeStamp query traverses the entire hierarchy without the user specifying how coordinates are to be transferred.

A further consequence of the model's design is that alignment structures can exist independently of the original data they describe. Because a timeline is defined solely by an origin, a measuring unit and a length, it can be constructed from metadata alone – a manifest listing track durations, an image catalogue recording pixel dimensions or simply a known recording length – without requiring access to the underlying files. The resulting *time skeleton* carries the full alignment topology (MatchClaims, TimelineGroups, AlignmentBundles) and can be shared, archived and reused even when the original data are large, proprietary or subject to copyright restrictions. When new data become available – say, a fresh recording or a new labelset – they attach to the existing skeleton via a single additional MatchClaim (minimally), inheriting all previously established connections (as demonstrated in Section 3.4).

The `timetoalign` library moves these contributions from specification to a working tool. Typed timelines and loader plugins lower the barrier to constructing aligned representations: the SUPRA example (Section 3.1), which connects five artefacts across three domains, requires fewer than 10 lines of loader code and one TimelineGroup constructor. The planned Parquet-based exchange format will enable publishing alignment data as self-describing, shareable artefacts independent of the datasets they connect – addressing one of the central gaps identified in Section 2.3.

5.2 LIMITATIONS

The model and library have several limitations that should be stated plainly. First, automatic alignment algorithms (DTW, sequence alignment, deep learning–based matchers) are external to the library: `timetoalign` provides the data model for their inputs and outputs, not the algorithms themselves. Integration with existing toolkits such as Sync Toolbox (Müller et al., 2021) is a natural next step. Second, the logical domain as currently modelled is tailored to Common Western Music Notation: concepts such as measures, beats and time signatures are first-class objects, whereas other paradigms are not yet represented. The three-domain triad itself is not Western-specific (graphical, logical and physical time transcend any specific musical tradition) and extending the logical domain to accommodate alternative time units should be straightforward, provided they can be expressed as discrete or continuous coordinate axes. Third, no user study or adoption data exist yet. Fourth, the library currently requires Python programming; GUI tooling for non-programmer users is not yet available, although integration with annotation tools such as TiLiA is under consideration. Finally, the data-exchange format is not yet implemented.

5.3 RELATION TO EXISTING STANDARDS

Our model draws heavily on several existing paradigms. The Timeline Ontology (Abdallah et al., 2006) and Segment Ontology (Fields et al., 2011) provide the closest structural antecedents: their ContinuousTimeLine and DiscreteTimeLine classes, TimeLineMaps, SegmentLine and Segment all have direct counterparts in our model. We extend this paradigm in three directions: the graphical domain (yielding six timeline types rather than two), flow control (Breaks, Jumps, FlowMaps) and MatchClaim objects carrying provenance and certainty metadata. The two approaches are complementary: the ontologies provide a semantic foundation for Linked Data integration, while our model provides a computational framework for typed coordinate transfer.

IEEE 1599 (Baggi and Haus, 2013) shares the goal of multimodal synchronisation but organises all connections through a single abstract 'spine' that generally models the score. Our MatchLine concept addresses a limitation of this design: because MatchClaims and AlignmentAnchors are modelled in their own right – with metadata, uncertainty values and flexible assembly – a MatchLine can be constructed for *any* reference timeline, effectively creating a purpose-specific spine. Any timeline that is commensurable with two others can serve as an intermediary, and the connection topology is determined by the data rather than prescribed by the format.

The match file format (Foscarin et al., 2022) encodes aligned note events for score–performance pairs, always pairing events on the two aligned sides (or using sentinels for unmatched notes). Our model generalises this in two ways: a MatchClaim may assert equivalence between two events, between an event and its projection onto another timeline (via C-maps), or – though less informatively – between bare coordinates, and MatchClaims need not imply synchrony at all, catering to musicological research on structural correspondence without temporal commitment (as demonstrated in Section 3.5).

MPM (Berndt, 2021) covers logical-to-physical alignment with detailed performance markup. Our model is more general (covering all three domains) but less specialised for the fine-grained performance parameters (articulation, dynamics, rubato maps) that MPM provides. More generic alignment structures found in OMR datasets (Calvo-Zaragoza et al., 2020) inspired our treatment of graphical timelines and cross-domain bounding-box transfer.

As a consequence of these affinities, we envisage not only ingesting data from these formats – which the implementation already does to some extent – but also converting *to* them, enabling round-trip interoperability with established toolchains.

5.4 FUTURE DIRECTIONS

Several directions merit further development. The most immediate is the Parquet-based exchange format, which will allow alignment data to be published and reused independently of the datasets it connects. We plan to translate existing alignment datasets into this format and to make them available as a shared specimen collection; we invite the community to use these resources and to suggest additional datasets for inclusion. Integration with automatic alignment algorithms – particularly Sync Toolbox (Müller et al., 2021) for DTW-based alignment and deep-learning approaches for cross-modal matching – would make the library useful as an end-to-end pipeline rather than a data-model layer alone.

A formal ontology mapping (RDF/JSON-LD serialisation of the alignment data) would bridge the gap to Semantic Web infrastructure and enable federated queries across distributed alignment datasets. Application to non-Western and contemporary repertoires would test the model's generality – particularly the logical domain's current orientation towards Common Western Music Notation – and likely prompt useful extensions. Finally, GUI tooling would lower the barrier for musicologists who do not program.

6 CONCLUSION

Musical data are temporal, yet the landscape of time-related research data remains deeply fragmented. Aligning diverse representations of the same music is central to both MIR and DM, but the resulting alignment data are rarely straightforward to reuse. We have presented TimeToAlign!, a cross-domain model for musical timelines and their alignments that organises time-related data into six typed timelines across three domains, connected by TimelineGroups, AlignmentBundles and explicit MatchClaims. Because alignment structures require only the dimensional metadata of the timelines they connect, they can be shared and reused independently of the original datasets, whether large, proprietary or restricted. Five examples, spanning historical piano rolls, score-based flow control, acousmatic music, multimodal recordings and rock song genesis, demonstrate that the model handles real-world complexity at scales ranging from two images to well over a hundred timelines across all three domains. The accompanying `timetoalign` library implements this model as an open-source Python tool with typed timelines, loaders for common formats, and a planned data-exchange format. The model, library, tutorial notebooks and collection of real-world test data are openly available and actively maintained.

ACKNOWLEDGEMENTS

The work by JH, MK and MN was supported by the Swiss National Science Foundation (SNSF) through the project 'Towards a Unified Model of Musical Form: Bridging Music Theory, Digital Corpus Research, and Computation' (grant no. 10000183; 2024-2028).

The work by PH, SP and GW was supported by the European Research Council (ERC) under the EU's Horizon 2020 programme (grant no. 101019375, 'Whither Music?'); CC acknowledges funding by the Austrian Science Fund (FWF) under grants PAT 8820923 ('Rach3: A Computational Approach to Study Piano Rehearsal') and PIN 1347924 ('AURA: Augmenting musical interaction via EVAs').

The work by MM was supported by the Deutsche Forschungsgemeinschaft (DFG, the German Research Foundation) under grant no. 500643750 (MU 2686/15-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich–Alexander–Universität Erlangen–Nürnberg (FAU) and the Fraunhofer Institute for Integrated Circuits IIS.

The work by AB and DM was supported by KreativInstitut.OWL, a consortium consisting of the OWL University of Applied Sciences and Arts, Detmold University of Music, and Paderborn University, funded by the Ministry of Economic Affairs, Industry, Climate Action and Energy of the State of North Rhine-Westphalia, Germany.

We thank Elona Shatri for producing and providing the OMR ground truth used in Figure 7.

DATA ACCESSIBILITY

Documentation and code for this model are permanently stored and available at https://doi.org/10.5281/zenodo.20056377.

COMPETING INTERESTS

All co-authors who are members of the TISMIR editorial team have recused themselves from any editorial involvement in this article.

AUTHORS' CONTRIBUTIONS

AB, AF, MG, FDM, DM, AP, KP, EP and DW helped shape the project's conceptual framing and overarching research aims. AB, MG, PH, MK, EP and CW curated, annotated and maintained research data. MM, MN, CC and GW secured funding for the work. DM and EP carried out investigative work, including data collection and experimentation. MG, DM, KP and EP contributed to the development of methodology and the design of models. SP prepared visualisations. CC, MG, PH, MM and SP co-authored portions of the original draft. AB, CC, SD, MG, PH, MK, MM, MN, AP, KP, CW and GW provided critical review, commentary and editorial revision. JH initiated, designed and led every stage of the project, from its conception and infrastructure to coordination, implementation and writing.

NOTES

1. Documentation page: https://timetoalign.github.io/
2. https://youtu.be/28p1sXOXkXo
3. https://www.w3.org/TR/owl-time/
4. TiLiA (a compression of 'TimeLine Annotator') is an open-source, cross-platform graphical user interface designed for creating, displaying and interacting with timeline-style annotations over audio and video. Martins and Gotham (2023) provide a provisional report; a more substantial paper is forthcoming.
5. Source code: https://github.com/TimeToAlign/timetoalign
6. https://doi.org/10.5281/zenodo.20056377

AUTHOR AFFILIATIONS

Johannes Hentschel https://orcid.org/0000-0002-1986-9545
Institute for Theory and History, Anton Bruckner University, Linz, Austria

Axel Berndt https://orcid.org/0009-0006-2807-592X
Paderborn University, Germany

Carlos Cancino-Chacón https://orcid.org/0000-0001-5770-7005
Institute of Computational Perception, Johannes Kepler University, Linz, Austria

Simon Dixon https://orcid.org/0000-0002-6098-481X
Centre for Digital Music, Queen Mary University of London, United Kingdom

Anne Foo https://orcid.org/0009-0003-9839-473X
Independent Researcher, United Kingdom

Mark Gotham https://orcid.org/0000-0003-0722-3074
Department of Digital Humanities, King's College London, United Kingdom

Patricia Hu https://orcid.org/0009-0006-8057-3993
Institute of Computational Perception, Johannes Kepler University, Linz, Austria

Maik Köster https://orcid.org/0009-0009-0236-408X
Institute for Theory and History, Anton Bruckner University, Linz, Austria

Felipe D. Martins https://orcid.org/0000-0002-9965-042X
Institute for Theory and History, Anton Bruckner University, Linz, Austria

Davide A. Mauro https://orcid.org/0000-0001-8437-4517
Paderborn University, Germany

Meinard Müller https://orcid.org/0000-0001-6062-7524
International Audio Laboratories Erlangen, Friedrich-Alexander Universität Erlangen-Nürnberg, Germany

Markus Neuwirth https://orcid.org/0000-0003-1990-052X
Institute for Theory and History, Anton Bruckner University, Linz, Austria

Alexander Pacha https://orcid.org/0000-0002-6064-7696
Institute of Visual Computing and Human-Centered Technology, Vienna Technical University, Vienna, Austria

Kevin R. Page https://orcid.org/0000-0002-1668-6540
Oxford e-Research Centre, University of Oxford, United Kingdom

Silvan Peter https://orcid.org/0009-0000-8328-291X
Institute of Computational Perception, Johannes Kepler University, Linz, Austria

Egor Polyakov https://orcid.org/0000-0003-0913-0429
Hochschule für Musik Weimar, Germany

Laurent Pugin https://orcid.org/0000-0002-9525-4331
RISM Digital Center / University of Bern, Switzerland

David M. Weigl https://orcid.org/0000-0003-1502-1528
Department of Music Acoustics – Wiener Klangstil, mdw – University of Music and Performing Arts Vienna, Austria

Christof Weiß https://orcid.org/0000-0003-2143-4679
Center for Artificial Intelligence and Data Science, Julius-Maximilians-Universität Würzburg, Germany

Gerhard Widmer https://orcid.org/0000-0003-3531-1282
Institute of Computational Perception, Johannes Kepler University, Linz, Austria

REFERENCES

Abdallah, S. A., Raimond, Y., and Sandler, M. (2006). An ontology-based approach to information management for music analysis systems. In *Proceedings of the 120th Convention of the Audio Engineering Society (AES)*, Paris, France.

Agrawal, R., Wolff, D., and Dixon, S. (2021). Structure-aware audio-to-score alignment using progressively dilated convolutional neural networks. In *ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Ontario, Canada (pp. 571–575).

Arifi, V., Clausen, M., Kurth, F., and Müller, M. (2003). Automatic synchronization of music data in score-, MIDI- and PCM-format. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Baltimore, Maryland, USA.

Arzt, A., and Widmer, G. (2015). Real-time music tracking using multiple performances as a reference. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain (pp. 357–363).

Babbitt, M. (1965). The use of computers in musicological research. *Perspectives of New Music, 3*(2), 74. 10.2307/832505.

Baggi, D. L., and Haus, G. (Eds.). (2009). Journal of Multimedia, Special Issue: The New Standard IEEE 1599 – Interacting with Music Contents by XML Symbols, Vol. 4.

Baggi, D. L., and Haus, G. M. (2013). The IEEE 1599 standard. In **D. L. Baggi and G. M. Haus** (Eds.), *Music Navigation with Symbols and Layers* (1st ed., pp. 1–20). Wiley.

Ballester, C., Bacot, B., Bigo, L., Borsan, V. N., Couturier, L., Déguernel, K., Dinel, Q., Feisthauer, L., Frieler, K., Gotham, M., Groult, R., Hentschel, J., d'Hooge, A., Le, D.-V.-T., Levé, F., Maccarini, F., Maričić, I., Micchi, G., Müller, M. ⋯ Giraud, M. (2025). Interacting with annotated and synchronized music corpora on the Dezrann web platform. *Transactions of the International Society for Music Information Retrieval*, *8*(1), 121–139. 10.5334/tismir.212.

Benetos, E., Dixon, S., Duan, Z., and Ewert, S. (2019). Automatic music transcription: An overview. *IEEE Signal Processing Magazine*, *36*(1), 20–30. 10.1109/msp.2018.2869928.

Berndt, A. (2021). The music performance markup format and ecosystem. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, Online (pp. 50–57). 10.5281/zenodo.5624429.

Brazier, C., and Widmer, G. (2021). Handling structural mismatches in real-time opera tracking. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Dublin, Ireland (pp. 366–370). 10.23919/EUSIPCO54536.2021.9616109.

Calvo-Zaragoza, J., Hajič, J. Jr., and Pacha, A. (2020). Understanding optical music recognition. *ACM Computing Surveys*, *53*(4), 1–35. 10.1145/3397499.

Cancino-Chacón, C. E., Grachten, M., Goebl, W., and Widmer, G. (2018). Computational models of expressive music performance: A comprehensive and critical review. *Frontiers in Digital Humanities*, *25*(5). 10.3389/fdigh.2018.00025.

Cancino-Chacón, C. E., Peter, S., Hu, P., Karystinaios, E., Henkel, F., Foscarin, F., Varga, N., and Widmer, G. (2023). The ACCompanion: Combining reactivity, robustness, and musical expressivity in an automatic piano accompanist. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Macao, China (pp. 5779–5787). 10.24963/ijcai.2023/641.

Cannam, C., Sandler, M., Jewell, M. O., Rhodes, C., and d'Inverno, M. (2010). Linked data and you: Bringing music research software into the semantic web. *Journal of New Music Research*, *39*(4), 313–325. 10.1080/09298215.2010.522715.

Dannenberg, R. B., and Mukaino, H. (1988). New techniques for enhanced quality of computer accompaniment. In *Proceedings of the International Computer Music Conference (ICMC)*, Cologne, Germany (pp. 243–249).

Dannenberg, R. B., and Raphael, C. (2006). Music score alignment and computer accompaniment. *Communications of the ACM*, Special Issue: Music Information Retrieval, *49*(8), 38–43. 10.1145/1145287.1145311.

Dorfer, M., Arzt, A., and Widmer, G. (2016). Towards score following in sheet music images. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA (pp. 789–795).

Dorfer, M., Hajič, Jr., J., Arzt, A., Frostel, H., and Widmer, G. (2018). Learning audio-sheet music correspondences for cross-modal retrieval and piece identification. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, *1*(1), 22–31. 10.5334/tismir.12.

Dreyfus, L., Lewis, D., and Page, K. (2025). A digital companion for musicological scholarship: The Lohengrin TimeMachine. *Journal of New Music Research*, *53*(3–4), 297–311. 10.1080/09298215.2025.2487100.

Fields, B., Page, K., De Roure, D., and Crawford, T. (2011). The segment ontology: Bridging music-generic and domain-specific. In *2011 IEEE International Conference on Multimedia and Expo*, Barcelona, Spain (pp. 1–6). IEEE.

Foscarin, F. (2020). *The Musical Score: A Challenging Goal for Automatic Music Transcription*. PhD thesis, Centre d'études et de recherche en informatique et communication.

Foscarin, F., Karystinaios, E., Peter, S. D., Cancino-Chacón, C., Grachten, M., and Widmer, G. (2022). The match file format: Encoding alignments between scores and performances. In *Proceedings of the Music Encoding Conference (MEC)*, Halifax, Nova Scotia, Canada (pp. 52–60).

Fremerey, C., Müller, M., Kurth, F., and Clausen, M. (2008). Automatic mapping of scanned sheet music to audio recordings. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Philadelphia, PA, USA (pp. 413–418).

Fremerey, C., Müller, M., and Clausen, M. (2010). Handling repeats and jumps in score-performance synchronization. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)* (pp. 243–248).

Gómez Gutiérrez, E. (2006). *Tonal Description of Music Audio Signals*. PhD thesis. Universitat Pompeu Fabra.

Gotham, M., Bemman, B., and Vatolkin, I. (2025). Towards an 'everything corpus': A framework and guidelines for the curation of more comprehensive multimodal music data. *Transactions of the International Society for Music Information Retrieval*, *8*(1), 70–92. 10.5334/tismir.228.

Gotham, M., Hentschel, J., Couturier, L., Dyke-Aylen, N., Rohrmeier, M., and Giraud, M. (2023a). The 'Measure Map': An inter-operable standard for aligning symbolic music. In *Proceedings of the 10th International Conference on Digital Libraries for Musicology*, Milan, Italy (pp. 91–99). ACM.

Gotham, M., Redbond, M., Bower, B., and Jonas, P. (2023b). The "OpenScore String Quartet" corpus. In *Proceedings of the 10th International Conference on Digital Libraries for Musicology*, Milan, Italy (pp. 49–57). ACM.

Grünbacher, P., and Neuwirth, M. (2024). *Towards feature-based versioning for musicological research*. In *Proceedings of the 18th International Working Conference on Variability Modelling of Software-Intensive Systems,* Bern, Switzerland (pp. 77–82).

Hentschel, J., Rammos, Y., Neuwirth, M., and Rohrmeier, M. (2025). A corpus and a modular infrastructure for the empirical study of (an)notated music. *Scientific Data*, *12*(1), 685. 10.1038/s41597-025-04976-z.

Hu, P., and Widmer, G. (2023). The Batik-plays-Mozart corpus: Linking performance to score to musicological annotations. arXiv preprint arXiv:2309.02399.

Jung, J., Kim, D., Lee, S., Cho, S., Soh, H., Bukey, I., Donahue, C., and Jeong, D. (2025). Unified cross-modal translation of score images, symbolic music, and performance audio. arXiv preprint.

Liem, C. C. S., Müller, M., Eck, D., Tzanetakis, G., and Hanjalic, A. (2011). The need for music information retrieval with user-centered and multimodal strategies. In *Proceedings of the International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies (MIRUM)*, Scottsdale, Arizona, USA (pp. 1–6).

Marchini, M., Ramirez, R., Papiotis, P., and Maestre, E. (2014). The sense of ensemble: A machine learning approach to expressive performance modelling in string quartets. *Journal of New Music Research*, *43*(3), 303–317. 10.1080/09298215.2014.922999.

Martins, F., and Gotham, M. (2023). "TiLiA": A timeline annotator for all. 25th International Conference on Human-Computer Interaction (HCI International 2023), Copenhagen, Denmark. 10.5281/ZENODO.14779020.

Mayor, O., Llimona, Q., Marchini, M., Papiotis, P., and Maestre, E. (2013). repoVizz: A framework for remote storage, browsing, annotation, and exchange of multi-modal data. In *Proceedings of the 21st ACM International Conference on Multimedia*, Barcelona, Spain (pp. 415–416). ACM.

Müller, M. (2021). *Fundamentals of Music Processing: Using Python and Jupyter Notebooks* (2nd ed.). Springer Verlag.

Müller, M., Arzt, A., Balke, S., Dorfer, M., and Widmer, G. (2019). Cross-modal music retrieval and applications: An overview of key methodologies. *IEEE Signal Processing Magazine*, *36*(1), 52–62. 10.1109/msp.2018.2868887.

Müller, M., Özer, Y., Krause, M., Prätzlich, T., and Driedger, J. (2021). Sync toolbox: A Python package for efficient, robust, and accurate music synchronization. *Journal of Open Source Software*, *6*(64), 3434. 10.21105/joss.03434.

Neuwirth, M., Harasim, D., Moss, F. C., and Rohrmeier, M. (2018). The annotated Beethoven corpus (ABC): A dataset of harmonic analyses of all Beethoven string quartets. *Frontiers in Digital Humanities*, *5*, 1–5. 10.3389/fdigh.2018.00016.

Page, K. R., Bechhofer, S., Fazekas, G., Weigl, D. M., and Wilmering, T. (2017). Realising a layered digital library: Exploration and analysis of the live music archive through linked data. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, Toronto, Ontario, Canada (pp. 1–10).

Peter, S. D., Hu, P., and Widmer, G. (2025a). How to infer repeat structures in MIDI performances. In *Proceedings of the Music Encoding Conference (MEC)*, London, United Kingdom.

Peter, S. D., Hu, P., and Widmer, G. (2025b). Pairing real-time piano transcription with symbol-level tracking for precise and robust score following. In *Proceedings of the 22nd Sound and Music Computing Conference (SMC)*, Graz, Austria.

Selfridge-Field, E. (1997). Introduction: Describing musical information. In **E. Selfridge-Field** (Ed.), *Beyond MIDI: The Handbook of Musical Codes* (pp. 3–38). MIT Press.

Shatri, E., and Fazekas, G. (2021). DoReMi: First glance at a universal OMR dataset. arXiv preprint. https://ui.adsabs.harvard.edu/link_gateway/2021arXiv210707786S/doi:10.48550/arXiv.2107.07786.

Shi, Z., Sapp, C., Arul, K., McBride, J., and Smith III, J. O. (2019). SUPRA: Digitizing the Stanford University Piano Roll Archive. In *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR),* Delft, Netherlands (pp. 517–523). 10.5281/zenodo.3527858.

Thomas, V., Fremerey, C., Müller, M., and Clausen, M. (2012). Linking sheet music and audio – Challenges and new approaches. In **M. Müller, M. Goto, and M. Schedl** (Eds.), *Multimodal Music Processing*, volume 3 of *Dagstuhl Follow-Ups* (pp. 1–22). Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 10.4230/DFU.Vol3.11041.1.

Thoresen, L. (2009). Sound-objects, values and characters in Åke Parmerud's Les objets obscurs, 3rd section. *Organised Sound*, *14*(3), 310–320. 10.1017/s1355771809990124.

Thoresen, L. (2010). Form-building patterns and metaphorical meaning. *Organised Sound*, *15*(2), 82–95. 10.1017/s1355771810000075.

Weigele, K. K., and Brecht, K. (Eds.). (2016). *Chorissimo! Blue: Chorbuch für die Schule*. Carus, Stuttgart.

Weigl, D., and Page, K. (2017). A framework for distributed semantic annotation of musical score: Take it to the bridge! In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China.

Weiß, C., and Peeters, G. (2021). Training deep pitch-class representations with a multi-label CTC loss. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online (pp. 754–761). 10.5281/zenodo.5624359.

Wiggins, G., Müllensiefen, D., and Pearce, M. (2010). On the non-existence of music: Why music theory is a figment of the imagination. *Musicae Scientiae*, *5*, 231–255. 10.1177/10298649100140S110.

TO CITE THIS ARTICLE:

Hentschel, J., Berndt, A., Cancino-Chacón, C., Dixon, S., Foo, A., Gotham, M., Hu, P., Köster, M., Martins, F.D., Mauro, D.A., Müller, M., Neuwirth, M., Pacha, A., Page, K.R., Peter, S., Polyakov, E., Pugin, L., Weigl, D.M., Weiß, C., & Widmer, G. (2026). Time to Align! Modelling Musical Timelines for Music Information Retrieval and Digital Musicology. *Transactions of the International Society for Music Information Retrieval, 9*(1), 384–404. DOI: https://doi.org/10.5334/tismir.296

Submitted: 1 July 2025 **Accepted:** 21 April 2026 **Published:** 23 July 2026

Transactions of the International Society for Music Information Retrieval is a peer-reviewed open access journal published by Ubiquity Press.