

‘TiLiA’: A Timeline Annotator for All

*A proposal for the HCI International Conference 2023, Workshop 1:
Interactive Technologies for Analysing and Visualizing Musical Structure*

Felipe Martins and Mark Gotham

‘TiLiA’ (**T**ime**L**ine **A**nnotator) is an open-source, cross-platform graphical user interface designed for creating, displaying, and interacting with timeline-style annotations over audio and video. Developed using the Python programming language with the tkinter library, TiLiA is complemented by an online platform built with Django and React. The repository is available at: <https://github.com/FelipeDefensor/TiLiA>.

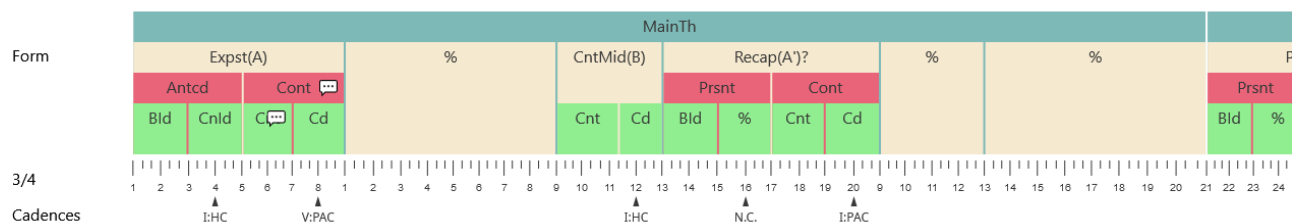


Figure 1: Excerpt of a TiLiA analysis (click to view online). This example includes a *hierarchy* timeline (labeled ‘Form’), a *beat* timeline (‘3/4’), and a *marker* timeline (‘Cadences’).

1 Background

Visual analyses of music have proven to be an enduringly popular, useful, and versatile tool for conveying a wide range of musical ideas.¹ As the famous adage goes, ‘a picture speaks a thousand words’. As these analytical images are primarily connected to the *analysis* rather than the *the music*, they are readily useful not only for the already visual sources of notated music, but also for the much wider range of musical styles and repertoires that are primarily transmitted aurally.

And whatever power a *static* picture has, an *interactive* one has considerably more. Formal analysis is one of the most clear beneficiaries of the visual summary of music. Form can be hard to parse (especially in real-time) but is often easy to display in relatively simple, at-a-glance summaries. Some applications offer basic forms of this from those really designed for score and/or audio editing,² to those specifically designed for the task,³ though only relatively simple annotations are possible, and take up from the community has been limited.

Overall, musical scholarship has not yet made the leap into this technological space with any degree of decisiveness or consistency. Despite some notable exceptions, even most visually minded music analysis is consigned to static images embedded in print (or in e-copies at best) and not subject to the expectations for FAIR, open-source data sharing that is now *de rigueur* in other fields. And even among the more computational communities, there has been little progress on the coordination of standards and corpora for formal analysis.⁴ In response to these challenges (user-friendly design, interoperable formats, ...) we present TiLiA: a timeline annotator for all.

¹See Isaacson (2023) for a survey of this long history.

²The **score editor** Sibelius, for instance, has offered a timeline since c.2019 with bar numbers and rudimentary checkpoints for changes to the time and key signatures, presumably because they are possible indicators of section breaks. Peebles (2013) discusses how to use an **audio editor** (Audacity) in a similar way to the tools discussed here.

³Yorgason (2018)’s ‘Audio Timeliner’ is notable for more capturing the needs of the community in a user-friendly way. And ‘Dezrann’ (Giraud et al. 2018) is a more actively maintained and multi-modal offering.

⁴Notwithstanding initial proofs-of-concepts such as Gotham & Ireland (2019)’s ‘Taking Form’.

2 Specifications

In this digital age, we have come to expect certain basic interactive functionality across all such visual interfaces, such as re-sizing content to fit the window. Musical contexts extend this with additional expectations, such as for real-time synchronisation (following a source). And to win over analytical community of prospective users, any such tool will need to represent that community’s analytical ideas in a natural and germane way, with a minimum of UI friction.

TiLiA supports a wide range of analytical expression by providing flexible options for users engaging at either the level of the GUI or the source code. GUI users benefit from the 3 timeline *types*, designed to support conventional formal annotation (e.g., the form-functional system of Caplin (1998)). Coders benefit from the abstract and readily extensible **Timeline** base-class.

TiLiA currently supports three timeline types: ‘hierarchy’, ‘marker’, and ‘beat’ (see fig.1). **Beat** timelines delimit the total timeline with internal ‘checkpoints’. This may take the form of the incremental measurement involved in measures and beats (as shown on fig.1), though it can also be applied to timestamps and in other, more creative ways. As the name implies, **marker** timelines mark specific, one-off positions on the timeline. Fig.1 shows this as used for marking cadences. **Hierarchy** timelines facilitate annotations in the style of ‘bubble diagrams’, similar to those used in Audio Timeliner and elsewhere. Hierarchy components have properties such as start, end, label, color, and comments, along with parent and children attributes representing hierarchical relationships. Users can easily group, split, and merge existing hierarchies to create new ones and control playback by clicking any such block. Each of these timeline *types* contains one or more *properties* to convey information and *components* specifying the graphical representation,

Conversion to the internal representation (in OO objects) is supported for the tabular and JSON formats associated with a range of existing corpora. TiLiA also supports **metadata** for enriching the analysis with contextual information about the analysts involved, the work under discussion, or indeed any user-defined field. This metadata is stored in plain JSON format, supporting easy interoperability, access, and integration with other linked open data sources.

The **desktop application** is available in versions for Windows, macOS and Linux Ubuntu (click here to view/download) and example analyses can be found at the home page of the same site, e.g., here. The TiLiA desktop application is supported by an **online platform** that lets users store, share, and visualize and even *query* existing analyses. Registration is free and only required when using the upload function. This reflects a goal of TiLiA to support not only the creation and storing of analysis in an digital-age manner, but also to support the still-rare practice of analytical *collaboration* and structured *searches* on the corpora (see fig.2).

Future plans include additional timeline types, notably a ‘range’ timeline for representing non-hierarchical events that can support overlapping and parameters beyond form such as instrumentation. (Click here for a demonstration.) Timelines for displaying the audio wave, various score formats, and the many components of harmonic data are also in development as are enhancements to the conversion routines that will enable full, automated display of more corpora within this interactive framework.

We now invite feedback from prospective users on all the above: existing functionality and future plans alike.

The screenshot shows the TiLiA query interface. At the top, there are two sections: 'File conditions' and 'Component conditions'. Under 'File conditions', there is a dropdown menu for 'title' and an empty input field, with a 'Remove' button and an 'Add condition...' button. Under 'Component conditions', there is a dropdown menu for 'formal_function' with 'Main theme' selected, an empty input field, a 'Remove' button, and an 'Add condition...' button. A red 'Search' button is located to the right of these sections. Below the filters is a table with the following data:

Title †	Composer	Label	Form. func.	Form. type	Start	End	Level	Comments
Piano Sonata n. 10 in C, i	W. A. Mozart	MainTh	Main theme		0:00:01.34	0:00:22.12	5	
Piano Sonata n. 10 in C, i	W. A. Mozart	MainTh	Main theme		0:04:36.28	0:04:57.10	5	
Piano Sonata n. 10 in C, ii	W. A. Mozart	MainTh	Main theme	Small ternary	0:00:02.77	0:02:42.77	4	
Piano Sonata n. 10 in C, ii	W. A. Mozart	MainTh	Main theme		0:04:54.41	0:06:33.00	5	
Piano Sonata n. 10 in C, ii	W. A. Mozart	MainTh	Main theme	Pequeno ternário	0:00:02.50	0:02:42.77	5	

Figure 2: An example of the query functionality. In this case, the query seeks entries containing the string ‘Main theme’ in the ‘formal function’ property of the ‘hierarchy’ timeline.

References

- Caplin, W. E. (1998), *Classical form : a theory of formal functions for the instrumental music of Haydn, Mozart, and Beethoven*, Oxford University Press, New York ; Oxford.
- Giraud, M., Groult, R. & Leguy, E. (2018), Dezrann, a Web Framework to Share Music Analysis, in S. Bhagwati & J. Bresson, eds, 'Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18', Concordia University, Montreal, Canada, pp. 104–110.
- Gotham, M. & Ireland, M. (2019), Taking Form: A Representation Standard, Conversion Code, and Example Corpora for Recording, Visualizing, and Studying Analyses of Musical Form, in A. Flexer, G. Peeters, J. Urbano & A. Volk, eds, 'Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019', pp. 693–699.
URL: <http://archives.ismir.net/ismir2019/paper/000084.pdf>
- Isaacson, E. (2023), *Visualizing Music*, Indiana University Press.
URL: <https://www.barnesandnoble.com/w/visualizing-music-eric-isaacson/1140906228?ean=9780253064738>
- Peebles, C. (2013), 'Using Audacity to Participate in Active Musical Listening', *Engaging Students: Essays in Music Pedagogy* .
URL: <http://flipcamp.org/engagingstudents/peebles.html>
- Yorgason, B. (2018), 'Teaching with Timelines', *Engaging Students* 6.
URL: <http://flipcamp.org/engagingstudents6/essays/yorgason.html>