CosmoNote: A Web-based Citizen Science Tool for Annotating Music Performances

Lawrence Fyfe, Daniel Bedoya, Corentin Guichaoua, Elaine Chew

CosmoNote: A Web-based Citizen Science Tool for Annotating Music Performances

Lawrence Fyfe, Daniel Bedoya, Corentin Guichaoua, Elaine Chew
Centre National de la Recherche Scientifique–UMR9912 / STMS Laboratoire (IRCAM)
{lawrence.fyfe, daniel.bedoya, corentin.guichaoua, elaine.chew}@ircam.fr

ABSTRACT

CosmoNote is a web-based citizen science tool for annotating musical structures, with a focus on structures created by the performer during expressive musical performance. The software interface enables the superimposition of synchronized discrete and continuous information layers which include note data representations, audio features such as loudness and tempo, and score features such as harmonic tension in a visual and audio environment. The tools provide the means for users to signal performance decisions such as segmentation and prominence using boundaries of varying strengths, regions, comments, and note groupings. User-friendly interaction features have been built in to facilitate ease of annotation; these include the ability to zoom in, listen to, and mark up specific segments of music. The data collected will be used to discover the vocabulary of performed music structures and to aid in the understanding of expressive choices and nuances.

1. INTRODUCTION

In the course of performing notated compositions, performers add their own expressive manipulations that are not scripted in the score, including variations in timing, loudness, articulation, and timbre [13]. These performer manipulations convey groupings and prominence, or structures, to listeners [5]. These structures are conceived first in the mind of the performer as they make sense of the piece of music and then, when performed, are transmitted to the minds of listeners. While these structures may be perceptible by listeners, they may, however, be difficult to discern with automated analysis. For example, an accented note that could mark the beginning or the end of a note grouping, might be ambiguous with automated analysis but unambiguous to a human listener.

The COSMOS[1] project was created to study these subtle performed musical structures. One of the goals of COSMOS is to use citizen science to study the structures as they are created in recorded music performances. To achieve this aim, we need a software tool that allowed listeners to annotate perceived structures in performed music. Data collected from citizen scientists will then form the basis for studying how performance shapes or re-shapes perceived musical structures. We will invite a wide range of annotators, ranging from musical novices to professional musicians, to contribute to this citizen science initiative. A video, "Le piano virtuose," introduces this research on musical expressivity[2].

To start, we determined the basic requirements for our project. First, in order to reach as many citizen scientists as possible, the tool should be widely accessible with no software setup. A web tool satisfies this objective. The web tool must be capable of presenting multiple representations of recorded music performances, primarily recorded on reproducing pianos. The tool must be able to play the audio and show the notes played, as well as expressive features extracted from the recordings. For the annotations, we wanted our citizen scientists to be able mark up musical structures such as boundaries, regions, and groups (of notes) on the performance visuals, as well as provide comments on their annotations. The annotations created by our citizen scientists need to be saved in a central location for retrieval and later analysis. The research question that we address in this paper is: how can we design and develop an annotation tool that meets these requirements? To answer this question, we created CosmoNote (cosmonote.ircam.fr), a web-based citizen science annotation tool.

The remainder of the paper is organized as follows: the next section describes related work and approaches to music annotation. In Section 3, we describe the design of CosmoNote, and present technical solutions for meeting the design requirements. Finally, we describe our plans for dissemination and community participation in Section 4.

2. RELATED WORK

Many tools have been developed for creating annotations for audio, in both domains of speech and music. Rather than list all of the various annotation projects, we will describe here only the music annotation tools most relevant to our own work. Annotation applications tend to involve either human or automated annotations, and sometimes a combination of both. We are only looking at human or combination human-computer applications. The projects are divided into non-web-based and web-based projects.

2.1 Non-Web-based

In an early effort at creating an annotation application, Tzanetakis and Cook [16] used their MARSYAS framework

[1] cosmos.cnrs.fr

[2] youtu.be/yXkwusNyte4

Web Audio Conference WAC-2021, July 5–7, 2021, Barcelona, Spain.

to determine whether a computer-assisted human temporal segmentation task benefited from automated segmentation. As part of a pilot user study, they presented users with the application and asked them to mark temporal boundaries based on what they call sound "texture", that is, changes in instrument or speaker, etc. Along similar lines, the CLAM Annotator [1] by Amatriain et al. and developed for the CLAM audio framework, was a combination human-computer application, where users could edit descriptors that were created automatically.

Timeliner [12] by Notess and Swan, was a human-based annotation application that allowed users of a digital music library to create annotations for audio files in the library. Annotations included marking time regions and specific time points of interest. Text labels could be created for each of these annotations. The playback of the audio file could be tied to the annotations. This project had many features that are relevant to the development of our own project, though with the limitation that it was not a web application. In the future work section of their paper, the Timeliner authors commented that the fact that Timeliner was a Java application made it difficult for them to share with their distance learning students who expected everything to run in a browser.

The MUCOSA project by Herrera et al. [8] offered a variety of different annotations and was built on top of the WaveSurfer speech annotation tool. WaveSurfer [15] was an early open-sourced tool for speech annotation that allowed for plugins, a reason why it was chosen for the MUCOSA project. Annotations like structure markings were shown as squares below a spectrogram and a waveform visualization. While the interface showed many information panels of interest, it was crowded since the visualizations and the annotations were vertically stacked on the interface rather than being overlapping layers. For our own project, we chose a layered visualization that allowed for more screen space for the visuals.

Li et al. [9] built an annotation system on top of the Audacity[3] audio editor. For this, they added separate tracks in addition to Audacity's waveform visualization track. The annotation tracks contained region markers and labels for regions. The objective was to establish a set of ground truth data for segmentation of songs, for example, regions for chorus and non-chorus parts of the songs. Like the MUCOSA annotation system, the waveform visualization and the annotations were vertically stacked.

Of all of the non-web tools that we examined, Sonic Visualiser [3] had a set of features that most closely matched our requirements. One interesting feature is that the annotations are layered on top of waveforms, spectrograms, and notes, allowing annotators to more directly place their annotations and saving screen space.

2.2 Web-based

The projects described in the previous section had interesting features but were not web tools. So we also looked specifically at web-based annotation tools.

Cartwright et al. [4] created Audio-annotator[4], a tool built on top of the WaveSurfer.js[5] waveform visualization library. Audio-annotator allowed users to create annotations by selecting a sound region. Annotations could be edited and users could listen to the sounds from the selected regions of their annotations. As part of their CrowdCurio project, they used their tool to ask users to annotate soundscapes of varying complexity based on waveform visualizations, spectrograms, or no visualization at all.

BAT[6], from Melendez-Catalan et al. [10], was another web annotation tool that asked users to annotate radio recordings with the goal of detecting music in radio broadcasts. Annotators were asked to distinguish between music and speech in the recordings by selecting time regions and then identifying them as music or speech. They built their interface using the same version of WaveSurfer.js used for the Audio-annotator. As a result, the two projects were very similar in appearance and functionality, offering region selection based on waveform and spectrogram visualizations.

The projects described above offered some inspiration for our own annotation tool but, in the end, they did not meet our requirements. In particular, all of the projects displayed their sound or music selections as waveforms and/or spectrograms. For description and explaining of performance, we needed to to be able to associate prosodic information with individual notes. Another limitation of the other projects is their annotation types. All of the projects offered region selections and some offered boundaries and/or comments. For our project, we needed more annotations options, including boundary markers , comments, and, because of project is note-based, the ability to select groups of notes. The need for a more customized web tool for annotations led us to develop our own tool.

3. DEVELOPMENT

Having established our requirements and having decided to develop our own annotation tool, we proceeded to the development as described in the following subsections: the preparing of the music data, seeing and hearing the music, annotating the music, and exporting the annotation data.

3.1 Preparing the Music Data

In order to present musical performances for our citizen scientists to annotate, we first needed to get the music performance data. Since we are interested in elucidating the structures that emerge from performances, we needed to record or get recordings of actual performances of various pieces. In particular, we wanted to show the notes for a given piece, according to when and how they are played, along with the audio. So we needed a system that recorded both audio and MIDI for each performance.

To do that, we can use existing player piano files or create bespoke recordings using a reproducing piano capable of recording performances as MIDI data. The example shown in Figure 1 was recorded on a Bösendorfer Enspire PRO piano[7]. During the recording process, the MIDI data is streamed from the piano to a computer while the audio is simultaneously recorded via microphones, ensuring proper alignment.

In the MIDI data, notes are defined by pairs of note-on and note-off events. To facilitate visualization of the notes, we need for each note to be a single event with a start time and an end time. To convert the MIDI data to the desired

[3] audacityteam.org
[4] github.com/CrowdCurio/audio-annotator
[5] wavesurfer-js.org
[6] github.com/BlaiMelendezCatalan/BAT
[7] boesendorfer.com/en/pianos/disklavier-edition

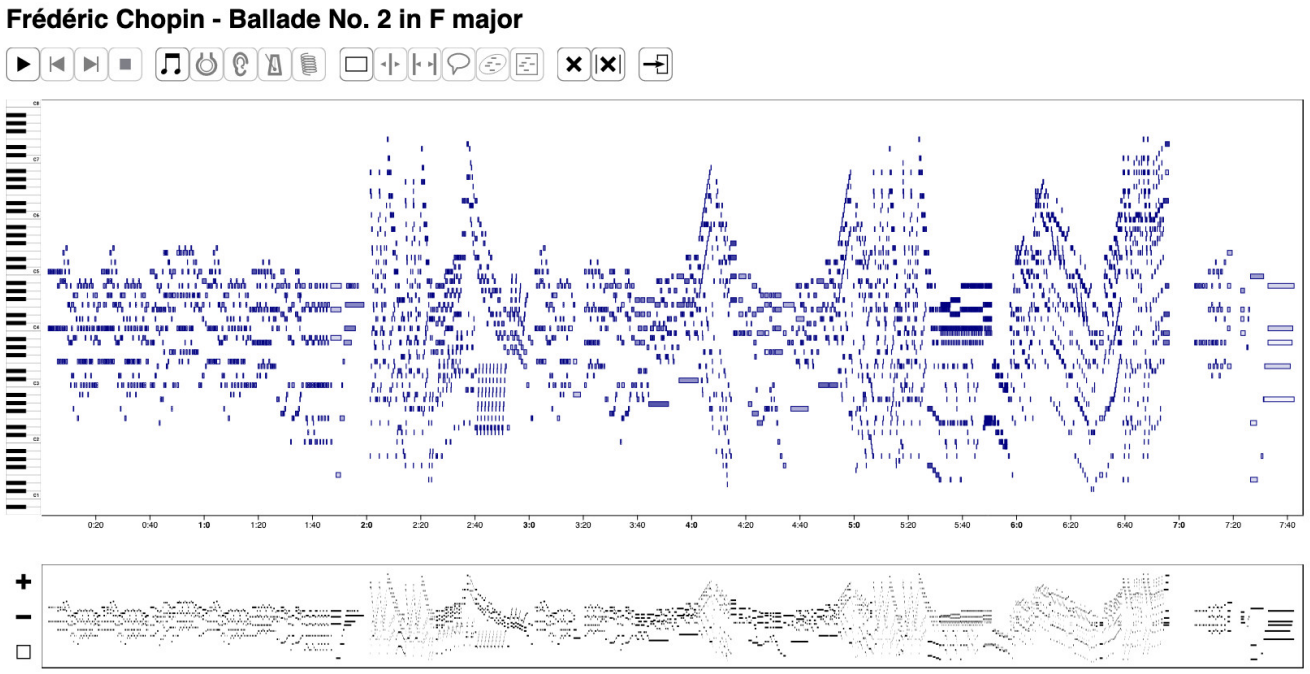


Figure 1: CosmoNote showing the notes of a recorded performance of *Ballade No. 2 in F major* by Frédéric Chopin.

note data, we developed a Python script that converts the MIDI to JSON, a widely supported data exchange format.

For the audio, to decrease download size, the audio file from the performance is encoded from WAV to FLAC. We also tried the Opus [17] audio format since it has a better trade-off between quality and file size, but FLAC worked better across different browsers at the time of development.

Finally, the note data and audio files for various pieces of music are loaded into a CouchDB[8] database. CouchDB was chosen because it works well with web applications by supporting HTTP for requests (thus avoiding database drivers) and using JSON[9] for data storage. MIDI and audio recordings of performed music are stored in the database as part of a collection. For example, a collection could be "Chopin Ballades". This allows us to provide our citizen science annotators with themed collections.

3.2 Seeing and Hearing the Music

With the music data ready on the server, we developed the main interface for CosmoNote as a client-side web application. For getting both note and audio data (as JSON) from the server to the client, we used PouchDB[10].

In order to show note data, we used D3 [2][11], a highly-customizable, SVG-based visualization library for creating graphs using a wide variety of data. For CosmoNote, this meant note values (borrowed from MIDI) on the y-axis and time on the x-axis. Beyond visuals, D3 also handles any type of user interaction that the underlying browser supports, an important consideration for asking our citizen scientists to create annotations.

[8] couchdb.apache.org
[9] tools.ietf.org/html/rfc4627
[10] pouchdb.com
[11] d3js.org

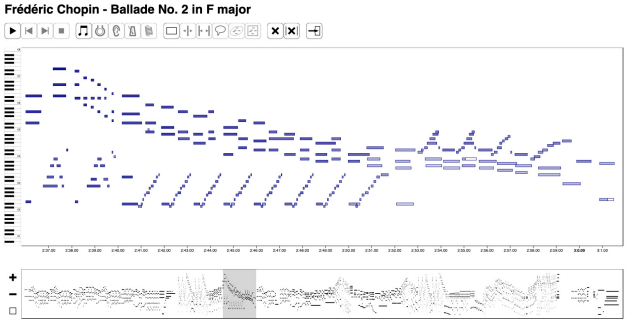
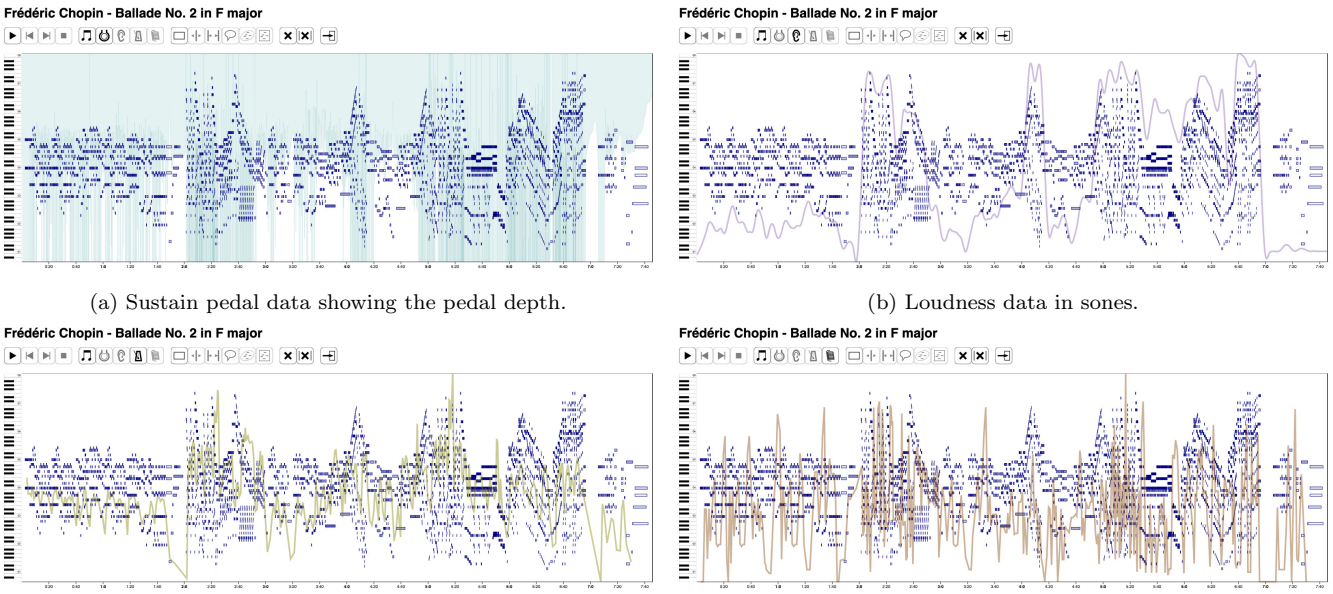


Figure 2: CosmoNote zoomed to show a subset of the notes. The smaller visualization below shows the context of the zoomed notes (inside the grey square) within the larger piece. The controls on the left, from top to bottom, increase the zoom, decrease it, and reset it.

Figure 1 shows the notes for Frédéric Chopin's *Ballade No. 2 in F major* as performed by Elaine Chew. In addition to showing the notes played and their length in time, the velocity of each note is depicted via the note's transparency with darker notes being louder and lighter (more transparent) notes being quieter.

Under the main note visualization panel is a smaller note visualization or context area that shows the same notes but serves a different function. By selecting a time range in the context area, annotators can zoom into a particular area of the main note visualization panel, allowing for a more detailed look at the notes for a part of the performance. Figure 2 shows a selection of notes in the context area while showing the zoomed notes in the main visualization panel.



(a) Sustain pedal data showing the pedal depth.

(b) Loudness data in sones.

(c) Tempo data in beats per minute.

(d) Tension data showing the rate of chord changes.

Figure 3: CosmoNote data visualization layers—sustain pedal, loudness, tempo, and tension—with MIDI note information as backdrop.

When an annotator wishes to listen to the piece, they can hit play at any time. When playback begins, a green vertical line appears over the note visualization as a play head to show progress through the piece. In order to keep the position of the play head synchronized to the audio playback, the play head is animated using `requestAnimationFrame()`[12], which enables animations to run at the frame rate currently used by the browser. When a new animation frame is requested for the play head, a check is made on the amount of time that has passed in the audio file by subtracting the start time of the file from the current time (both obtained via the `AudioContext` object). The time passed is then converted into a position within the main note visualization. Using this technique[13], the play head is always in synch with the audio playback and the time increments for the movement of the play head are small enough to ensure smooth animation.

The audio playback can be paused or stopped at any time during playback. Stopping simply resets the playback position to the beginning of the piece. In addition, during playback, annotators can skip to any part of the audio by clicking an arbitrary time position in the main visualization and the audio will stop and then immediately start playback from that position. Zooming also affects audio playback. When a piece is zoomed in to a particular subset of notes in the main visualization and an annotator hits the play button, only the zoomed notes will be played.

CosmoNote can show more than note data for a given piece. It can also show additional information like sustain pedal movement, the loudness curve, instantaneous tempo changes over time, and harmonic tension. The additional data, shown as information layers over the note layer or even without the note layer, is computed from the audio and MIDI, and synchronized with a MusicXML score for the piece. Figure 3 shows the various information layers with the note layer as backdrop.

The sustain pedal data, taken from the original MIDI, is the teal-colored area curve in Figure 3a. The movement of the sustain pedal is shown on the y-axis with the distance from the top of the graph representing the pedal depression distance. That is, the closer the line is to the x-axis, the more the pedal is depressed. With this orientation, it makes sense to show the data as an area graphic in which the area shows both that the sustain pedal is being used and how much at a glance.

The loudness curve is the purple curve in Figure 3b. Loudness data (in sones) is obtained from the audio file using the MATLAB[14] MA Toolbox [14] as a global representation of the perceived intensity of the notes. It corresponds roughly to velocity data shown for the notes in places although the loudness is influenced by the number of notes played and their pitches, as well as how quickly the key is depressed for individual notes.

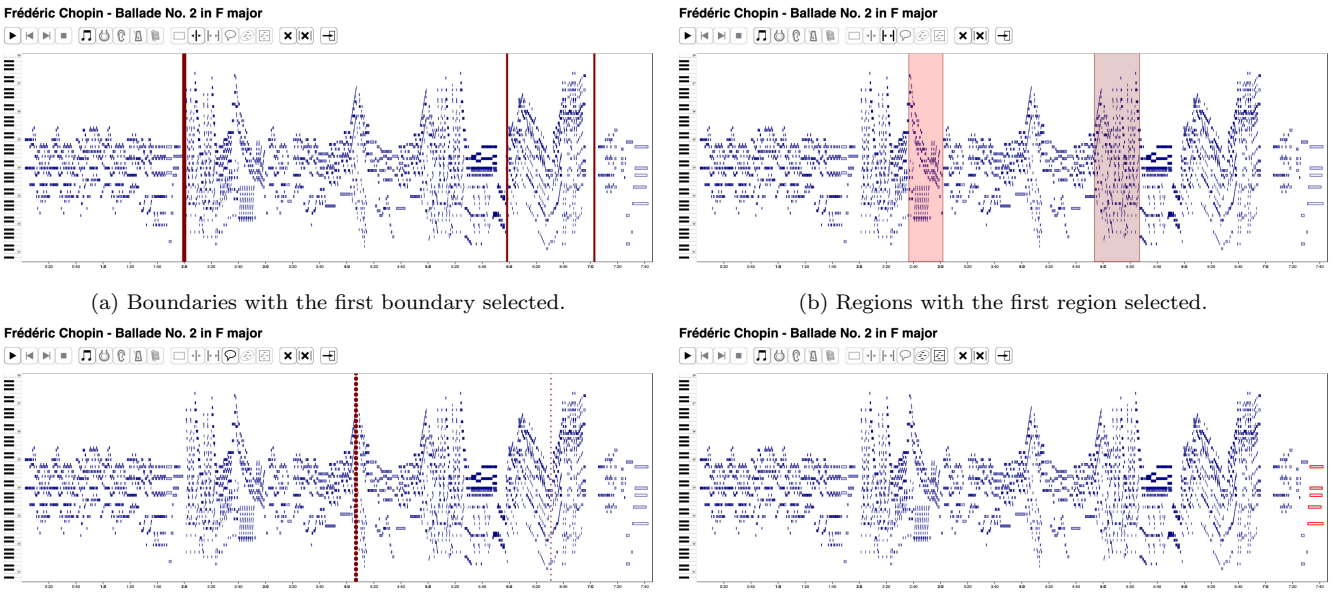
Tempo (in beats per minute) is the green curve in Figure 3c, which corresponds to the rate of music playback. Tempo is computed using timestamps of the onset of each beat—where the beats are located by alignment to the MusicXML score score using Nakamura's alignment tool [11]—throughout a performance and is computed as the inverse of the time between beats. As an example of using tempo data, the parts where the curve shows a steep ascent followed by a descent could help in the identification of tipping points [6]—devices for heightening suspense in expressive performance—in need of structural annotation.

The tension curve is the brown curve in Figure 3d. Tension is computed from the notes of the score using the Xml-

[12]developer.mozilla.org/en-US/docs/Web/API/window/requestAnimationFrame

[13]html5rocks.com/en/tutorials/audio/scheduling

[14]mathworks.com/products/matlab.html



(a) Boundaries with the first boundary selected.

(b) Regions with the first region selected.

(c) Comments with the first comment selected.

(d) A selected group of five notes.

Figure 4: The CosmoNote interface showing four types of annotations: boundaries, regions, comments, and groups, again with note data as backdrop.

TensionVisualiser tool[15] and has three dimensions [7]: cloud diameter representing dissonance, cloud momentum or the rate of chord change, and tensile strain or the distance from the global key. Figure 3d shows cloud momentum though all three dimensions can be visualized.

With all of the data displayed at the same time, the visualization can get difficult to read. To mitigate this problem, each information layer shown in Figure 3 can be turned off individually. Even the notes layer can be turned off to allow focus on one or more of the music feature information layers.

3.3 Annotating the Music

The heart of the CosmoNote is the ability for our citizen scientists to transcribe the structures presented in music performances. CosmoNote features four types of annotations: boundaries, regions, groups, and comments. To make an annotation, annotators select the corresponding button on the toolbar above the visualization panel. Once an annotation type is selected, they may place as many of that kind of annotation as desired. Figure 4 shows examples of the four annotation types. In the ensuing paragraphs, we describe each kind of annotation.

Boundaries, the solid red lines in Figure 4a, allow annotators to delineate edges between structures as indicated by the performer's expressive choices in the music. Boundaries can be placed at any time point, moved, or deleted. The strength of boundaries can be set to weak, medium, or strong, where strong boundaries could denote a pronounced change in musical character and weak boundaries could represent a minor shift in expressive quality projected by the performer. The strength of the boundaries is shown via transparency with opacity increasing with strength. When the audio is playing, an annotator can use the skip buttons to skip the playback to the next or the previous boundary, allowing annotators to hear the results of their boundary placement.

Regions, the translucent red squares in Figure 4b, delineate sections or areas of interest. They can be used to mark transitions or the lead up to a tipping point. In a sense they perform a similar function to boundaries but encompass all of the notes between two boundaries rather than simply denoting the boundaries themselves. Regions can be moved or resized; and, any number of regions can be created.

Comments, the red dotted lines in Figure 4c, enable annotators to mark elements of interest and to write some text about them. When an annotator places a comment at the time point of interest in the main visualization, they can then add the text for their comment below the visualization. Comments provide a means to point at something of interest that is not captured by the other annotation types.

Groups, the five red-lined notes shown at the end of the piece in Figure 4d, provide a way for annotators to select a group of notes of interest, for example, to mark prominence. Any number of notes can be selected to create a group and any number of groups can be created. The process for creating groups is slightly different from the other annotations. It is a multi-step process: 1) the annotator clicks the group button to start a group, then 2) the annotator selects the notes for the group, and 3) the annotator clicks the group finalizing button. The selected notes are then set as a group.

3.4 Exporting the Annotation Data

Once enough annotation data is collected from our citizen science annotators, the data can be exported from the database with a Python script that converts the data from its native JSON format into the comma-separated values (CSV) format[16]. The reason for converting the data to CSV is to enable the data to be used in a variety of software in-

[15] dorienherremans.com/tension

[16] tools.ietf.org/html/rfc4180

cluding MATLAB. The data can then be analyzed to better understand the vocabulary of expressive musical gestures and the choices employed in performance.

4. CONCLUSIONS AND FUTURE WORK

We have completed the major development work on CosmoNote, an online tool for crowd-sourced annotations of recorded performances. In the course of developing CosmoNote, we created the following novel features:

- The display of notes for the music visualization that is synchronized with the audio playback
- The visual display of note velocity via transparency
- The ability to click on an arbitrary time point to start audio playback from that point
- The ability to zoom into a set of notes and to see and hear just the audio for those notes
- The option to view information layers like sustain pedal, loudness, tempo, and harmonic tension
- The ability to create multiple types of annotations including boundaries, regions, groups, and comments
- The ability to skip the audio playback from boundary to boundary

Following a pilot study, CosmoNote will be released to the public, with periodic thematic campaigns addressing different performance collections. As the initiative gathers momentum, we will update the web application with new functionalities to address emerging challenges and user feedback. The discussions that ensue will help the development of a common standard for transcribing expressive elements in performed music, and facilitate the sharing of annotated databases for research and technological development.

5. ACKNOWLEDGMENTS

This result is part of the project COSMOS that has received funding from the ERC under the EU's Horizon 2020 research and innovation program (Grant agreement No. 788960).

6. REFERENCES

[1] X. Amatriain, J. Massaguer, D. Garcia, and I. Mosquera. The CLAM Annotator: A Cross-Platform Audio Descriptors Editing Tool. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, pages 426–429, 2005.

[2] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[3] C. Cannam, C. Landone, and M. Sandler. Sonic Visualiser: An Open Source Application for Viewing, Analysing, and Annotating Music Audio Files. In *Proceedings of the ACM International Conference on Multimedia*, pages 1467–1468, 2010.

[4] M. Cartwright, A. Seals, J. Salamon, A. Williams, S. Mikloska, D. MacConnell, E. Law, J. P. Bello, and O. Nov. Seeing sound: Investigating the effects of visualizations and complexity on crowdsourced audio annotations. *Proceedings of the ACM on Human-Computer Interaction*, 1:1–21, 2017.

[5] E. Chew. From Sound to Structure: Synchronizing Prosodic and Structural Information to Reveal the Thinking Behind Performance Decisions. In *New Thoughts on Piano Performance: Research at the Interface between Science and the Art of Piano Performance*, pages 143–4, 2016.

[6] E. Chew. Playing with the edge: Tipping points and the role of tonality. *Music Perception: An Interdisciplinary Journal*, 33(3):344–366, 2016.

[7] D. Herremans and E. Chew. Tension ribbons: Quantifying and visualising tonal tension. In *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR)*, pages 8–18, 2016.

[8] P. Herrera, Ò. Celma, J. Massaguer, P. Cano, E. Gómez, F. Gouyon, M. Koppenberger, D. García, J. Mahedero, and N. Wack. MUCOSA: A Music Content Semantic Annotator. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, pages 77–83, 2005.

[9] B. Li, J. A. Burgoyne, and I. Fujinaga. Extending Audacity for Audio Annotation. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, pages 379–380, 2006.

[10] B. Meléndez-Catalán, E. Molina, and E. Gómez. BAT: An open-source, web-based audio events annotation tool. In *Proceedings of the International Web Audio Conference*, 2017.

[11] E. Nakamura, K. Yoshii, and H. Katayose. Performance Error Detection and Post-Processing for Fast and Accurate Symbolic Music Alignment. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, pages 347–353, 2017.

[12] M. Notess and M. Swan. Timeliner: Building a Learning Tool into a Digital Music Library. In *EdMedia + Innovate Learning*, pages 603–609. Association for the Advancement of Computing in Education (AACE), 2004.

[13] C. Palmer and S. Hutchins. What is musical prosody? *Psychology of Learning and Motivation*, 46:245–278, 2006.

[14] E. Pampalk. A matlab toolbox to compute music similarity from audio. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2004.

[15] K. Sjölander and J. Beskow. Wavesurfer-an open source speech tool. In *Sixth International Conference on Spoken Language Processing*, 2000.

[16] G. Tzanetakis and P. R. Cook. Experiments in computer-assisted annotation of audio. In *Proceedings of the International Conference on Auditory Display (ICAD)*, pages 111–115, 2000.

[17] J.-M. Valin, G. Maxwell, T. B. Terriberry, and K. Vos. High-Quality, Low-Delay Music Coding in the Opus Codec. In *AES 135th Convention*, 2013.