SymPlot: A Web-Tool to Visualise Symbolic Musical Data

Paula Muñoz-Lago, Ana Llorens, Emilia Parada-Cabaleiro, Álvaro Torrente
Instituto Complutense de Ciencias Musicales (ICCMU)
Universidad Complutense de Madrid, Spain
pmunoz@iccmu.es, allorens@iccmu.es

***Abstract*—Some complex musical parameters might be especially difficult to understand for someone with no theoretical expertise in music. Musicians and music scholars alike normally evaluate such parameters visually by departing from scores, which present the musical events at once. Yet for the understanding of such symbolic representations, musical training is essential, making scores mostly incomprehensible for amateurs. Data visualisation has been applied to meaningfully represent complex musical parameters, thus enabling music amateurs to grasp concepts such as texture or structure. Although scores are one of the "primary" sources to understand music, previous work shows a strong bias towards the visualisation of acoustic data, in detriment of the visualisation of symbolic information. To bridge the gap, we present `SymPlot`, a web-based open source tool to automatically visualise textural density, scoring, and structure from MusicXML files. Due to the multidisciplinary nature of the topic, in this project we have applied the Scrum's agile methodology, an iterative incremental approach specifically tailored for interdisciplinary projects. The tool, aimed at enhancing musical understanding in amateurs and students, as well as in scholars of other disciplines who need to incorporate music into their discourses, i.e., historians, philologists, etc., enables visualisation of local features at various hierarchical levels, highlighting similarities both within and across scores. Our evaluation of `SymPlot`—based on a five-level rating-scale test performed by 50 participants—suggests that colours increase users' understanding of complex musical parameters.**

***Index Terms*—visualisation, music, data science, digital humanities, score, computational musicology, symbolic music data**

I. Introduction

Upon listening, non-trained users are normally able to detect the structure of a given song by noticing easily graspable musical elements, e.g., the lines of the lyrics or the repetitions of musical motives or themes [1]. Nevertheless, the comprehension of complex musical forms may require a level of abstraction that goes beyond pure listening; indeed, expert users normally resort to musical scores as the main source for analysing and understanding musical structure. Music scores widely spread in open-access web-repositories, are however, owing to their complexity, often incomprehensible for amateurs and music lovers. In this context, computational techniques can be applied to enhance musical understanding [2].

Data mining is an integral part of data science [3], its goal being to convert reliable data into meaningful information [4]. Typically, data mining tasks involve data visualisation, which allow users to explore the information graphically and to find trends—the lower the cognitive effort required to understand the graph, the higher its value [5]. In the humanities, linguistic data have been successfully visualised to improve understanding of narrative [6]. Similarly, in the field of Music Information Retrieval (MIR), data visualisation has been thrivingly applied in plotting a variety of musical attributes, e.g., key [7], structure [8], or timbre [9]. Nevertheless, these works show a marked preference for audio data, while symbolic ones have been disregarded almost completely [10]. Furthermore, despite musical parameters' intrinsic interdependence, e.g., that between texture, scoring, and structure [1], [11], the conjunct visualisation of these elements remains unexplored.

In this work we present `SymPlot`,[1] a web-based open source tool aimed at facilitating, by graphical means, the understanding of musical texture, scoring, and structure, as well as the interactions between them. In order to achieve meaningful visualisation of these elements from symbolic data, close collaboration between computer scientists and musicologists was essential. For this reason, this work adopts the Scrum's agile methodology [12], [13], chosen as the most adequate to face such a multidisciplinary project. To disseminate the use of the tool across a wide range of users, we distribute it through a freely accessible user-friendly web-tool (built with Python's[2] `Flask`[3]) that automatically generates a variety of visualisations from MusicXML [14] files. Although our system is primarily oriented towards pedagogical applications, it is also conceived as a navigational tool to support musical listening as well as analytical tasks performed by musicians and scholars.

The rest of the manuscript is structured as follows: Section II outlines the state of the art; Section III defines the relevant musical parameters; Section IV discusses the methodology; Section V exposes the MIR-based visualisation system; Section VI explains the visualisations, Section VII evaluates the web-tool; and Section VIII presents our conclusions.

This paper is part of the Didone Project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, Grant agreement No. 788986

[1]http://symplot.iccmu.es/
[2]https://www.python.org/
[3]https://palletsprojects.com/p/flask/

978-1-7281-9134-8/20/$31.00 ©2020 IEEE
DOI 10.1109/IV51561.2020.00092

II. State of the art

Research in digital humanities has been criticised for its strong bias towards data extraction and processing in detriment of data interpretation and analysis [15]. In an attempt to address this limitation, the automatic representation of time-dependent information through visual means has been considered in a variety of areas in the Humanities [15]. For instance, previous work on dialogue [16] and story-telling [6] has shown how useful data visualisation can be in the study of narratology. Similarly, in the musical domain, related computational techniques have also been employed to explore a variety of musical parameters, such as timbre [9], [17], key [7], musical structure [8], and performative aspects [18], [19], such as musician's expressive strategies [20], which can be visualised with the `Scape-plot` web-tool [4]. Nevertheless, most of these works—and in general most of the research outcomes from computational musicology—show a preference for audio data, while symbolic musical representations have received significantly less attention [10].

Regardless of the source type (audio or symbolic) employed, technical comprehension of music necessarily requires the understanding of a variety of musical features and their interactions. For instance, although texture is an essential aspect of music, the study thereof as a separate element has rarely been deemed as meaningful [21], mainly due to its close interdependence with other musical parameters, such as scoring and structure. Indeed, complex textures, such as accompanied melody [22], are grasped more easily when the various lines are distributed among instruments and/or voices with contrasting timbres than when played in a simple polyphonic instrument, e.g., the piano. This is the case of *Schubert's* song cycle *Winterreise* for voice and piano [23] in comparison with *Mendelssohn's Lieder ohne Worte* for piano [24]. Orchestral timbres have been both classified [25] and visualised [17]. Similarly, even though musical structure might be independent from texture, alterations in the latter commonly influence the understanding of the former [21].

Recent focus on texture, scoring, and structure in the field of music analysis has promoted an increasing interest in the application of computational methods to retrieve them from symbolic data. Despite the fact that Machine Learning (ML) techniques have been profitably applied to detect and classify different types of texture in scores, such as homophonic and polyphonic ones [21], the graphical representation of texture has not yet been performed. Similarly, methods for retrieving and graphically representing musical structure through score-based computational methods have been developed [26]. Due to the intrinsic relationship between scoring and structure, the two parameters have been jointly visualised [27], resulting in freely-accessible web-tools such as `Orcheil`[5]. Yet, in spite of the direct influence of texture and scoring in the emergence of musical structure [21], [22], these three elements have not yet been visually represented in conjunction.

[4] http://www.mazurka.org.uk/software/online/scape/
[5] http://orcheil.ca/

III. Musical Parameters

Musical scores are codified in a bi-dimensional space, with the passing of time being represented along the horizontal axis (the musical events are unidirectionally sorted from left to right), and frequency along the vertical one: within a staff, higher pitches are represented at the top, and lower ones at the bottom; similarly, within each instrumental family (e.g., woodwinds or strings) on an orchestral score, instruments with higher ambitus appear on the top staves. Abstract musical elements to be performed (e.g., character or speed) are encoded within this hyper-plane, along more subjective indications, such as expressive marks (e.g., *dolce* or *allegro*). Textural and timbral density, which respectively emerge from the number of different melodic lines and the number of instruments and/or voices performing these (i.e., the scoring), can be retrieved from scores too. Critically, besides pitch, rhythm, and other essential musical features, textural and timbral density, as well as scoring, can also be relevant in the conformation of musical structure.

A. Textural Density

The concept of textural density in music refers to the number of different musical lines in a given composition [28]; from now on we will refer to these musical lines simply as lines. The number of lines determines whether textural density is dense, moderate or sparse, with the sparsest textural density corresponding to monophonic compositions, i.e., with one line only, and the densest to works with a high number of differentiated lines, such as complex orchestral pieces. A single line may be played/sung by one or more instruments/voices (each of which being a part). Therefore, it is related to scoring. To study this relation, we have also defined the notion of timbral density.

B. Scoring and Timbral Density

We define scoring as the specific combination of a complex of sounding parts (e.g., the instruments of an orchestra) [29], i.e., the specific instruments and/or voices for which a given piece of music is written—each of the instruments and/or voices are termed a part. Scoring, then, relates to the notion of "instrumentation"; yet, besides denoting the selection of instruments for a musical composition [29], scoring takes into account vocal parts too. In a musical work, several instruments or voices may play or sing at the same time, i.e., in a musical composition, there may be a fluctuating number of parts, as determined by the scoring for each specific passage. This determines the timbral density: the higher the number of parts sounding at the same time, the denser the timbral density, and vice versa. In music, timbre is abstractly defined as the tonal quality particular to each instrument or voice. As each of the individual instruments/voices has its own specific timbre [30], this parameter is directly determined by the scoring of the piece.

C. Structure and Form

We can understand musical structure as the organisation of the musical materials in a given piece, resolved into relatively simpler constituent units that appear and sometimes re-emerge throughout the musical composition [30]. Musical structure defines form, which can be seen as the division of a structure in finite sections and their relations, sometimes in the manner of well-known schemes [31]. Some examples of such moulds are the sonata, rondo, and ternary forms [32]. Among the latter, the *da capo* and *dal segno* types stand out, with respectively a complete or an abridged repeat of the opening materials after a contrasting central section.

Research has shown that listeners perceive abstract structural information from acoustic musical features in a bottom-up manner [33] and that changes in texture, and therefore in its density, can also influence the perception of formal boundaries [1]. Therefore, it is easier to grasp the musical structure of a given piece by studying elements that go beyond the pitch and rhythmic conformation of the materials [34]. In other words, both textural and timbral density, in turn influenced by scoring, are factors of structural influence in music.

IV. Methodology

In order to maximise the success of this multidisciplinary project, we followed the Scrum's agile methodology [12], particularly suited to efficiently handle the interaction between different fields, in our case musicology and computer science. The Scrum's agile methodology [13] is an iterative incremental approach in which a refined version of the outcome from the previous iteration is released at every iteration, i.e., at every *sprint*. This methodology serves to guide the efforts towards realistic goals through a progressive improvement of the result at each sprint, which promotes motivation and improves a mutual understanding within multidisciplinary groups. Each of the sprints was articulated in four phases: design (led by the musicologist), development and test (led by the computer scientist), and validation (led by the musicologist). In the present work, a total of four sprints were considered, and a specific goal was collaboratively defined for each of them. In the first sprint, the relationships between textural density and scoring across measures was highlighted —sparse and dense textures were represented with a lower or higher number of divergent graphical lines (from now on we will refer to the graphical lines as stripes); each instrumental/vocal part was represented with a different colour. In the second sprint, the output from the first sprint was simplified by representing the textural density through the graph width (the wider, the higher the number of lines), and the timbral density—relating the scoring—through a grey scale (the darker, the more parts playing/singing the line). Furthermore, in this sprint, structural information was added using alphabetic labels and the indication "Introduction" when pertinent, a typical praxis in music theory. In order to enable comparison across scores, in the third sprint the output from the previous one was re-scaled by modifying the measurement unit from measures to quaver beats. Finally, to give users the possibility of focusing on each specific parameter, in the fourth sprint the parameters previously evaluated were individually visualised on the same criteria as in the second sprint but displayed in separated graphs. Although every sprint took the output of the previous one as the departure point, the goal of each iteration was not necessarily to evolve towards a better solution, but to develop a complementary one; indeed, the output from each sprint is useful on its own.

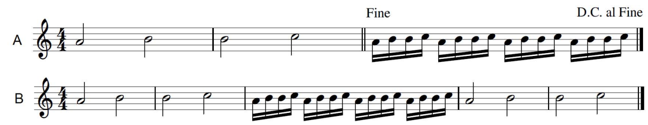

Fig. 1. Example of repeat expansion. Above (staff A), the repetition of the first two measures is indicated trough the repeat mark *D.C. al Fine*; below (staff B), the expanded representation, i.e., without repeat marks, is given.

V. MIR System

Due to the importance of textural density, scoring (as well as the related timbral density), and structure in music analysis, this work aims to facilitate the understanding of these parameters through their visual representation. In every sprint, the following phases were carried out: data extraction, data preprocessing, and information retrieval.

A. Data Extraction

In this phase, the symbolic data, digitally encoded in MusicXML format [14], was processed using the Python's library `music21` [35] in order to extract the selected musical parameters. This stage returned a Python structured object in which all the information of the score was codified. In the subsequent phases of every sprint, the number of lines (i.e., textural density) is determined by evaluating the relationship between time-dependent musical information and the number and type of parts in a score (i.e., scoring), while the structure is retrieved on the basis of the repeat signs.

B. Data Preprocessing

This phase takes as input the Python's object obtained in the previous one and optimises it for its further utilisation. Essentially, since repeat signs (repeat bars and repeat marks) constrict structure visualisation, in this phase those signs were "expanded", i.e., the music that is to be repeated in performance but which is written down only once was rewritten in the corresponding position of the time-line, cf. Figure 1. Even though a repeat expander method has been already implemented in `music21`, i.e., the `musci21.repeat.Expander`,[6] by the time this project was developed, scores with complex nested repeats were not properly handled by the mentioned method. In order to face this limitation, an algorithm to expand repeat bars with several endings (i.e., the bars for first and second endings) and repeat marks with different indications (e.g., *Al Segno* or *D.C al Fine*) was developed.[7]

[6] https://web.mit.edu/music21/doc/moduleReference/moduleRepeat.html#music21.repeat.Expander

[7] https://github.com/DIDONEproject/Repetitions-Expander

C. Information Retrieval

In this phase, to reduce the processing time as much as possible when handling large amounts of data, the computational effort was optimised by choosing the most efficient data structures. A hash table was considered, storing the time slice unit as key (i.e., a bar or a quaver beat), and another hash table as value. This internal key-value pairs contained the notes played during that time slice as keys (each note was represented with a tuple containing its name and duration), and the scoring as value (i.e., the instrument/voice or group of instruments/voices playing/singing each note). In Equation (1) the inner hash table is represented, where: *ni* stands for notes and instruments; *x* for time slice unit; *n* and *d* for note name and duration; and $instr$ for the instrument name.

$$ni_x = \begin{Bmatrix} [(n,d),...] : [instr_1, instr_2, ...] \\ [(n,d),...] : [instr_3, ...] \\ ... \end{Bmatrix} \tag{1}$$

By doing this, each hash table key, i.e., each tuple containing the note's name and duration, was correlated to with a different line, which served to determine the number of lines played/sung at each specific time slice. To "translate" the concept of textural density into a sort of quantitative measurement, to be graphically represented, the number of performed lines was extracted for each time slice. Equation (2) shows the computation needed to obtain such a measurement, where *x* stands for the time slice unit, *ni* refers to the main hash table, and *len* indicates length.

$$textureDens_x = len(ni_x) \tag{2}$$

Timbral density, as determined by the number of instruments/voices sounding simultaneously, is computed by adding the length of the instrument's lists respective to each line for each time slice, previously defined in our hash table. Equation (3) shows the computation needed to obtain the timbral density, where *z* indicates the line, *z.value* refers to the list of instruments playing each line, *ni* stands for the main hash table, *len* indicates length, and *sum* demotes summation.

$$timbralDens_x = sum(len(z.value)\ for\ z\ in\ ni_x) \tag{3}$$

Finally, to determine the musical structure, a script based on the repeat marks previously retrieved (cf. Section V-B), was developed. Basic ternary forms, such as the *da capo* form (ABA), as well as other more complex, such as different *dal segno* designs (e.g., Intro-ABA or AA'BA') can be automatically retrieved with our tool. Nevertheless, the recognition of structural information that does not depend on repeat sings, such as the end of the *development* section in *Sonata forms*, has not yet been implemented.

VI. SymPlot visualisations

Like on musical scores, all the generated graphs visualise the information within a bi-dimensional space, where the time line (indicated as bars or quaver beats) is represented across the horizontal axis, and the texture related information, i.e., performed lines (in the first sprint) and textural density (in the subsequent sprints), is given in the vertical axis. Colours where considered to highlight other musical properties, such as scoring. As indicated in Section IV, `SymPlot` generates four different visualisations, each resulting from one sprint. To generate these, the Python's libraries `Matplotlib`[8] and `Seaborn`[9] were used. All the visualisation examples shown in this section are based on a MusicXML codification of the aria "Son regina e sono amante", from the opera *Didone abbandonata* (music by Domenico Natale Sarro, 1724), the XML of which can be found in GitHub [10].

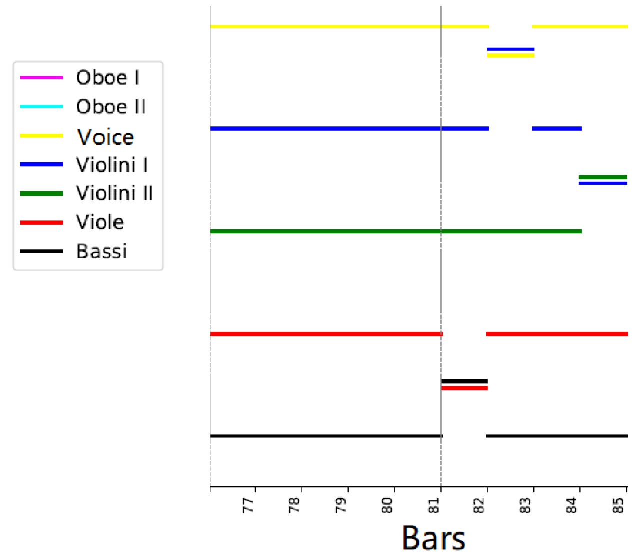

Fig. 2. Snippet of visualisation 1. Across bars (indicated from bar 77 to bar 85), parts are represented with coloured stripes, whereas lines correlate with the stripes' position (a line is shown as joined or separated stripes). Vertical marks across the scoring indicate page breaks in the MusicXML source.

A. Visualisation 1

In this visualisation, changes in textural density and scoring are represented over bars. A different colour is assigned to each part (instrumental or vocal), and every line is indicated by a unique position in the graph's vertical axis. When a given line is performed by several parts, the graph would display the coloured stripes for those parts joined together, showing a convergent disposition (cf. Figure 2, bar 82, Viole and Bassi). On the contrary, when each line is performed by one part, all stripes would be represented separately in their corresponding coordinate, showing a divergent disposition (cf. Figure 2, bars 83–85, Viole and Bassi). Elements aimed to facilitate navigation within the graph, such as time and key signatures, as well as vertical marks regularly indicating the page breaks, were also considered (cf. Figure 2). This visualisation, showing every line in relation to the scoring, promotes an intuitive understanding of orchestration principles, e.g., by highlighting which instruments commonly play the same melody.

[8]https://matplotlib.org/
[9]https://seaborn.pydata.org/
[10]https://github.com/DIDONEproject/SonRegina-Sarro

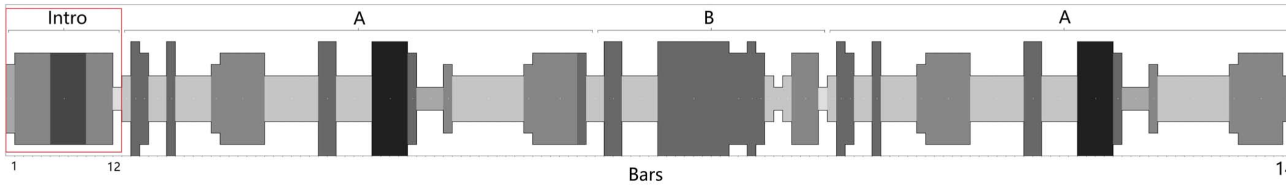

Fig. 3. Visualisation 2. Textural and timbral density (represented through the graph's vertical width and the grey shadowing respectively), and structure (indicated with labels at the top of the graph; the introduction is squared in red) are represented across bars.

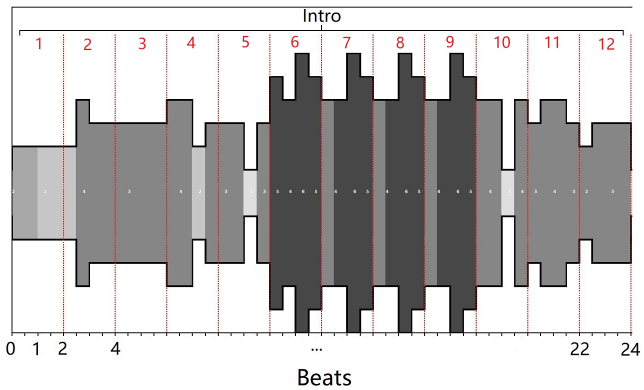

Fig. 4. Snippet of visualisation 3. Textural and timbral density (represented through the graph's vertical width and the grey shadowing respectively), structure (indicated at the top of the graph), and bars (numbers and boundaries marked in red), are represented across quaver beats (displayed until the 24).

B. Visualisation 2

This visualisation represents how the textural and timbral density change over a given piece, indicating the latter's structure too. Textural density is represented by the graph's vertical width (i.e., the higher the number of lines, the wider the graph), while the timbral density is indicated by a grey scale (i.e., the higher the number of parts, the darker the colour). At the top of the graph the score structure is given, indicating each section with a different label—note that only musical forms determined by repeat marks are currently retrieved (cf. Section V-C). Beyond the graph's vertical width, in order not to lose perspective when enlarging the graph in a specific position, the number of lines is also indicated in the middle of the graph.

This graph highlights the relationship between the textural and the timbral densities, connecting them, at the same time, with musical structure. In Figure 3 an example of this visualisation is given, showing that in the introduction section of the composition in question, areas equal in vertical thickness show dissimilar grey tonalities, meaning that they have the same number of lines but different number of parts. This visualisation can be particularly revealing when having, for instance, just one line (i.e., a small height width) but a dark grey, meaning that many parts are playing a single line at unison. Furthermore, when analysing more than one score at a time (i.e., by selecting several MusicXML files in the web-tool), in order to encourage the detection of commonalities, a pdf file containing all the graphs consecutively is created. To enable fair comparison between them, the grey scale employed in the multi-graph visualisations was standardised, meaning that each graph's grey scale was normalised, process carried out by determining the maximum timbral density across the evaluated scores and assigning the darkest shadowing to it, i.e., black, while the remaining values were computed according to this maximum.

C. Visualisation 3

This visualisation, represents, like visualisation 2, musical structure and textural and timbral density, yet in this case across quaver beats instead of bars. Again, textural and timbral density are represented through the graph's vertical width and grey shadowing respectively, and a normalised grey scale was also considered for the multi-graph visualisations. The goal of this visualisation is to enable fair comparison between scores in different metres alike on the basis of a common measurement unit (i.e., a quaver beat). To avoid discordances in the graphs' lengths, the graph length is determined by the number of beats (which depends on the number of bars and the time signature)—note that in the visualisation 2 the graph length is determined by the number of bars (without considering the time signature). In order to respect scores' graphical subdivisions, a guideline indicating bar boundaries is also displayed along with their numbering (cf. red dotted line in Figure 4).

D. Visualisation 4

Finally, in the last visualisation, graphs "complementary" to visualisation 2 (Figure 3) are given to individually represent certain features. This enables the user to focus on the specific musical parameters that were previously shown in combination, i.e., textural density (graph's vertical width), timbral density (grey shadowing), and structure (labels at the top of the graph). Two complementary graphs were considered: Visualisation 4–A, which shows changes in textural density and structure across bars (cf. Figure 5); Visualisation 4–B, which represents through a colour map concise information concerning timbral density, similarly across bars (cf. Figure 6). These graphs enable easier and simplified understanding of the previous visualisations.

VII. SymPlot Usage & Evaluation

In order to allow users to graphically represent scores, we have created a user-friendly web-tool, in which they can choose among a variety of scores and the described visualisations (cf. Section VI) are automatically generated in pdf format, after the execution in the server-side of the processes

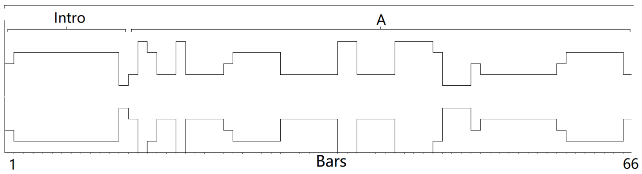

Fig. 5. Snippet of visualisation 4–A. Textural density and structure are indicated, across bars (until bar 66), through the graph's vertical width and with the various labels at the top of the graph, respectively.

discussed in Section V. The web-tool, built with Python's `Flask`, is hosted on `Heroku`,[11] and uses the Google Drive API[12] under `PyDrive` as file storage service.[13] The web page consists of a simple design with a file selector and a short description of `SymPlot`'s purposes. After choosing the files to work with (up to four, to avoid the server overload), the system is started. During the running time, status signs are shown next to the file names, indicating the process' development. At the starting point, a loading icon is shown; nevertheless, as the system is being executed, a ✓(success) or $\times$ (error) will be displayed. Once the execution is completed, a download button, leading to a Google Drive folder with the results, will be available. If the user chooses more than one file, multi-graph visualisations will be also automatically generated along with the individual visualisations.

To evaluate the efficiency of the visualisations presented in this paper, users' perceived improvement on their understanding of the considered musical concepts, i.e., textural density, timbral density, and structure, was assessed. A total of 50 participants (25 with basic or no previous musical knowledge and 25 expert musicians), evaluated each visualisation through a five-level rating scale: 1 stands for "the visualisation does not helps at all" and 5 for "the visualisation is very useful". Users' responses showed no significant differences among visualisations 2 to 4, indicating that these are similarly understood by the participants regardless of their musical expertise: Welch's t-test yielded for the three comparisons $p \geq 0.37$; $t \leq -0.89$; and mean score values between 3 and 4 ($3.12 \leq mean \leq 3.80$). In contrast, valorisation of visualisation 1 diverged among non-expert and expert users, being rated significantly higher by the latter: Welch's t-test yielded $p = 0.02$; $t = -2.40$; and $mean = 3.92$ for non-experts, $mean = 4.44$ for experts. Negative t for all the comparisons indicates that the non-expert group perceived the visualisations less understandable than the expert group. Yet, this difference was minimal for the visualisation 2 to 4: *mean difference* of 0.24, 0.28, and 0.20, for visualisations 2, 3, and 4, respectively; slightly higher for visualisation 1: *mean difference* of 0.52. This is due to the fact that visualisation 1 is specially clear for expert users (it is the only one with an average rating higher than 4), but also the best understood by non experts, which may indicate that colour is a useful variable to increase visualisation comprehensibility.

[11] https://www.heroku.com/
[12] https://developers.google.com/drive
[13] https://pythonhosted.org/PyDrive/

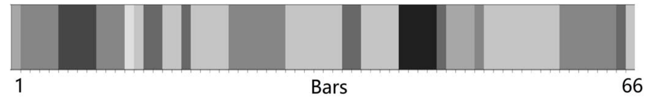

Fig. 6. Snippet of visualisation 4–B. Timbral density is represented, across bars (until bar 66), through a colour map. Darker colours indicate a higher number of parts, whereas lighter tonalities stand for fewer parts.

VIII. Conclusions and Future Work

With the development and test of `SymPlot` we showed that visualising a musical score by mapping time and other musical-related dimensions, e.g., scoring, into a graphical space, is greatly effective for the comprehension of specific musical aspects, such as textural density, timbral density, and structure. Distributed as a user-friendly web-based tool, `SymPlot` aims to improve the overall understanding of a musical piece, for instance by supporting non-trained users in keeping track of a song while it is being played. The presented visualisations enable users to anticipate the events while being aware of its general structure, encouraging thus a listening experience of higher quality. Besides amateurs and music lovers, also technicians, professional musicians, and music teachers could benefit from `SymPlot` in their activities, as the tool offers an intuitive way to understand complex musical concepts such as structure, texture, and timbre.

One of the main limitations of `SymPlot` is that only musical structures defined by repeat marks can be currently retrieved, while those implicitly indicated by other musical parameters, such as harmonic modulations or melodic variations, are not yet automatically identified. In this regard, one of our main priorities in the near future will be to develop a tool capable of detecting both theoretically-codified implicit formal moulds too, such as sonata or rondo forms, and hierarchical structures, i.e., each section's inner structure within bigger forms. We also plan to integrate performance duration of scores according to tempo indications. In this manner, two scores with equal time signature and number of bars but with different tempo markings would result into graphs of different length; the "performance" duration would be indicated too. In other words, the integration of tempo marks as factor for analysis will generate more precise and less abstract visualisations. Finally, since the use of colours has shown to encourage users' comprehension, we also plan to use them, instead of the grey scale, to highlight composers' resorting to specific instrumental families. More specifically, primary colours will be assigned to the different families in order to be able to produce clearly graspable colour mixing. In this manner, parts from different families playing/singing at the same time would be represented through secondary colours, as determined by the combination of the primary colours of their respective families, instead of through darker shadowing. The maintenance of the Web Tool as well as its migration to a computationally more powerful server will be carried out. In this manner we will overcome the present limitation on the maximum number of files to be submitted to Symplot.

REFERENCES

[1] R. B. Dannenberg and M. Goto, "Music structure analysis from acoustic signals," in *Handbook of Signal Processing in Acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds. New York, NY, USA: Springer New York, 2008, pp. 305–331.

[2] T. Crawford, L. Gibson *et al.*, *Modern Methods for Musicology: Prospects, Proposals, and Realities (Digital Research in the Arts and Humanities)*. Farnham, UK: Ashgate, 2009.

[3] W. van der Aalst, *Process Mining: Data Science in Action*. Berlin, Germany: Springer, 2016.

[4] T. Li, M. Ogihara, and G. Tzanetakis, *Music Data Mining*. Boca Ratón, FL, USA: CRC Press, 2011.

[5] N. Cao and W. Cui, *Introduction to Text Visualization*. Paris, France: Atlantis Publishing Corporation, 2016.

[6] E. Segel and J. Heer, "Narrative visualization: Telling stories with Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1139–1148, 2010.

[7] E. Gómez and J. Bonada, "Tonality Visualization of Polyphonic audio," in *Proceedings of the International Computer Music Conference*. Barcelona, Spain: ICMC, 2005, pp. 57–60.

[8] H.-H. Wu and J. P. Bello, "Audio-based music visualization for music structure analysis," in *Proceedings of the Sound and Music Computing Conference*. Barcelona, Spain: SMC, 2010, pp. 1–6.

[9] S. Soraghan, A. Renaud, and B. Supper, "A Perceptually Motivated Visualisation Paradigm for Musical Timbre," in *Proceedings of the Conference on Electronic Visualisation and the Arts*. London, UK: BCS, 2016, pp. 53–60.

[10] C. Mckay and I. Fujinaga, "JSymbolic: A feature extractor for MIDI files," in *Proceedings of International Computer Music Conference*. New Orleans, LA, USA: ICMC, 2006, pp. 302–305.

[11] J. Li, "Music Analysis Through Visualization," in *Proceedings of the International Conference on Technologies for Music Notation and Representation*. Cambridge, UK: TENOR, 2016, pp. 220–225.

[12] M. Fowler, J. Highsmith *et al.*, "The agile manifesto," *Software Development*, vol. 9, pp. 28–35, 2001.

[13] R. S. Kenneth, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Upper Saddle River, NJ, USA: Addison-Wesley, 2012.

[14] M. Good, "MusicXML: An Internet-friendly Format for Sheet Music," in *XML Conference and Expo*, Santa Clara, CA, USA, 2001, pp. 3–4.

[15] E. Gardiner and R. G. Musto, *The Digital Humanities: A Primer for Students and Scholars*. New York, NY, USA: Cambridge University Press, 2015.

[16] A. Tat, M. Carpendale, and T. Sheelagh, "Visualising Human Dialog," in *Proceedings of the International Conference on Information Visualisation*. London, UK: IEEE, 2002, pp. 16–21.

[17] K. Siedenburg, "An Exploration of Real-time Visualizations of Musical Timbre," in *Proceedings of the International Workshop on Learning Semantics of Audio Signals*. Graz, Austria: LSAS, 2009, pp. 17–30.

[18] J. Langner and W. Goebl, "Visualizing Expressive Performance in Tempo—Loudness Space," *Computer Music Journal*, vol. 27, no. 4, pp. 69–83, 2003.

[19] D. Weigl and W. Goebl, "Rehearsal Encodings With a Social Life," in *Proceedings of the Music Encoding Conference*. Boston, MA, USA: MEC, 2020.

[20] R. Hiraga and N. Matsuda, "Visualization of music performance as an aid to listener's comprehension," in *Proceedings of the Working Conference on Advanced Visual Interfaces*. Gallipoli, Italy: AVI, 2004, pp. 103–106.

[21] M. Giraud, F. Levé *et al.*, "Towards Modeling Texture in Symbolic Data," in *Proceedings of the International Society for Music Information Retrieval Conference*. Taipei, Taiwan: ISMIR, 2014, pp. 59–64.

[22] W. Piston, *Orchestration*. New York, NY, USA: W.W. Norton, 1955.

[23] F. Schubert, *12 Lieder aus "Wintereise"*, T. Hoppstock and W. Müller, Eds. Darmstadt, Germany: Prim Musikverlag, 2006.

[24] F. Mendelssohn-Bartholdy, *Lieder ohne Worte*, H. Germer, Ed. Leipzig, Germany: Bosworth, 1899.

[25] A. Antoine and E. R. Miranda, "Musical Acoustics, Timbre, and Computer–Aided Orchestration Challenges," in *Proceedings of the International Symposium on Musical Acoustics*. Montreal, Canada: ISMA, 2017, pp. 151–154.

[26] M. Wattenberg, "Arc diagrams: visualizing structure in strings," in *Symposium on Information Visualization*. Boston, MA, USA: IEEE, 2002, pp. 110–116.

[27] W. Chan, H. Qu, and W. Mak, "Visualizing the Semantic Structure in Classical Music Works," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 1, pp. 161–173, 2010.

[28] G. Fankhauser and W. Harbison, *Beyond Harmony: An Introduction to Analysis of Form and Structure in Music*. Dubuque, IA, USA: Kendall Hunt, 2016.

[29] K. Kreitner, M. Térey-Smith *et al.*, "Instrumentation and orchestration," *Grove Music Online*, accessed on March 18 2020. [Online]. Available: http://www.oxfordmusicOnline.com

[30] I. Bent and A. Pople, "Analysis," *Grove Music Online*, accessed on March 27 2020. [Online]. Available: http://www.oxfordmusicOnline.com

[31] A. Whittall, "Form," *Grove Music Online*, accessed on March 3 2020. [Online]. Available: http://www.oxfordmusicOnline.com

[32] C. Rosen, *Sonata Forms*. New York, NY, USA: W.W. Norton, 1988.

[33] P. E. Keller, "Rhythm and Time in Music Epitomize the Temporal Dynamics of Human Communicative Behavior: The Broad Implications of London's Trinity," *Empirical Musicology Review*, pp. 17–27, 2012.

[34] M. E. Bonds, "The Spatial Representation of Musical Form," *The Journal of Musicology*, vol. 27, no. 3, pp. 265–303, 2010.

[35] M. S. Cuthbert and C. Ariza, "music21: A toolkit for Computer-aided Musicology and Symbolic Music Data," in *Proceedings of the International Society for Music Information Retrieval Conference*. Utrecht, Netherlands: ISMIR, 2010, pp. 637–642.