Measuring the structural complexity of music: from structural segmentations to the automatic evaluation of models for music generation

Jacopo de Berardinis, Angelo Cangelosi *Member, IEEE,* and Eduardo Coutinho

***Abstract*—Composing musical ideas longer than motifs or figures is still rare in music generated by machine learning methods, a problem that is commonly referred to as the lack of long-term structure in the generated sequences. In addition, the evaluation of the structural complexity of artificial compositions is still a manual task, requiring expert knowledge, time and involving subjectivity which is inherent in the perception of musical structure. Based on recent advancements in music structure analysis, we automate the evaluation process by introducing a collection of metrics that can objectively describe structural properties of the music signal. This is done by segmenting music hierarchically, and computing our metrics on the resulting hierarchies to characterise the decomposition process of music into its structural components. We tested our method on a dataset collecting music with different degrees of structural complexity, from random and computer-generated pieces to real compositions of different genres and formats. Results indicate that our method can discriminate between these classes of complexity and identify further non-trivial subdivisions according to their structural properties. Our work contributes a simple yet effective framework for the evaluation of music generation models in regard to their ability to create structurally meaningful compositions.**

***Index Terms*—Music structure analysis, Evaluation measures**

I. Introduction

Music is a powerful medium that conveys meaning to listeners by combining a variety of musical elements synchronously and sequentially. At the perceptual level, the basic attributes involved in music perception are loudness, pitch, contour, rhythm, tempo, timbre, spatial location and reverberation [1]. Whilst listening to music, our brains continuously track and analyse these signals according to diverse gestalt and psychological schemas. Some of them entail higher order musical dimensions (e.g., metre, key, melody, harmony), which reflect (contextual) hierarchies, intervals and regularities between the different musical elements. Others involve continuous predictions about what will come next in the music as means of tracking structure and conveying meaning [2].

Structural elements of music can range from local/short-term organisational levels (e.g., chord, a sequence of notes/sounds) – the "micro" level – to the longer temporal scales capturing the form of a composition or compositions (e.g., sonata form in classical music, or verse/chorus form in popular music) – the "macro" structure. Within these levels, patterns can be identified and music can be segmented in various ways on the basis of specific musical characteristics at different temporal levels (e.g., dynamics, patterns of durations/rhythm, melodic patterns, instrumentation, etc.) [3].

Given that the same musical material may induce structure at different interrelated levels, one interesting feature of musical organisation is its hierarchical nature. For instance, a piece of music may be analysed in terms of its overall form (e.g., divided into meaningful sections), but within those sections we can further divide music into sub-levels that reflect, for instance, rhythmic or harmonic structure. Naturally, given the diversity of musical styles and compositional/performative approaches, different pieces/performances will have different kinds and amounts of (hierarchical) structure, and therefore diverge in terms of structural complexity[1].

The varied and sophisticated patterns of structure that characterise music are a key distinguishing factor when compared to other acoustic mediums (e.g., speech, soundscapes). In fact, the importance of music structure to musical appreciation is paramount [5] and a wide range of musical parameters as well as structural features are fundamental to covey different types of meaning to listeners [6], which in turn can trigger a cascade of other responses (e.g., dancing, emotions) [7].

In the last few years, composing music with machine learning systems has attracted great interest from academia and industry [8]. Companies started offering automatic music composition solutions for entertainment content, such as soundtracks for video games and commercials. Researchers, instead, are leaning towards *computer-assisted composition*, augmenting the creative potential of artists and composers [9]; and *machine improvisation*, a category of intelligent systems capable to temporarily replace a performer during a live session [10]. Improving the generative capabilities of these systems does not only opens up the investigation of new forms of music, but is also considered a pinnacle to understand machine creativity [11].

Nonetheless, dealing with the structural complexity of music has been a tremendous challenge, especially for generating long and musically meaningful pieces endowed with form and long-term structure (e.g. sections) [12, 13]. Indeed, current state-of-the-art systems generate pieces that are mostly characterised by local or short-term form, with motives – the

A. Cangelosi is with the Machine Learning and Robotics Group (MLR), Department of Computer Science, University of Manchester, M13 9PL, UK.

E. Coutinho is with the Applied Music Research Lab (AMLAB), Department of Music, University of Liverpool, L69 7WW, UK.

J. de Berardinis is with King's College London and the University of Liverpool; e-mail: jacopo.deberardinis@kcl.ac.uk.

[1]For a detailed perspective on the theoretical analysis of music structure, we refer to [4], and to [3] or [2] for a computational treatment of the subject.

shortest musical ideas, dominating the synthetic compositions [14]. This is particularly prominent in music generated from long-short term memory (LSTM) recurrent neural networks (RNNs) [15], and is linked to the well-known problem of learning long-term dependencies from sequential data [16] – a long-standing goal in machine learning research.

The advent of self-attention networks (SAN) in music modelling ameliorated this problem [17], with Transformer models now capable of generating music possessing structural properties that remain more coherent across a larger temporal scale compared to LSTMs [18]. Nevertheless, there is general consensus on the fact that the automatic generation of music with a realistic level of structural complexity is still an open problem for most genres. In fact, although structures at different temporal scales can now be found in generated music, those are rarely organised to convey a coherent musical idea throughout the piece, and inter-related with each other based on the principles of *repetition*, *variation* and *contrast* [19].

Compared to the symbolic domain, the problem of structural complexity is more challenging for *audio waveform generation*, as it requires processing significantly longer sequences (a three-minute-long audio segment sampled at 44.1 kHz will have an input length of about 8 million time steps). Not only does this exacerbate the problem of learning long-term dependencies, but generative models of waveform also have to capture an additional wide range of musical properties (e.g. timbre). In the audio domain, autoregressive models have been demonstrated to model local signal variations effectively and capture temporal correlations across tens of seconds [20]. A recent state-of-the-art automatic composition system is JukeBox [21] – generating audio music conditioned on artist, style, and lyrics. This model, counting billions of (learnable) parameters, was trained for several weeks using more than 512 V100 GPUs. Nevertheless, when describing the generated musical repertoire, the authors reported that they could not "*hear long term musical patterns, and [...] choruses or melodies that repeat*" [21]. Analogously, when using JukeBox to generate completions close to the original pieces, they found that the generated continuations would "*deviate completely into new musical material after about 30 seconds*".

In our view, to start tackling this problem it is necessary to evaluate the structural complexity of music generated by automatic systems, in order to have a reference point that can be used to improve their composition capabilities. However, the evaluation of music generation methods is another open issue in the field, considering the lack of a standard evaluation methodology that can enable and foster a fair and objective comparison of music generation systems on a large scale [22].

Even though computational methods quantifying specific musical properties have been previously addressed, there is still an open gap in devising measures of structural complexity that can easily be reused for the evaluation of generated music. To the best of our knowledge, current works focus on measuring tonal [23, 24], harmonic [25] and rhythmic [26] complexity of music, as well as properties related to musicality [27] and individuality [28] of performances. Notably, the work by Streich [29] encompasses both tonal, rhythmic and timbral complexities – which are considered independently as musical facets, and argues that the exploration of human-perceived complexity should not be limited to pure information-theoretical approaches, such as entropy measures and Kolmogorov complexity. Nonetheless, the closest measure of complexity entailing structural properties of the music signal is the structural change [30] – a vector-valued meta-feature that can be computed from any arbitrary frame-wise audio feature (e.g. a chromagram) to quantify its amount of change at different temporal scales. Each vector element is expected to capture the structural change of a given feature at a certain temporal scale, thus resembling Foote's convolution with a checkerboard kernel [31], where the window size of the time scale parameterises the kernel. Although the convolution method yields a novelty curve that can be used for structural segmentation, it is not yet clear how the meta-feature would relate to the presence of music structures rather than arbitrary structures. In addition, the detection and the identification of music structures generally requires taking multiple features into account rather than relying on a single descriptor [32].

A. Our contributions

In this article, and building upon our previous work [33], we introduce a new set of metrics that tries to address a specific gap – the automatic evaluation of music structural complexity. Our method leverages a state of the art computational method for music structure analysis (MSA) to detect structures and their nested organisation within a composition. The resulting structural segmentation is then analysed and summarised with a set of metrics we devised to formally describe the decomposition process of the identified musical ideas. In lieu of subjectively defining structural complexity, our approach is based on the hypothesis that the former is a latent property that can be captured by a set of metrics. Nonetheless, given the scope of this work, when addressing music structural complexity we are primarily looking at the presence and richness of music structures at different temporal scales, rather than seeking a more general information theoretic interpretation of structural complexity, thereby aligning with Streich's views [29].

We tested this method on a large dataset comprising music with different types of structural complexity, and found that our metrics can explain structural properties inherent to each complexity class. We also showed and provided examples on how these metrics can be used for evaluating the structural complexity of music. Although our method is defined on audio music, the obtained results demonstrated that our metrics also work on synthesised MIDI music – thereby addressing both the audio and the symbolic domains. The main contributions of this paper are a set of metrics quantifying structural properties of music, together with a novel evaluation framework for the automatic analysis of structural complexity from music.

II. Evaluation of automatic composition methods

Evaluation is always required when submitting a novel music generation method. Nonetheless, different evaluation criteria and strategies are used heterogeneously and in isolation from each other. In most cases, evaluation relies on manual and subjective judgements by human listeners whom provide

Fig. 1. Illustration of the hierarchical segmentation of a piece sampled from the distribution of an untrained LSTM (*left*); one generated from an LSTM network trained on a dataset of symbolic classical music (*centre*); and the other chosen from a collection of classical compositions for piano (*right*), all with the same duration. For each plot, the innermost circle corresponds to the first level in the hierarchy, where all audio frames belong to the same segment enclosing the whole piece. From the second level, segments start decomposing into finer structural components (colours denote their identity, although repetitions occur due to their limited availability), until every frame forms a community per se at the bottom of the hierarchy (the outermost circle).

ratings on specific properties related to the music composition itself (e.g. pitch range, mode, rhythmical consistency) or their subjective evaluation of the listening experience (e.g., likeability, originality). In some (rare) cases, expert listeners are asked to evaluate the generated pieces by analysing their musical properties as a music teacher would do with the composition of a student [34].

Overall, in line with the taxonomies reported in [35] and [22], evaluation methods for music generation can be organised into the following categories – rarely used in conjunction.

Music modelling evaluation. It concerns the evaluation of the prediction performance of an autoregressive music model – a specific family of music generation systems (also known as *predictive models for music*) that are trained to predict the next musical token (e.g. a note, chord, or a quantised representation of musical material) given the context of the previous events in a musical sequence (analogously to language models). This type of evaluation is based on the assumption that a model that can effectively predict music – having learnt associations between past and future musical content, can potentially encapsulate notions of music perception and composition. Hence, evaluating the predictive capabilities of a music model provides an indicator of the learned musical features possibly reflecting theoretical properties of music. The most common quantitative evaluation measures in the literature are the *log-likelihood* of the model's predictions on the test set, *frame-level accuracy* [36], as well as general *classification measures* such as F-measure, precision, recall and perplexity [37].

Statistical comparisons. Methods belonging to this category are based on computing some descriptive statistics on a set of generated compositions so that they can be compared with those extracted from the training data. Examples of these statistics at the piece level are *pitch and note counts*, *pitch class and note length histograms*, *average pitch interval* and so forth. Hence, this comparison provides a weak measure of the resemblance of the generated sequences to those contained in the training set [22], which can also be interpreted as a "plagiarism score" in a way [38]. Nonetheless, a high level of similarity with the training material might also indicate an overfitting trend or a poorly configured sampling strategy.

Composition evaluation. The purpose of this evaluation is to formally assess the quality and the plausibility of generated pieces in terms of musical properties and/or theoretical rules. This can be done via computational measures derived from musicologists methods [39], or by involving a community of music experts for review [34]. Given the scarcity of computational measures that can automate this process, the manual evaluation of generated compositions, on the other hand, is a laborious task requiring a high level of musical expertise. In addition to potentially not being accessible, this evaluation methodology also involves subjectivity at different levels.

Listening tests. This last group collects two of the most common evaluation methods found in music generation works. Both these strategies are based on listening tests involving human participants, often without any musical training.

- *Turing test* (alias *discrimination test*). A group of listeners with different musical background is presented with pieces either composed by humans or generated by a model. Listeners are asked to discriminate among these groups, which basically corresponds to answering the question: *was this piece composed by a human or by a machine?* Whereas a model generating music that cannot be clearly distinguished from human work is a positive indicator of its generative capabilities, this "pass-or-fail" methodology does not allow comparisons with other automatic composition systems. Furthermore, Turing tests have been heavily criticised over the past decades [40, 41], particularly due to the complex design of listening experiments under these settings [22].
- *Blind comparison*. This methodology permits to compare music generated from different systems (usually a very few pieces per model under analysis) by letting listeners rate compositions based on specific properties, or express a preference among a given music selection including one piece from each system. The final goal is to measure the extent to which each generated track shows certain properties that would be expected from real compositions. This approach thus provides an evaluation method that allows the ranking of each model according to the so obtained measurements. From a critical perspective, this methodology is sensitive to potential biases emerging from the selection of tracks in the

experiment. The latter is particularly concerning in light of the limited collections under analysis.

To the best of our knowledge, most works based on listening tests rely on *crowd-sourcing platforms*, where participants receive a fee for their evaluation (e.g. Amazon Mechanical Turk); or on *web-based platforms* anyone can access to contribute their feedback. Hence, these experiments should be carefully designed [22], as ensuring an adequate level of control can be challenging considering that participants may not be easily filtered with a desired degree of specificity.

In conclusion, there is a lack of systematic and standardised methods for evaluation of the music generated by automatic systems, which is a major limitation in this area given that there is still no consensus on how music generated from different models can be evaluated and compared.

III. Computational analysis of music structure

The computational analysis of music structure is an active field of research, encompassing several aspects of music and involving numerous technical challenges [19]. From a general perspective, the main goal of MSA consists in decomposing or segmenting a given music representation into patterns or temporal units that correspond to musical parts, and to group these segments into musically meaningful categories depending on the use cases. Therefore, the task of MSA is typically split into two distinct sub-problems: the detection of the temporal boundaries where a transition between two consecutive segments occurs (*boundary detection*); and the labelling of the obtained segments according to their similarity or musical function (*structural grouping*).

Most methods for automatic MSA only estimate single-level (flat) segmentations, where segments typically corresponds to sections (e.g. intro, chorus, verse in Western popular music). Depending on the music genre of the music collection under analysis, the temporal granularity of these segments is usually fixed, as the duration of large-scale structural patterns is generally style-dependent. Methods for flat MSA have already enabled novel applications in music information retrieval, ranging from methods facilitating the finding and access access music information in large multimedia collections [42], to active-music listening interfaces – allowing users to enjoy music in more interactive ways than conventional playback [43]. Nevertheless, the segmentation estimated by an algorithm for flat MSA only provides a bird's-eye view of the structural properties of a music piece, meaning that any further decomposition of such large-scale segments would not be detected.

Music form, indeed, is conceived by composers and perceived by listeners following a hierarchical organisation. Sectional patterns further decompose into progressively shorter musical ideas, unveiling phrases, measures, motives and so forth. This nested organisation of music finds the most granular level with tones and chords – the staples of a composition. Hierarchical MSA specifically takes this organisation into account, as it detects structural elements at different scales. Given a music track, these methods produce a multi-level segmentation – a hierarchy of flat segmentations, where each level offers a structural segmentation at a certain granularity.

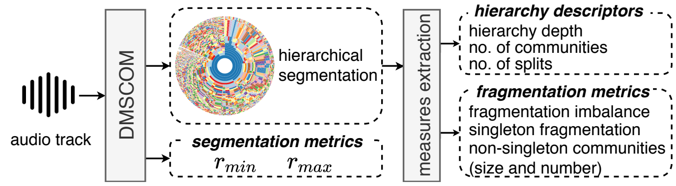


Fig. 2. Acquisition process and categorisation of the structural metrics.

To get a better technical understanding of our work, we introduce the following preliminary concepts and notation. Let $X = \{x_1, x_2, \ldots, x_T\}$ denote the set of frames sampled from a given audio track at some fixed resolution (e.g. 10Hz). A *flat segmentation* S of X is defined by temporally partitioning X into a sequence of labelled time intervals, denoted as *segments*. This can be encoded as $S : [T] \rightarrow Y$, i.e. a mapping of samples $t \in [T] = \{1, \ldots, T\}$ to a set of segment labels $Y = \{y_1, \ldots, y_k\}$. Depending on the labelling system, Y may consist of functional labels, such as *intro*, *verse* and *chorus*, or generic section identifiers such as *A* and *B*.

Let $S(i)$ identify the label of the segment containing the i-th frame in X. A *segment boundary* is any time instant at the boundary between two segments: it usually corresponds to a change of label $S(t) \neq S(t+1)$ for $t > 1$, though boundaries between similarly labelled segments can also occur (e.g. an AA form). With these concepts, we can define a *hierarchical segmentation* of depth m as a tree of flat segmentations

$$H = (S_1, S_2, \ldots, S_m),$$

where each level refines the preceding, with the ordering typically implying a coarse-to-fine structural analysis of the corresponding track. A hierarchical MSA procedure can be seen as a divisive hierarchical clustering method, with structural patterns being progressively refined across the hierarchy to detect finer structures. Following this decomposition approach, all samples belong to the same *"mother segment"* in S_1; in contrast, if structural hierarchies are not bounded, every sample will be associated to a distinct label in the last segmentation level S_m, thereby forming a (trivial) structural segment on its own called *singleton*.

IV. Measuring music structural complexity

The analysis of hierarchical segmentations can reveal insights into the richness and complexity of music structure. As an example, we show in Fig. 1 how a simple visualisation of structural hierarchies permits visualising structural differences between random, generated and real music. Here, a sunburst diagram is used (as a compact alternative of a dendrogram) to visualise a hierarchical segmentation of a track: from the top level, where all the audio samples are clustered in the same group (the unique slice in the inner-most circle), to the bottom layer, where each temporal fragment of the composition forms its own group (note the full separation in the outer-most circle).

By analysing how music structures progressively break up in a composition, structurally informative descriptors can be used to formalise this process. Our method does so in two steps.

First, hierarchical segmentations are computed with the *dynamic musical structure communities* (DMSCOM) algorithm [33]. Second, we derive structural descriptors from them.

A. Structural segmentation of audio music recordings

DMSCOM is a recently proposed state-of-the-art algorithm that produces rich and deep hierarchical segmentations of music pieces from raw audio. Compared to other procedures, it has the advantage of being unsupervised and requiring minimal setup. In addition, the algorithm does not limit the size and type of segments to detect nor the topology of the estimated hierarchies. DMSCOM segments music hierarchically and performs both boundary detection and structural grouping.

The process starts with the extraction of two sets of acoustic features from a raw audio file: *chroma features*, describing the distribution of the harmonic content of the spectrum into a fixed number of bins corresponding to pitches of a musical scale; and *mel-frequency cepstral coefficients (MFCC)*, encoding the timbral properties of the signal. The instrumentation and the timbral properties of a sound source are indeed of great importance for the human perception of musical structure [44], and the same can be said for the pitch content, upon which harmonic and melodic sequences are built [45]. In fact, harmonic features alone have turned out to be effective mid-level representations for music structure analysis [46]. Nevertheless, focusing on a single audio descriptor could potentially lead to undetected structural boundaries, as previous research found that a listener's attention mostly shifts among timbral and chroma features throughout a piece [32]. For this reason, DMSCOM takes both these features into account, to create a single compact descriptor that retains timbral and harmonic/melodic properties of the track in a graph object.

Following their extraction, both features sets are beat-synchronised – by averaging all the vectors belonging to the same estimated beat, to reduce data dimensionality and remove transient noise. This is done by using a dynamic programming algorithm for beat tracking that directly operates on the spectogram [47]. Then, the self-similarity of each (beat-synchronised) feature set is computed using a Gaussian kernel, thus yielding two self-similarities matrices (SSMs) of size $N \times N$, where N is the number of beats. From the SSM computed on chroma features, the recurrence graph R is obtained by weighting edges according to the similarity of the corresponding beat-aggregated chroma vectors; whereas the proximity graph Δ is defined analogously from the timbral SSM, with the only exception that only edges connecting temporally consecutive beats are preserved. In sum, R captures harmonic and melodic repetitions in a given track, whereas Δ preserves the sequential nature of music by connecting consecutive nodes according to their timbral consistency.

Since MFCC and chroma-based features are both related to human perception of musical structure, combining them into a single representation would provide a rich and informative descriptor for MSA. To that end, the recurrence and the proximity graphs are fused into a single graph $G = (V, E)$ in such a way as to avoid proximity connections being excessively outnumbered by the repetition connections. In the resulting music graph G, nodes still correspond to beats and edges now encode their timbral and harmonic relationships, with the topology of the network ensuring the connectedness of temporally subsequent nodes. The edge set is represented as an adjacency matrix $E \in \mathbb{R}^{|V| \times |V|}$ where $|V| = N$ denotes the number of nodes (or beats) and each $E_{i,j}$ holds the relationship between nodes i and j. The procedure for the creation of the music graph is akin to [48], although the SSM computed on the beat-synchronised chroma features undergoes a *dynamic filtration* step before the the recurrence graph R is constructed. This is done to retain structurally meaningful connections, and it is controlled by the total strength of the network and a hyper-parameter λ – the *coefficient of filtration*, controlling the severity of the filtration process.

Structural segments at different levels of granularity are then detected from the music graph G. Each segment collects nodes with similar features, with the propensity of a node being part of a group (or community) depending on the resolution level at a certain layer in the hierarchy. A community thus corresponds to the identity of a structural segment, collecting nodes with homogeneous musical properties at a certain resolution level. With DMSCOM, this is achieved by using the multi-resolution hierarchical community detection procedure of [49] on G. The key element of this recursive procedure is the resistance parameter r, used to control the granularity of structural patterns at a certain segmentation level. In particular, self-loops with weight equal to r are introduced for all nodes in the adjacency matrix E in order to control the propensity of nodes forming communities: when $r < 0$ we can reveal super-structures, since nodes are more reluctant to form small-scale communities; when $r > 0$ we incentive individual links thereby revealing sub-structures. For a more detailed overview of DMSCOM and its experimental evaluation, we refer to [33].

B. Metrics of music structural complexity

In this work, we use a collection of metrics quantifying specific properties of the hierarchical segmentation process to characterise music structural complexity. They were grouped into three categories: *segmentation metrics*, *hierarchy descriptors* and *fragmentation metrics* (see Fig. 2).

Segmentation metrics. This group of metrics includes parameters related to the segmentation algorithm – the extreme values of the resistance parameter r needed to hierarchically partition a given track (c.f. Section IV-A). r_{min} is the smallest negative value of the resistance parameter s.t. all nodes of the music graph belong to a single community. In other words, it relates to the amount of negative "force" that has to be applied to each node in the graph to enclose all of them within the same segment, i.e., how much nodes are resisting to form a single community. r_{max} is the resistance of the music graph to decompose fully into singleton communities – the most atomic structures. It corresponds to the smallest positive r s.t. each node forms a community on its own (*singleton*).

Hierarchy descriptors. One of the most intuitive properties to describe the complexity of a hierarchical segmentation is the number of levels it contains – the *depth of the hierarchy*, with 2 being the minimum possible depth (from the mother

TABLE I
TAXONOMY OF THE STRUCTURAL METRICS AND OUTLINE OF THEIR FUNCTION IN RELATION TO THE HIERARCHICAL SEGMENTATIONS.

Group	Metric	Aggregation	Values	Description
Segmentation metrics	r_{min}	-	1	Smallest value of r to enclose all nodes in a single community.
	r_{max}	-	1	Smallest value of r to break the graph completely into singletons.
Hierarchy descriptors	hierarchy depth	-	1	Relative number of segmentation levels in the hierarchy.
	number of communities	-	1	Total relative number of detected communities.
	number of splits	-	1	Total relative number of community splits across the hierarchy.
Fragmentation metrics	singleton fragmentation	-	1	Propensity of communities to become singletons (single-node communities) towards the bottom of the hierarchy.
	no. of non-singletons communities	hierarchy	4	Relative number (proportion) of non-singleton communities per level, aggregated across the hierarchy (M, SD, CV, SampEn).
	size of non-singleton communities	hierarchy	4	Relative size of non-singleton communities per level, aggregated across the hierarchy (M, SD, CV, SampEn).
	fragmentation imbalance	level, hierarchy	12	Degree of distribution of nodes from parent to children communities, aggregated for each level (min, max, M) then across the hierarchy (M, SD, CV, SampEn).

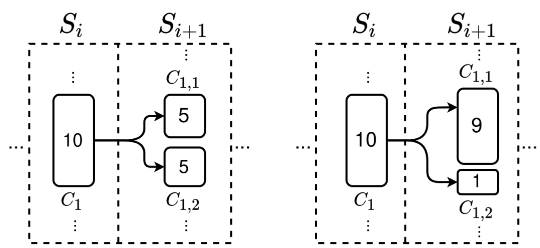


Fig. 3. Examples of maximally balanced (*left*) and imbalanced (*right*) splits.

segment to all singletons) and $|V|$ the maximum (the mother segment losing a node at every level). Other structurally informative indicators are the number of communities and the number of splits, which indicate the amount of structural elements identified across the hierarchy and the amount of splits from which they originate. More precisely, a split is counted whenever a parent community at level i originates at least a new community at level $i+1$, in addition to the trivial one preserving the same nodes of the parent. Given that the depth of the hierarchy also depends on the music piece length, these three metrics are scaled with respect to the maximum values they can take in a given music graph. In this way, it is possible to compare tracks with different duration and metre.

Fragmentation metrics. These metrics describe key aspects of the decomposition trend of hierarchical segmentations.

The *fragmentation imbalance* of a split is an indicator of how nodes distribute from a parent community C_l with $|C_l|$ nodes at level i to its children communities $C_{l,1}, \ldots, C_{l,m}$ at level $i+1$. It ranges from 0, when $|C_{l,k}| = \frac{|C_l|}{m}$ $\forall k \leq m$, to 1 for maximal imbalance – when $\exists$ $C_{l,k}$ s.t. $|C_{l,k}| = |C_l| - m + 1$, with all the other new communities being singletons (Fig. 3). Because the fragmentation imbalance is computed from an individual split, obtaining a single metric for the whole hierarchy requires two steps of aggregation: first we aggregate the fragmentation imbalance of all the communities splitting between each couple of successive levels (*level aggregation*), then we aggregate across this hierarchy (*hierarchy aggregation*). We use the minimum, maximum and mean functions for level aggregation, and mean (M), standard deviation (SD), coefficient of variation (CV) and sample entropy (SampEn) for hierarchy aggregation. SD, CV and SampEn are used to study the dynamicity and predictability of the the hierarchical fragmentation process. In particular, SampEn is a modification of approximate entropy that is independent from sequence length [50]. In statistical signal processing, approximate entropy is used as a measure of irregularity and unpredictability of fluctuations of time-series data [51]. Sequences with several repetitive patterns receive small SampEn; less predictable (more complex) ones yield higher values.

To describe the degree of fragmentation of communities at a certain level, which indicates the persistence of nontrivial structural components, we consider the proportion of *non-singleton communities* together with their relative size. Given a segmentation level S_i the former one is obtained by counting the number of non-singleton communities and scaling it by the total number of communities in S_i. Similarly, the size of non-singletons communities – the number of nodes they contain, is scaled by $|V|$. As we obtain a time series for each metric – one for the number and one for the size of non-singleton communities per level, only hierarchy aggregation is required.

Finally, another metric – *singleton fragmentation* – describes how far in the hierarchy nodes tend to form singletons, indicating the pace at which the most atomic structural components – beats – tend to separate from larger structures. Given that the singleton fragmentation pertains to each $v \in V$, values are computed independently for each node and then averaged. More formally, assuming that a node v becomes a singleton in S_i (the i-th level in the structural hierarchy S), the fragmentation imbalance of v is simply defined as $\frac{i}{|S|}$, where $|S|$ denotes the hierarchy depth (the total number of segmentation levels in S). This metric ranges in the $[0, 1]$ interval. Values close to 0 indicate a slower and persistent fragmentation of the graph; if nodes tend to become singletons towards the end of the hierarchy, values would tend to 1.

V. EXPERIMENTS AND RESULTS

In this section, we describe our test framework, experimental procedures and results pertaining to the investigation of metrics to quantify the structural complexity of music pieces.

A. Music dataset

Our first step was to create a music dataset that included music with different levels of structural complexity. In particular, we included three types of subsets, which we think establishes a good test-bed: *human-composed* music (high complexity; "real" music), *computer-generated* music (which we expect will have intermediate complexity) and *random* music (minimal complexity). In our view, each of these subsets is associated with a different level of music structural complexity, which allows to investigate whether our metrics permit discriminating between these broad complexity levels.

Another way to interpret the proposed experimental methodology in light of the three subsets is as follows. First, random and real music are needed to verify whether the structural complexity metrics can confirm our objective expectations: the former having little or no structure, and the latter possessing a realistic/maximal level of music structural complexity. If this is confirmed, hence the structural complexity metrics conform with our assumptions, then we would obtain a lower bound (random music) and an upper bound (real music) for the structural complexity of music. Therefore, the next question is to see where generated music stands in this space.

1) Human-composed (real) music: This subset includes a selection of "real" music, i.e., music written by human composers, which includes a partition of the Pianomidi [52] and SALAMI [53] datasets. The first, is a well-known dataset of classical music for piano (in symbolic format) spanning from the baroque era to the late Romantic and impressionist periods. The second, is a dataset used for audio-based MSA, providing live performances of pop/rock/blues music together with their structural annotations. We expect real music[2] to exhibit the highest degree of structural complexity, with short-term (e.g., motifs), mid-term (e.g., phrases) and long-term (e.g., sections) structural elements emerging from the compositions.

2) Computer-generated music: This subset includes music generated by three state-of-the-art machine learning models: the *Basic RNN*, the *Lookback RNN* and the *Attention RNN* [54]. These three models are particularly interesting because they have different levels of ability to produce musical content with long-term structures. Furthermore, all these models have a comparable number of learnable parameters, use the same encodings of symbolic music, and were trained on the same music corpus using similar strategies and optimisation methods. This ensures that the musical properties of the generated compositions – and in particular, their increased level of structural complexity, can be attributed to the architectural design of these models rather than being the result of other factors that we would not be able to trace. Therefore, these models provide a controlled testbed for our experiments.

The *Basic RNN* is a vanilla LSTM recurrent neural network [15] trained for one-step ahead prediction on symbolic music sequences. This architecture is representative of several works in the literature, from the first attempts at music modelling with LSTMs [55] to more recent architectures [56].

The *Lookback RNN* is an extension of the Basic RNN which introduces time-delayed connections and requires several additional inputs at each time step (a rich conditioning signal). More precisely, in addition to the previous musical token and assuming that all pieces have time signature of 4/4, a Lookback RNN receives the following information: (i) the specific events from one and two measures ago; (ii) whether the last token was repeating the event from one or two bars before it; (iii) two labels denoting whether the network has to repeat the event from one or two measures ago, respectively; (iv) the current relative position within the measure in terms of quarters. These architectural changes aim to facilitate the model to learn structural regularities from sequences by providing prior knowledge of metrical structure to the network [57].

Finally, the *Attention RNN* is another architecture that expands the memory capacity of the LSTM by means of an attention mechanism [58], which enables the network to contextually access the generated output sequence up to a certain number of elements. This frees the network from having to store musical content in the LSTM cell's state. Attention mechanisms have become staple in modern architectures for music generation [18], because they are more effective at modelling long-term dependencies in sequential data – a desideratum for the generation of music with increased levels of structural complexity [59]. Although generated music still does not posses a clear structural identity [60], autoregressive models tend to produce music exhibiting repetition and variation only at a local level [36]. Nevertheless, these subsets were specifically chosen due to the increasing level of structural complexity their music is expected to exhibit according to [54], starting from the *Basic RNN*.

3) (Quasi-)Random music: The last subset contains music artificially generated or manipulated in a way to compromise most of its structural integrity. First, we include a group of pieces generated with a Basic RNN (the same model in Section V-A2) after a single epoch of training on the Pianomidi dataset. Second, we used a scrambling method [61] that randomly shuffles beat-aggregated feature vectors to destroy any structural relationship at the beat level on all tracks of SALAMI and Pianomidi datasets. We expect that music pieces from both these groups will have minimal structure.

4) Data overview: All sets comprise music pieces with duration of 180 ± 20 seconds – a duration we find suitable for the identification of structures spanning from short- to long-term scales. To obtain audio tracks, the MIDI files in our collection are synthesised using `FluidSynth` and the freely available `FR3` General-MIDI soundfont[3]. The diversity of musical material (live performances, synthesised symbolic music) and the inclusion of different genres (classical, pop, rock, etc.) provides a challenging experimental setup to test the robustness and the generalisation of the structural metrics.

B. Structural complexity metrics and summaries

Following the procedure detailed in Section IV-B, we computed all metrics for the music tracks in our dataset. To produce hierarchical segmentations, the coefficient of filtration of DMSCOM was set to its default value ($\lambda = 4$), as it was found to achieve state of the art results on SALAMI [33].

[2]The terms *real* and *human-composed* music are used interchangeably.

[3]https://www.fluidsynth.org/

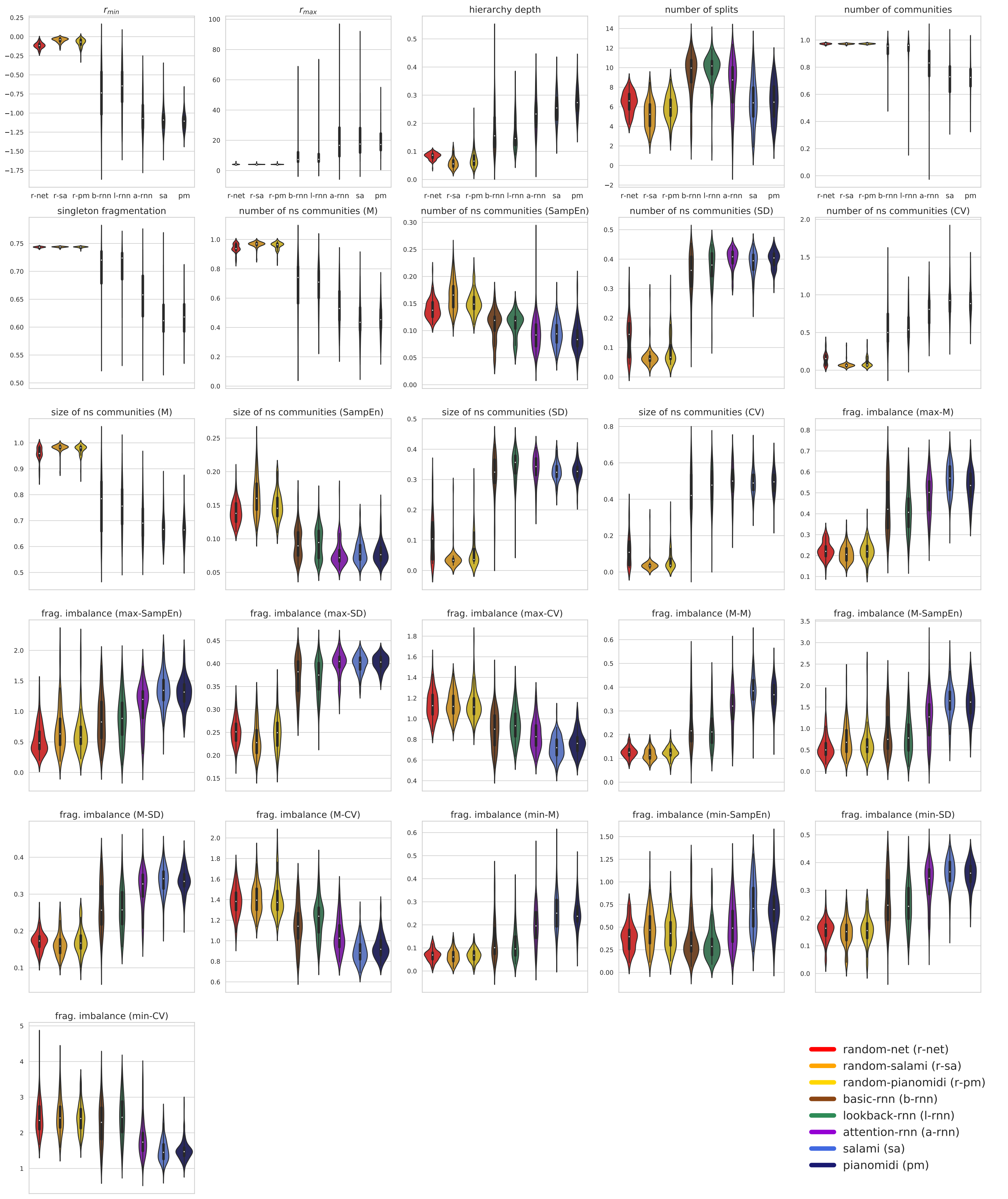


Fig. 4. Overview of the structural metrics divided by each music subset in our collection – *random music* generated from an untrained LSTM (r-net) or by beat-shuffling on SALAMI (r-sa) and pianomidi (r-pm); and *computer-generated* music from a basic (b-rnn), lookback (l-rnn) and attention (a-rnn) LSTM; *real classical music* for piano selected from the SALAMI (sa) and pianomidi (pm) datasets. For those metrics requiring *hierarchy aggregation* only, the name of the functional is reported in brackets; when *level aggregation* needs to be applied before the former, both functionals are reported.

Each metric was then grouped according to the music selection each track belongs to (e.g. Pianomidi), thereby enabling the analysis and the comparison of the distributions of these groups (Figure 4). To detect statistically significant differences between the music selections, pairwise Kolmogorov-Smirnov tests were performed for all metrics (Bonferroni corrections were applied to control for family-wise error rate of multiple comparisons). This in-depth statistical analysis, together with a table reporting the mean and the standard deviation of each metric per music selection, are provided in Appendix A.

From the results we found that real music, compared to the other subsets, is harder to segment, as it results in deeper hierarchies following a more complex decomposition trend. The values of the resistance parameter r indicate that real music requires more energy to enclose all nodes within a single community (r_{min}), as well as to fully decompose networks into singletons (r_{max}). This is particularly evident between real and random music, with generated music only approaching the human-composed class with the *Attention RNN* – a pattern we found for several other metrics.

A similar trend can be observed for the hierarchy descriptors. Hierarchies obtained from the segmentation of human-composed music are the deepest in terms of segmentation levels, with the fewest relative number of communities resulting from a reduced number of splits. In contrast, random music is segmented in shallow hierarchies with the highest number of estimated communities, although still originating from a few splits. Generated music, instead, sits in between the former groups for hierarchy depth and number of communities, stemming from the largest number of splits. Therefore, it can be observed that the relative number of splits is a structural property shared between real and random music, and allows to distinguish these groups from generated music.

Regarding the decomposition trend of music, a level-to-level analysis of non-singleton communities and their fragmentation across the hierarchies revealed the following insights. For real music, the relative number and the size of non-singleton communities per level are the least complex to predict, with the highest standard deviation and coefficient of variation. On the contrary, the trend of non-singleton communities in random music is the most complex process, with the lowest amount of variation. The decomposition trend of human-composed music thus follows some regularity, which is particularly evident for the relative size of non-singleton communities.

The choice of level aggregation (LA) function did not influence the analysis of the fragmentation imbalance, as both *min*, *max* and *mean* provided similar insights. From this analysis, we found that real music produces the most imbalanced splits (0.38 and 0.37 average fragmentation imbalance with mean LA for SALAMI and Pianomidi respectively) compared to the other groups. The higher fragmentation imbalance of human-composed music, together with its reduced number of non-singleton communities per level, indicates a *leaky segmentation* behaviour. This means that the hierarchical segmentations of real music tend to be inflated by the number of nodes separating from larger communities as singletons, which in turn contributes to increase the hierarchy depth. The singleton fragmentation metric provides further insights into the pace at which this leaky segmentation occurs across hierarchies. Given that human-composed music has the lowest singleton fragmentation, the full decomposition of structural segments into singletons does not occur right at the bottom of hierarchies – a behaviour which is more pronounced for random and generated music. Indeed, the leaky segmentation of real music is more gradual throughout the hierarchies, rather than happening mostly towards their bottom levels.

C. *Structural complexity of different music subsets*

As can be inferred from the analysis above, there is redundancy between the various metrics, which can indicate the existence of latent variables. This was confirmed via a correlation analysis, which revealed strong and significant correlation between more than 80% of the metrics. To identify potential latent variables, we employed principal component analysis (PCA) on the whole set of metrics after discarding the segmentation metrics. This ensures comparability of the latent variables independently of the MSA procedure used to produce hierarchical segmentations (r-values are DMSCOM-specific). The first two principal components explained 83% of the variance in the structural metrics. Hereinafter, we will focus on the first two principal components and refer to them as *structural summaries* – a compact descriptor of the structural properties captured by the original metrics.

As latent variables were identified, we analysed the distributions of the structural summaries (denoted as PC0 and PC1) of the various music subsets. These are plotted in Fig. 5.

To detect differences between the distributions of each selection, we computed a series of Kruskal-Wallis H-tests and found that they differ significantly for both summaries ($\chi^2 = 630.85$ and $\chi^2 = 342.89$, $p < 0.0001$; for PC0 and PC1, respectively). The results of the pairwise comparisons conducted jointly for the structural summaries using bivariate Kolmogorov-Smirnov (KS) tests (after Bonferroni corrections) are shown in Fig. 6. Due to the large number of comparisons, the results are reported as a *heatmap* which highlights statistically significantly different ($p < 0.05$) subsets in yellow and non-significant in green (similar distributions).

The pairwise analysis revealed five distinct clusters of structural complexity: 1) *random-salami* and *random-pianomidi* (or *randomised-human*; 2) *random-net* on its own; 3) *Basic RNN* and *Lookback RNN* (hereinafter, *simple RNN*; 4) *Attention RNN*; and 5) "real" music (*SALAMI* and *Pianomidi*).

In sum, the structural summaries allow discriminating between the different music subsets in our dataset, and can unveil further subdivisions in each of these groups. These subdivisions are retained structurally meaningful, as they confirm that scrambled music still preserves some degree of structure (as the perturbation is operated at the beat level), and the importance of attention mechanisms for automatic composition models; however, the architectural changes of the *Lookback RNN* did not significantly contribute to the structural complexity of the generated music compared to a vanilla LSTM model. To conclude, the structural summaries provide a formal and automatic way to inspect where a given music piece or collection sits within the structural complexity plane.

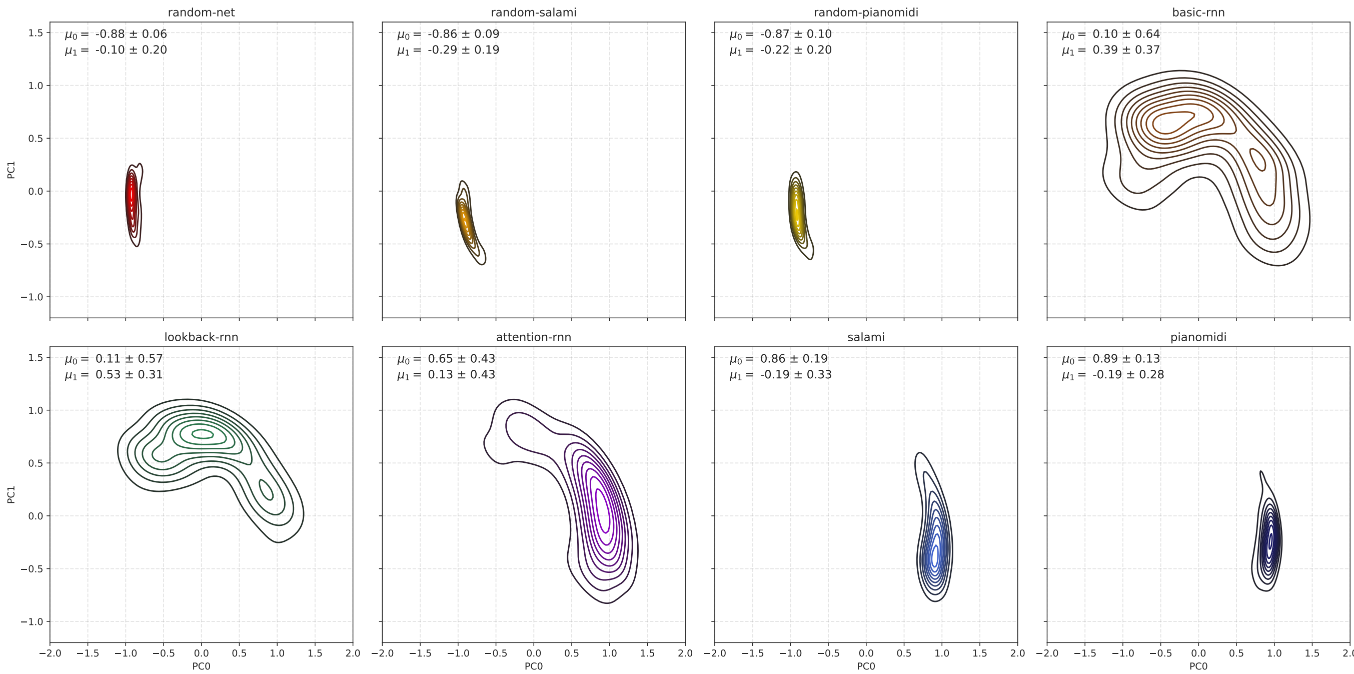


Fig. 5. Illustration of the distributions of the structural summaries for each group after bivariate kernel density estimation. The mean and the standard deviation for each dimension (PC0 and PC1) are reported in the top-left of each plot.

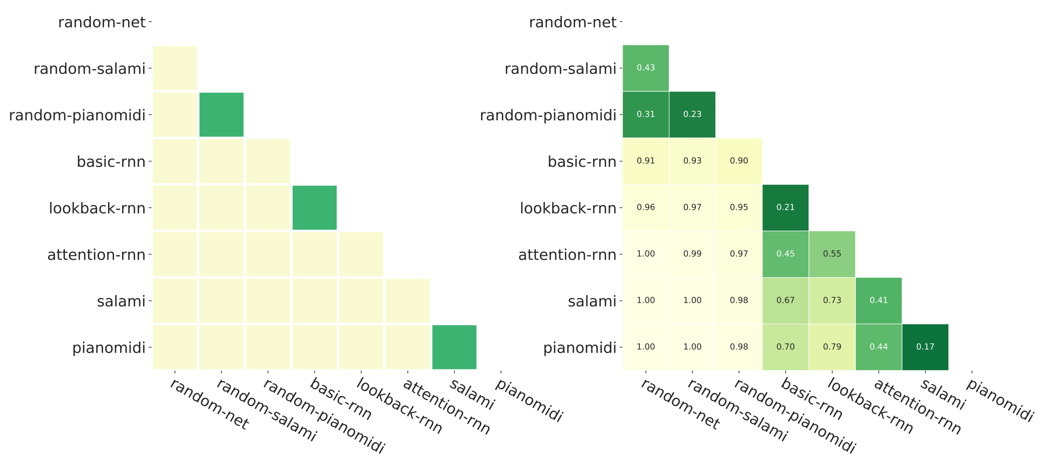


Fig. 6. Statistical analysis of the structural summaries: pairwise comparison of the groups (*left*), with yellow denoting statistical difference ($p < 0.05$) and green otherwise; Kolmogorov-Smirnov scores (*right*) as a distance function between groups, illustrated with a colour-map ranging from green to yellow.

VI. Structural Summaries: A Framework for Measuring Music Structural Complexity

The analysis of the structural summaries on our dataset provides a compact and effective framework to evaluate the structural complexity of computer-generated music. One possibility is to compare a corpus of generated pieces with the five reference classes of structural complexity we described in the previous section. Similarly to our previous analysis, Kolmogorov-Smirnov (KS) tests could then be used to compute the pair-wise differences between the bivariate distribution of the structural summaries extracted from the given collection and those of the reference classes. Furthermore, the resulting KS scores, ranging in $[0, 1]$, can then be interpreted as a dissimilarity measure between the generated corpus and each of these classes (Figure 6, *right*).

If the intention is to evaluate a single track, the Mahalonobis distance between its structural summaries (a data point) and the distribution of each reference class could also be computed. The Mahalonobis distance is suitable for this purpose as it is an effective multivariate distance function that measures the distance between a single data point and a distribution. An example of this approach is shown in Table II for Vivaldi's "La Caccia" (Autunno part III) – a classical music piece for orchestra from the Baroque period. In addition to the original orchestral version, we included: a structurally simplified version of the former piece, that is used for educational

TABLE II
MAHALONOBIS DISTANCE OF THE STRUCTURAL SUMMARIES EXTRACTED FROM EACH VERSION OF VIVALDI'S LA CACCIA W.R.T. THE REFERENCE COMPLEXITY GROUPS. THE DISTANCE OF THE CLOSEST REFERENCE CLASS IS HIGHLIGHTED IN BOLD FOR EACH TRACK.

	random-net	randomised-human	simple RNN	attention RNN	human
Original	28.68	22.42	2.08	0.90	**0.52**
Simplified	18.26	13.45	3.13	**3.00**	4.63
Randomised	1.22	**0.44**	3.67	5.63	11.65

purposes (recorder practise in secondary school); as well as a randomised version of it, following the same scrambling procedure outlined in Section V-A3. As shown, both the original and the randomised versions received the smallest distance to their expected classes - *human* (0.52) and *randomised-human* (0.44), respectively. The simplified version, instead, has structural properties closer to those of generated music, and, in this particular case, to the *Attention RNN* outputs. These results are thus in line with the consideration that the structural simplification of the educational track was artificially operated to make it easier for novice students to analyse and play the piece on the recorder. Although the Mahalonobis distance of the structural summaries of the simplified version from their closest distributions – 3.13 from the *Simple RNN* and 3.00 from the *Attention RNN*, is not as low as those of the original and random versions, there is still reasonable margin to the other reference complexity classes.

From a statistical perspective, the use of our framework would be more reliable if distributions are to be compared, rather than individual tracks. Indeed, comparing two distributions under the same assumptions would provide a more robust statistical indicator, rather than comparing a data point against a distribution. This approach would also align to the expected use case for automatic evaluation. Experimenters would generate a reasonable number of tracks from their music generation system, extract a number of metrics to quantify specific musical properties of the compositions, along with their structural summaries. The latter would then be compared to the reference complexity classes for structural evaluation. In any case, both the corpus and the single-track evaluations necessitate the principal components matrix from our previous experiments (Section V-C) before any comparison is possible. In fact, as a preliminary step, the structural complexity metrics extracted from the hierarchical segmentation of the given track(s) need to be projected onto the principal components, so that the structural summaries can be obtained.

VII. CONCLUSIONS

In this paper, we addressed the automatic analysis of structural complexity of music – an open problem in the field of computational music analysis which is currently jeopardising the systematic evaluation and the comparison of music generation systems. Our approach builds upon computational methods for hierarchical music structure analysis (MSA), capable of unveiling the nested organisation of music from long and articulated musical ideas (e.g. sections) to progressively shorter and simpler structural components (e.g. motifs). Given a music track or a synthesised symbolic piece, a structural segmentation is first estimated as a hierarchical object using a state-of-the-art method for hierarchical MSA. This is followed by the extraction of a set of metrics to formally describe these hierarchies and the decomposition of music structures therein.

To test the ability of our metrics to characterise structural properties of music, we computed them on a dataset including random, real and computer-generated music – groups which we expect to be associated with different degrees of structural complexity. After analysing their distribution on each group, we found that not only our metrics permit to discriminate between them, but further non-trivial subdivisions can also be identified according to the structural properties of the compositions. Our results thus revealed how these hierarchies differ as mathematical objects, and demonstrated the effectiveness of our metrics as structural descriptors of music.

We also showed how these metrics, together with their statistical analysis on the dataset, can provide a compact framework for automatically evaluating the structural complexity of a given collection of music or individual tracks. To the best of our knowledge, our method is the first to achieve this and comes with the following strengths: (i) it relies on simple metrics and functionals describing the decomposition process of music into nested and progressively more granular structures; (ii) our metrics exclusively capture structural aspects of music, due to the preliminary MSA step; (iii) we did not attempt at subjectively defining structural complexity, but we relied on the assumption that pseudo-random and human-composed music would belong to different complexity classes. In addition, as our method takes music recordings as input, the resulting framework can be used to evaluate both audio-based and symbolic music generation systems, although a sonification step of compositions is needed in the latter case.

Overall, this work demonstrated that structurally informative descriptors can be extracted from the hierarchical segmentation of music, and made a first step towards the automatic evaluation of the structural complexity of computer-generated music. Planned future work includes a broader analysis of computer-generated music, and the investigation of our structural summaries from a musicological perspective.

REFERENCES

[1] D. J. Levitin, *This is your brain on music: The science of a human obsession*. Penguin, 2006.
[2] L. B. Meyer, *Emotion and meaning in music*. University of chicago Press, 2008.
[3] F. Lerdahl, R. S. Jackendoff, and R. Jackendoff, *A generative theory of tonal music*. MIT press, 1983.
[4] P. Goetschius, *Lessons in music form: A manual of analysis of all the structural factors and designs employed in musical composition*. Oliver Ditson Company, 1904.
[5] W. W. Gaver and G. Mandler, "Play it again, sam: On liking music," *Cognition and Emotion*, vol. 1, no. 3, pp. 259–282, 1987.
[6] A. D. Patel, "Language, music, syntax and the brain," *Nature neuroscience*, vol. 6, no. 7, pp. 674–681, 2003.
[7] P. Gomez and B. Danuser, "Relationships between musical structure and psychophysiological measures of emotion." *Emotion*, 2007.

[8] R. Fiebrink, B. Caramiaux, R. Dean, and A. McLean, *The machine learning algorithm as creative musical tool*. Oxford University Press, 2016.

[9] A. Papadopoulos, P. Roy, and F. Pachet, "Assisted lead sheet composition using flowcomposer," in *International Conference on Principles and Practice of Constraint Programming*. Springer, 2016, pp. 769–785.

[10] C. P. Martin, K. O. Ellefsen, and J. Torresen, "Deep predictive models in interactive music," *arXiv preprint arXiv:1801.10492*, 2018.

[11] G. Brunner, Y. Wang, R. Wattenhofer, and J. Wiesendanger, "Jambot: Music theory aware chord based generation of polyphonic music with lstms," in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2017, pp. 519–526.

[12] D. Herremans, C.-H. Chuan, and E. Chew, "A functional taxonomy of music generation systems," *ACM Computing Surveys (CSUR)*, vol. 50, no. 5, pp. 1–30, 2017.

[13] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, *Deep learning techniques for music generation*. Springer, 2020.

[14] D. Eck and J. Schmidhuber, "A first look at music composition using lstm recurrent neural networks," *Istituto Dalle Molle Di Studi Sull' Intelligenza Artificiale*, p. 48, 2002.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, 1994.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[18] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *International Conference on Learning Representations*, 2018.

[19] M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer, 2015.

[20] S. Dieleman, A. v. d. Oord, and K. Simonyan, "The challenge of realistic music generation: modelling raw audio at scale," *arXiv preprint arXiv:1806.10474*, 2018.

[21] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[22] L.-C. Yang and A. Lerch, "On the evaluation of generative models in music," *Neural Computing and Applications*, vol. 32, no. 9, pp. 4773–4784, 2020.

[23] B. Di Giorgi, S. Dixon, M. Zanoni, and A. Sarti, "A data-driven model of tonal chord sequence complexity," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 11, pp. 2237–2250, 2017.

[24] C. Weiss and M. Müller, "Tonal complexity features for style classification of classical music," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 688–692.

[25] L. Mihelač and J. Povh, "The impact of the complexity of harmony on the acceptability of music," *ACM Transactions on Applied Perception (TAP)*, vol. 17, no. 1, pp. 1–27, 2020.

[26] G. T. Toussaint and K. Trochidis, "On measuring the complexity of musical rhythm," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 2018, pp. 753–757.

[27] A. Pease, K. Mahmoodi, and B. J. West, "Complexity measures of music," *Chaos, Solitons & Fractals*, vol. 108, pp. 82–86, 2018.

[28] C. Wöllner, "How to quantify individuality in music performance?" *Frontiers in psychology*, vol. 4, p. 361, 2013.

[29] S. Streich *et al.*, *Music complexity: a multi-faceted description of audio content*. Citeseer, 2006.

[30] M. Mauch and M. Levy, "Structural change on multiple time scales as a correlate of musical complexity." in *ISMIR*, 2011, pp. 489–494.

[31] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *2000 ieee international conference on multimedia and expo. icme2000. proceedings. latest advances in the fast changing world of multimedia (cat. no. 00th8532)*, vol. 1. IEEE, 2000, pp. 452–455.

[32] J. B. Smith and E. Chew, "Using quadratic programming to estimate feature relevance in structural analyses of music," in *Proceedings of the 21st ACM international conference on Multimedia*, 2013, pp. 113–122.

[33] J. de Berardinis, M. Vamvakaris, A. Cangelosi, and E. Coutinho, "Unveiling the hierarchical structure of music by multi-resolution community detection," *Transactions of the International Society for Music Information Retrieval*, vol. 3(1), pp. 82–97, 2020.

[34] B. L. Sturm and O. Ben-Tal, "Taking the models back to music practice: Evaluating generative transcription models built using deep learning," *Journal of Creative Music Systems*, vol. 2, no. 1, 2017.

[35] F. Carnovalini and A. Rodà, "Computational creativity and music generation systems: An introduction to the state of the art," *Frontiers in Artificial Intelligence*, vol. 3, p. 14, 2020.

[36] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *Proceedings of the 29th International Conference on Machine Learning*, 2012, pp. 753–757.

[37] A. Ycart, E. Benetos *et al.*, "A study on lstm networks for polyphonic music sequence modelling." ISMIR, 2017.

[38] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, "Music transcription modelling and composition using deep learning," *arXiv preprint arXiv:1604.08723*, 2016.

[39] C.-H. Chuan and D. Herremans, "Modeling temporal tonal relations in polyphonic music through deep networks with a novel image-based representation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[40] C. Ariza, “The interrogator as critic: The turing test and the evaluation of generative music systems,” *Computer Music Journal*, vol. 33, no. 2, pp. 48–70, 2009.

[41] A. Pease and S. Colton, “On impact and evaluation in computational creativity: A discussion of the turing test and an alternative proposal,” in *Proceedings of the AISB symposium on AI and Philosophy*, vol. 39. Citeseer, 2011.

[42] F. Kurth, M. Müller, D. Damm, C. Fremerey, A. Ribbrock, and M. Clausen, “Syncplayer-an advanced system for multimodal music access.” in *ISMIR*, vol. 5, 2005, pp. 381–388.

[43] M. Goto and R. B. Dannenberg, “Music interfaces based on automatic music signal analysis: New ways to create and listen to music,” *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 74–81, 2018.

[44] M. J. Bruderer, M. F. McKinney, and A. Kohlrausch, “Structural boundary perception in popular music.” in *ISMIR*, 2006, pp. 198–201.

[45] J. Paulus, “Improving markov model based music piece structure labelling with acoustic information.” in *ISMIR*, 2010, pp. 303–308.

[46] M. A. Bartsch and G. H. Wakefield, “Audio thumbnailing of popular music using chroma-based representations,” *IEEE Transactions on multimedia*, vol. 7, no. 1, pp. 96–104, 2005.

[47] D. P. Ellis, “Beat tracking by dynamic programming,” *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[48] B. McFee and D. Ellis, “Analyzing song structure with spectral clustering,” in *15th International Society for Music Information Retrieval Conference*, 2014, pp. 405–410.

[49] C. Granell, S. Gomez, and A. Arenas, “Hierarchical multiresolution method to overcome the resolution limit in complex networks,” *International Journal of Bifurcation and Chaos*, vol. 22, no. 07, p. 1250171, 2012.

[50] J. S. Richman and J. R. Moorman, “Physiological time-series analysis using approximate entropy and sample entropy,” *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 278, no. 6, pp. H2039–H2049, 2000.

[51] S. M. Pincus, “Approximate entropy as a measure of system complexity.” *Proceedings of the National Academy of Sciences*, vol. 88, no. 6, pp. 2297–2301, 1991.

[52] K. Bernd, “Webpage of the pianomidi dataset,” http://www.piano-midi.de/, 1996, accessed: 2020-07-02.

[53] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, and J. S. Downie, “Design and creation of a large-scale database of structural annotations,” in *12th International Society for Music Information Retrieval Conference*, Oct. 2011, pp. 555–560. [Online]. Available: https://doi.org/10.5281/zenodo.1416884

[54] W. Elliot, “Generating Long-Term Structure in Songs and Stories,” https://magenta.tensorflow.org/2016/07/15/lookback-rnn-attention-rnn/, 2016, accessed: 2020-07-02.

[55] D. Eck and J. Schmidhuber, “Finding temporal structure in music: Blues improvisation with lstm recurrent networks,” in *Proceedings of the 12th IEEE workshop on neural networks for signal processing*. IEEE, 2002, pp. 747–756.

[56] B. Sturm, J. F. Santos, and I. Korshunova, “Folk music style modelling by recurrent neural networks with long short term memory units,” in *16th International Society for Music Information Retrieval Conference*, 2015.

[57] D. D. Johnson, “Generating polyphonic music using tied parallel networks,” in *International conference on evolutionary and biologically inspired music and art*. Springer, 2017, pp. 128–143.

[58] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[59] J. de Berardinis, S. Barrett, A. Cangelosi, and E. Coutinho, “Modelling long-and short-term structure in symbolic music with attention and recurrence,” in *Proceedings of The 2020 Joint Conference on AI Music Creativity*, 2020, pp. 1–11.

[60] N. Collins, “Musical form and algorithmic composition,” *Contemporary Music Review*, vol. 28, no. 1, pp. 103–114, 2009.

[61] E. Fedorenko, J. H. McDermott, S. Norman-Haignere, and N. Kanwisher, “Sensitivity to musical structure in the human brain,” *Journal of Neurophysiology*, vol. 108, no. 12, pp. 3289–3300, 2012.

.

Appendix

Statistical analysis of the structural metrics

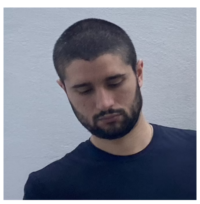
This section provides further details on the results of our experiment reported and illustrated in Section V-B. As part of the methodology, each structural metric is considered independently (before dimensionality reduction), separated for each subset – random, computer-generated and real music, and aggregated by music selection (e.g. random-net) in our dataset. Following aggregation, the mean and the standard deviation of each metric per music selection are reported in Table III.

To complement this study, we also report the results of the statistical analysis, performed independently on each structural metric, in relation to the values taken by each music subset. As done for the structural summaries, for each metric, Kolmogorov-Smirnov tests are used to detect statistically significant differences between the various music selections (with Bonferroni corrections being considered to account for multiple comparisons). These are illustrated in Fig 7, following the same conventions introduced in Section V-B.

TABLE III

OVERVIEW OF THE STRUCTURAL METRICS COMPUTED ON THE DATASET – MEAN AND STANDARD DEVIATION ARE REPORTED FOR EACH MUSIC SUBSET, WITH THE MAXIMUM VALUES PER-METRIC IN BOLD. LA AND HA DENOTE LEVEL AND HIERARCHY AGGREGATION RESPECTIVELY.

measure	LA	HA	random-net	random-sa	random-pm	basic-rnn	lookback-rnn	attention-rnn	salami	pianomidi
r_{min}			-0.12 ± 0.04	-0.05 ± 0.03	-0.08 ± 0.05	-0.74 ± 0.35	-0.68 ± 0.3	-1.04 ± 0.24	-1.09 ± 0.16	$\mathbf{-1.11 \pm 0.11}$
r_{max}			4.2 ± 0.46	4.09 ± 0.22	4.1 ± 0.28	11.26 ± 10.03	11.17 ± 9.6	20.46 ± 14.32	$\mathbf{21.73 \pm 12.83}$	19.76 ± 9.0
hierarchy depth			0.08 ± 0.01	0.06 ± 0.02	0.07 ± 0.03	0.18 ± 0.08	0.16 ± 0.06	0.23 ± 0.07	0.25 ± 0.05	$\mathbf{0.28 \pm 0.05}$
number of splits		-	6.43 ± 1.07	5.21 ± 1.4	5.92 ± 1.28	9.27 ± 2.14	$\mathbf{9.83 \pm 1.5}$	8.18 ± 2.57	6.47 ± 2.32	6.53 ± 1.95
number of communities			$\mathbf{0.97 \pm 0.01}$	$\mathbf{0.97 \pm 0.0}$	$\mathbf{0.97 \pm 0.0}$	0.91 ± 0.1	0.92 ± 0.11	0.79 ± 0.17	0.72 ± 0.13	0.72 ± 0.11
singleton fragmentation			$\mathbf{0.74 \pm 0.0}$	$\mathbf{0.74 \pm 0.0}$	$\mathbf{0.74 \pm 0.0}$	0.7 ± 0.05	0.71 ± 0.04	0.66 ± 0.05	0.62 ± 0.04	0.62 ± 0.03
ns communities (number)		M	0.94 ± 0.03	$\mathbf{0.97 \pm 0.02}$	0.96 ± 0.03	0.68 ± 0.18	0.68 ± 0.13	0.54 ± 0.14	0.45 ± 0.13	0.45 ± 0.1
		SampEn	0.14 ± 0.02	$\mathbf{0.17 \pm 0.03}$	0.15 ± 0.02	0.11 ± 0.03	0.11 ± 0.02	0.09 ± 0.03	0.09 ± 0.02	0.09 ± 0.03
		SD	0.14 ± 0.07	0.07 ± 0.04	0.09 ± 0.05	0.35 ± 0.08	0.37 ± 0.06	$\mathbf{0.4 \pm 0.03}$	0.38 ± 0.04	$\mathbf{0.4 \pm 0.03}$
		CV	0.15 ± 0.08	0.07 ± 0.04	0.1 ± 0.06	0.59 ± 0.3	0.58 ± 0.21	0.79 ± 0.22	$\mathbf{0.92 \pm 0.22}$	$\mathbf{0.92 \pm 0.19}$
ns communities (size)	-	M	0.96 ± 0.03	$\mathbf{0.98 \pm 0.02}$	0.97 ± 0.02	0.76 ± 0.11	0.75 ± 0.09	0.69 ± 0.08	0.67 ± 0.06	0.66 ± 0.05
		SampEn	0.14 ± 0.02	$\mathbf{0.16 \pm 0.03}$	0.15 ± 0.02	0.09 ± 0.02	0.09 ± 0.02	0.09 ± 0.02	0.08 ± 0.02	0.08 ± 0.02
		SD	0.11 ± 0.08	0.04 ± 0.04	0.06 ± 0.05	0.31 ± 0.07	$\mathbf{0.34 \pm 0.06}$	$\mathbf{0.34 \pm 0.05}$	0.33 ± 0.03	0.33 ± 0.03
		CV	0.12 ± 0.09	0.04 ± 0.04	0.06 ± 0.06	0.43 ± 0.14	0.47 ± 0.11	0.49 ± 0.1	0.49 ± 0.07	$\mathbf{0.5 \pm 0.07}$
fragmentation imbalance	mean	M	0.13 ± 0.02	0.12 ± 0.02	0.12 ± 0.02	0.25 ± 0.1	0.22 ± 0.08	0.31 ± 0.08	$\mathbf{0.38 \pm 0.07}$	0.37 ± 0.06
		SampEn	0.56 ± 0.27	0.74 ± 0.38	0.67 ± 0.37	0.9 ± 0.5	0.82 ± 0.43	1.25 ± 0.53	$\mathbf{1.63 \pm 0.39}$	1.61 ± 0.39
		SD	0.17 ± 0.03	0.16 ± 0.03	0.17 ± 0.03	0.27 ± 0.07	0.26 ± 0.06	0.32 ± 0.05	$\mathbf{0.34 \pm 0.04}$	$\mathbf{0.34 \pm 0.03}$
		CV	1.38 ± 0.14	$\mathbf{1.41 \pm 0.14}$	$\mathbf{1.41 \pm 0.16}$	1.15 ± 0.2	1.22 ± 0.18	1.05 ± 0.16	0.9 ± 0.11	0.94 ± 0.11
	min	M	0.07 ± 0.03	0.06 ± 0.03	0.07 ± 0.03	0.14 ± 0.09	0.12 ± 0.07	0.2 ± 0.08	$\mathbf{0.26 \pm 0.08}$	0.25 ± 0.07
		SampEn	0.39 ± 0.15	0.48 ± 0.2	0.44 ± 0.19	0.38 ± 0.24	0.32 ± 0.18	0.53 ± 0.25	$\mathbf{0.74 \pm 0.27}$	0.71 ± 0.24
		SD	0.16 ± 0.04	0.15 ± 0.05	0.16 ± 0.05	0.26 ± 0.09	0.25 ± 0.08	0.32 ± 0.07	$\mathbf{0.36 \pm 0.05}$	$\mathbf{0.36 \pm 0.05}$
		CV	2.43 ± 0.47	$\mathbf{2.47 \pm 0.49}$	2.41 ± 0.4	2.28 ± 0.61	2.44 ± 0.58	1.85 ± 0.52	1.51 ± 0.32	1.5 ± 0.26
	max	M	0.22 ± 0.04	0.21 ± 0.04	0.22 ± 0.04	0.44 ± 0.13	0.41 ± 0.1	0.49 ± 0.1	$\mathbf{0.57 \pm 0.08}$	0.53 ± 0.08
		SampEn	0.54 ± 0.23	0.71 ± 0.37	0.66 ± 0.34	0.86 ± 0.39	0.87 ± 0.37	1.1 ± 0.34	$\mathbf{1.35 \pm 0.27}$	1.31 ± 0.24
		SD	0.25 ± 0.03	0.23 ± 0.04	0.25 ± 0.04	0.37 ± 0.04	0.37 ± 0.04	0.4 ± 0.03	$\mathbf{0.4 \pm 0.02}$	$\mathbf{0.4 \pm 0.02}$
		CV	$\mathbf{1.14 \pm 0.14}$	$\mathbf{1.14 \pm 0.12}$	$\mathbf{1.14 \pm 0.14}$	0.9 ± 0.18	0.94 ± 0.16	0.85 ± 0.15	0.72 ± 0.11	0.77 ± 0.11



Jacopo de Berardinis is a Postdoctoral Research Associate in Informatics at King's College London, and an Honorary Research Assistant at the University of Liverpool (Applied Music Research Lab). Previously, he received his doctoral degree in Machine Learning from the University of Manchester, and his master's degree in Computer Science from Reykjavik University (Iceland) and the University of Camerino (Italy). His main research interests revolve around the application of machine learning techniques to the field of MIR, with the goal of designing computational methods for the automatic analysis of music – serving the interests and needs of artists, musicologists, music psychologists and researchers. In particular, his work focuses on music structure analysis, predictive modelling of music, music emotion recognition, and automatic music composition. He is currently working for Polifonia, a European H2020 project aiming to bring together music, people, places and events from the sixteenth century to the modern day, by pioneering the first interconnected global database on the Web.

Angelo Cangelosi is Professor of Machine Learning and Robotics at the University of Manchester (UK). He is Turing Fellow at the Alan Turing Institute London, Visiting Professor at Hohai University and at Universita' Cattolica Milan, and Visiting Distinguished Fellow at AIST-AIRC Tokyo. His research interests are in developmental robotics, language grounding, human robot-interaction and trust, and robot companions for health and social care. Prof. Cangelosi has produced more than 300 scientific publications, had led many UK and international projects (e.g. THRIVE, EnTRUST, APRIL, BABEL, ROBOTDOC, ITALK) and has been general/bridging chair of numerous workshops and conferences including the IEEE ICDL-EpiRob Conferences. He is Editor of the journals Interaction Studies and IET Cognitive Computation and Systems, and in 2015 was Editor-in-Chief of IEEE Transactions on Autonomous Development. His latest book "Cognitive Robotics" (MIT Press), coedited with Minoru Asada, will be published in 2021. ORCID: 0000-0002-4709-2243

Eduardo Coutinho is a Senior Lecturer in Music Psychology at the Department of Music from the University of Liverpool. Before his appointment at Liverpool, he worked at the University of Augsburg, Imperial College London, Swiss Center for Affective Sciences and University of Sheffield. Coutinho received his diploma in Electrical Engineering and Computer Sciences from the University of Porto (Portugal, 2003), and his doctoral degree in Music Psychology and Computer Sciences from the University of Plymouth (UK, 2009). In 2013, he received the Knowledge Transfer Award from the National Center of Competence in Research in Affective Sciences, and in 2014 the Young Investigator Award from the International Neural Network Society. He develops his research in an interdisciplinary context often combining Music Psychology and Computer Sciences, and he has published significantly in peer-reviewed journals and conferences in both areas. His expertise is in the study of emotional expression, perception and induction through music, the study on music in everyday life and the automatic recognition of emotion in music and speech. Currently his work focuses on the development of music interventions in Healthcare.

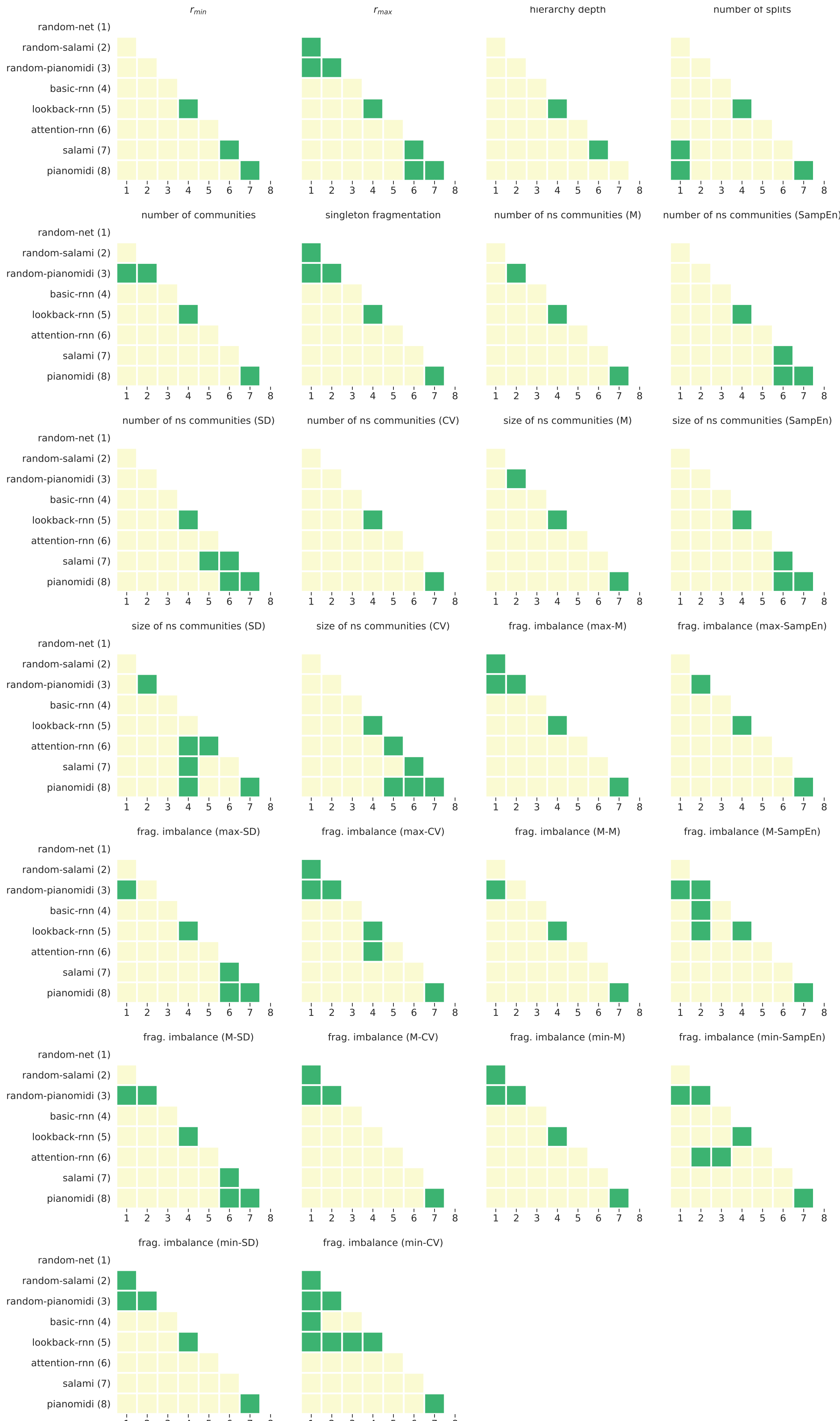


Fig. 7. Pairwise statistical analysis of the music subsets for each structural metric (yellow denotes statistical difference).