Analysis and Visualization of Musical Structure using Networks.

Alberto Alcalá-Alvarez, Pablo Padilla-Longoria

Institute for Applied Mathematics (IIMAS), UNAM, Mexico

Abstract

In this article, a framework for defining and analysing a family of graphs or networks from symbolic music information is discussed. Such graphs concern different types of elements, such as pitches, chords and rhythms., and the relations among them, and are built from quantitative or categorical data contained in digital music scores. They are helpful in visualizing musical features at once, thus leading to a computational tool for understanding the general structural elements of a music fragment.Data obtained from a digital score undergoes different analytical procedures from graph and network theory, such as computing their centrality measures and entropy, and detecting their communities. We analyze pieces of music coming from different styles, and compare some of our results with conclusions from traditional music analysis techniques.

Introduction

In the present work we discuss a framework for analysing symbolic music information, *i.e.,* considering digital transcriptions (*e.g.*, MIDI or MusicXML files) of the music fragment or piece in question, which is parsed (using the Music21 Python library) and translated as a set of graphs or networks (built using the NetworkX library in Python) identifying certain musical features of general interest. This procedure lets us picture and take into account the overall construction of the piece, by plotting and measuring the occurrence of different kinds of elements as well as relations among them. Due to our use of the Music21 Python library, which includes a large set of tools to deal with digital symbolic music formats such as MIDI and XML files, for the time being we constrain ourselves to music written in conventional western notation. That is, into a five line staff or pentagram, with the usual elements and signs such as measure bars and rhythmic values of half, quarter, eighth etc. However, this does not force us to exclusively focus on western academic music, as the analyzed data may include rhythms expressed as durations in miliseconds and pitches expressed in cents. This means that, in principle, we may apply the present framework to microtonal music and unmeasured music, given that we have it properly codified in a symbolic digital file. We believe this method may be extended to deal with electronic textures and continuous sounds in general.

Several approaches applying graph and network theory for music analysis have been published in recent years, for example Szeto and Wong [2006], Walton [2010], Ren [2014], Buongiorno Nardelli [2023], Brown and George [2023]. Certain aspects in our scope are similar to some of those discussed in such papers, yet it constitutes a parallel proposal. For example, we aim to analyse music beyond a particular musical language or genre, and trying to bring together both vertical and horizontal aspects of music, with the possibility of including diverse types of musical objects. Also we incorporate a dynamic approach to the evolution of families of graphs associated to music data. We focus on the possibilities of computational analysis and visualization that could be useful for contemporary concert music and non-Western repertoires, while seeking consistency with tonal, modal and serial music analysis.

The present article is meant to showcase one possible general working pipeline for modeling symbolic music information with graphs, and is in no way an exhaustive. Such graphs may contain elements of very diverse nature, and for reasons of space, we demonstrate the foundations of our proposal focusing mainly on pitch information (we consider a multigraph including duration values).

The present article is broadly structured in the following way:

1. In the first section, we develop some of the basic mathematical concepts and tools;

2. Next, we describe the general methodology we follow.

3. In the third section we define and illustrate several graphs from symbolic music data and discuss some of the metrics, properties and other associated objects we consider for each graph; we also incorporate the dynamic dimension of our proposal, through the use of sliding time windows and time series.

4. Finally we discuss the analysis of some graphs associated to music fragments from three different styles and languages, and present our conclusions.

Contents

1 Preliminaries.

1.1 Graphs and networks.

A **graph** G is a mathematical object consisting of two sets:

- A set $V(G) = V$ of vertices or nodes, which in our setting will consist of musical objects such as pitch classes, chords and rhythms (durations).

- A set $E(G) = E$ of edges, which are pairs of vertices, possibly ordered, in which case the order shows a direction from one vertex to another. In our setting, edges will represent the concurrence of musical elements, for example a chord-node will be connected to all pitch-class-nodes corresponding to the notes it contains, as well as to all the rhythm-nodes with the duration values the chord takes along the fragment.

The **order** of a graph is its number of vertices, and its **size** is its number of edges.

A **network** is a kind of graph whose edges are all directed (also called arrows) and elements (both nodes and edges) have an extra attribute associated to them, which is generically called **weight**. In different contexts, this weight may stand for size (when talking about nodes), capacity or speed (when speaking of arrows) of elements. The **degree** of a vertex v, $\deg(v)$, is the number of edges containing v, and if edges are directed then we may speak of the **in-degree** and **out-degree** of v, respectively denoted $\deg_{in}(v)$, $\deg_{out}(v)$.

Definition 1.1. A **cluster** in a graph is an induced subgraph with dense inner connections and sparse interconnections.

The **density** of a graph is the proportion between the size of a graph and the maximum possible size of a graph of the same order (for undirected graphs, $\begin{pmatrix} |V| \\ 2 \end{pmatrix}$, the size of the complete graph of order $|V|$).

In this work, we present graphs whose nodes are either pitch classes, represented by integers from 0 to 11, chords, written as tuples of pitch classes without repetition, and rhythms, expressed in quarter-length-duration, 'qL' for short. These nodes are tagged as a tuple consisting of a decimal number together with the legend 'qL'. No more than one edge between two nodes in our graphs will be allowed. Instead, we will add a weight to each of them, according to the number of occurrences of the corresponding relations in the analyzed fragment. We will also add another weight to the nodes, counting the total duration along the fragment of the corresponding element in the score, and will not be considering edges from a node to itself (called loops).

1.1.1 Node Centrality.

Vertices in a graph may be considered to be important under different criteria. One that is particularly useful is to measure *how connected* each vertex is to other vertices, that is, how many edges are incident with that vertex, or how many *neighbors* it has. This leads to defining different measures called **centralitiy measures**, which numerically describe the number of connections of a vertex. When talking about pitches or chords, a node with high centrality is a pitch or chord that leads to many others, and so may be a pivot chord or a sort of "tonic". On the other hand, nodes with a low centrality can be taken as passing notes or chords, or perhaps ornamental elements.

Out of the different notions and definitions of centrality measures, we consider **degree centrality** and **eigenvector centrality**.

- Degree centrality is the normalised degree of vertices, with respect to the maximum possible degree of a vertex ($n-1$ in a graph on n vertices). That is, for a vertex v in a graph with n vertices, its **degree centrality** is

$$\delta(v) = \frac{\deg(v)}{n-1}\,.$$

- **Eigenvector centrality** takes advantage of the fact that for non-negative matrices (such as the adjacency matrix of a graph or network) the Perron-Frobenius Theorem (see theorem **??** in the **??**) gives exactly one eigenvector, called the **leading eigenvector**, whose entries are all non-negative (see for example Newman [2018]). Moreover, the leading eigenvector corresponds to the largest eigenvalue of the matrix. Thus, considering a graph with ordered set of vertices $V = \{v_1, v_2, \dots, v_n\}$, the **eigenvector centrality** of v_i is the i–th coordinate of the leading eigenvector of the adjacency matrix of the graph. That is, it is the i–th coordinate of the only non-negative solution to the equation

$$A\bar{\mathbf{x}} = \lambda\bar{\mathbf{x}}\,,$$

where A is the adjacency matrix and λ its largest eigenvalue.

1.1.2 Entropy.

Entropy is a term used in different fields to denote some notion or measure of disorder (randomness) of a system. In Information Theory, which is the scope we adopt in this exposition, the entropy of a system is related to Probability Theory, and is inversely proportional to the "amount of information" we can get out of the *message*. Whereas the information content associated to the event is the logarithm of the inverse of its probability of occurrence. Equivalently, entropy is inversely proportional to how easy it is to guess the state of the system by "asking yes-or-no-questions" (see Information Theory).

Mathematically, entropy is a number which captures at once the weighted probabilities of occurrence of the elements in the system. As in the case of communities, there is more than one way to define a numerical definition of entropy, though all of them behave more or less alike.

One of the usual ways to define the **entropy** of a set $X = \{x_1, x_2, \dots, x_n\}$ with probability distribution $\{p(x_1) = p_1, p(x_2) = p_2, \dots, p(x_n) = p_n\}$, where $p(x_i)$ stands for the probability of occurrence of the event or element x_i, is the **Shannon formula**:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i)\,.$$

This number is always bounded below by 0 and above by $\log_2(n)$. These bounds correspond, respectively, to the case where all probabilites are 0 except for one which is equal to 1 , and the case where all elements $x_1, x_2, \ldots, x_n$ are equiprobable, each with probability equal to $1/n$.

In the first situation we get

$$H(X) = -1ï¿œ\log_2(1) = -\log_2(1) = 0$$

and in the second case:

$$H(X) = -\sum_{i=1}^{n} \frac{1}{n}\log_2(\frac{1}{n}) = \sum_{i=1}^{n} \frac{1}{n}\log_2(n) = \log_2(n)\,.$$

Such extreme situations may be interpreted, correspondingly, as the case with most information (when $H(X) = 0$, the state of the system in question is known with absolute certainty) and the one with the least possible amount of information (when all events or elements are equally likely to occur, we have less clues about the message or system).

This statistical definition of entropy has been applied in music analysis, particularly to the study of harmonic progressions in W.A. Mozart's work (see Nielsen [2009]).

1.1.3 Entropy in networks.

In the case of graphs, there are also several ways to define entropy (the reader may refer to Dehmer and Mowshowitz [2011] for a survey on different entropy measures). One is to associate weights to nodes and/or edges, and apply the Shannon formula to their distribution. Such weights may be given by some graph-driven indicator, such as the number of neighbors or some other centrality measure for nodes.

Another definition worth considering is the **von Neumann entropy**, which is defined as follows:

Let A be the symmetric weight matrix of an undirected weighted graph G with vertex set $V = \{v_1, \dots, v_n\}$. Let $D = \mathrm{diag}(d_1, \dots, d_n)$ where $d_i = \deg(v_i)$. The **Laplacian matrix of** G is defined as

$$L = D - A,$$

whose set of eigenvalues $\lambda_1, \dots, \lambda_n$ is called the **Laplacian spectrum**.

The **von Neumann entropy** of G is defined as

$$H_{\mathrm{vN}}(G) = -\sum_{i=1}^{n} \frac{\lambda_i}{vol(G)} \log\left(\frac{\lambda_i}{vol(G)}\right),$$

where

$$vol(G) = \sum_{i=1}^{n} \lambda_i = tr(L).$$

Thus, the von Neumann entropy of a graph is the Shannon entropy for the distribution of the eigenvalues of its associated Laplacian matrix.

Given a cluster K in a graph G, the probability of a vertex $v \in K$ to have inner connections in K is given by

$$p_i(v) \coloneqq \frac{N_K(v)}{N(v)},$$

where $N_K(v)$ denotes the number of vertices in K which are neighbors of v, and $N(v)$ denotes the total number of neighbors of v in G.

On the other hand, the probability of v having outer connections is naturally defined as

$$p_o(v) \coloneqq 1 - p_i(v).$$

Finally, the **graph entropy** (see, for example Kenley and Cho [2011]) of a vertex $v \in K$

is given by

$$H(v) := -p_i(v)\log_2 p_i(v) - p_o(v)\log_2 p_o(v),$$

and the **graph entropy of** G by

$$H(G) := \sum_{v\in V(G)} H(v).$$

Notice that this quantity does depend on the number of vertices, hence comparing graphs of different order with this measure might be misleading. Hence, we will rather work with the average per node, given by:

$$H(G) := \frac{1}{|V(G)|}\sum_{v\in V(G)} H(v).$$

With this definition of entropy, we obtain a measure normalised with respect to the number of nodes in a graph, which reflects its structure *as a graph*.

1.1.4 Communities.

In a graph, communities are subgraphs which are highly connected. They consist of subsets of elements that have strong relationships among themselves, and so they outline parts of the graph which are structurally important. Though there is not a universally accepted definition for communities in a graph, we can say they are subgraphs whose vertices are densely connected among themselves, but have few connections with nodes in other communities.

There are different algorithms to establish or detect communities in a graph. In this work we consider the Clauset-Newman-Moore greedy modularity maximization algorithm (see Clauset et al. [2004]), included in the Networkx package (Hagberg et al. [2008]). This algorithm is based on merging communities that maximise the change in modularity.

Modularity is a statistical measure of the possibility of finding community structure in a graph. Given a set of communities, it quantifies the density of connections in each of them, against the expected number of connections. A higher modularity indicates there are in fact communities which are very densely connected within, yet not with other communities. Hence, modularity is inversely proportional to the density of a graph (a higher density implies a lesser chance of finding subgraphs with very few connections with other subgraphs). Formally, it is calculated as

$$Q(\mathfrak{C}) = \frac{1}{2|E|} \sum_{u,v \in V} (A_{u,v} - \frac{\delta_u \delta_v}{2|E|}) \delta_{C_u,C_v} \,,$$

where $\mathfrak{C}$ is a set of communities, $A_{u,v}$ is the (u,v)-entry of the adjacency matrix A, δ_v is the degree of v, and δ_{C_u,C_v} is the Kronecker delta for the communities C_u, C_v, the communities of u and v, respectively. This can be also expressed as

$$Q(\mathfrak{C}) = \sum_{C \in \mathfrak{C}} \left(\frac{m_C}{|E|} - \theta \left(\frac{\delta_C}{2|E|} \right)^2 \right) ,$$

where C ranges over the communities of the network, m_C is the number of intra-community links, δ_C is the sum of degrees of nodes in community C, and θ is the resolution parameter, which establishes a tradeoff between intra-community and inter-community edges.

Starting with every node as the single member of a community in a given network, the Clauset-Newman-Moore greedy algorithm iteratively merges communities seeking to maximise the modularity in every step. Its outcome is the “most relevant” partition of a graph into clusters. Yields high-modularity communities.

In contrast with the described method for identifying communities in a network, there is a graph entropy based algorithm (see section 1.1.3), consisting of clustering nodes together to minimise entropy. Beginning with a seed cluster, iteratively minimizing graph entropy yields another clustering algorithm for community detection: neighbors of each node are added or deleted if such an operation results in a lower entropy of the cluster. The process stops when it becomes no longer possible to get a lower entropy in each obtained cluster. The result is a set of node clusters with minimal entropy.

1.2 Discrete time series.

A **discrete time series** is a sequence of values from observations made at certain points in time (usually, evenly distributed). The main attributes for describing time series are:

- Trends.
- Seasonality.
- Presence and behaviour of irregular fluctuations.

In this work we consider only finite discrete time series, each containing a series of measurements of a certain metric of the graphs and networks associated with a sequence of fragments covering the score under analysis.

Given two discrete time series, possibly of different length, **dynamic time warping** (DTW; see, for example Müller [2007]) is a usual tool for comparing them. DTW consists of finding an optimal strictly index-increasing matching of points in both series (in which each point in a series may be connected to more than one element of the other). Optimal here means with minimal cost, where the cost is the sum of all absolute differences between matched points. This yields a notion of closeness between time series. However, it is not strictly a distance or metric, since the triangle inequality cannot be guaranteed.

2 Methodology

As we mentioned earlier, our proposal seeks, through the use of graphs, to model and describe diverse music elements and their relations throughout a given score (more generally, a fragment of a transcription), from an agnostic data-focused point of view. We relie on the use of computational tools for materialising our proposal, and hence aim to work with digital music notation files. It is worth mentioning that the availability and suitability of such a file involves a whole set of considerations and difficulties. On the other hand, if no transcription is already available, it is possible to merely consider a list of chronologically ordered music events, encoded as some suitable type of digital object.

The exposition presented here is not meant to be exhaustive, but rather illustrative of a general procedure applicable to different types of musical objects and notions. The proposed method is summarised in the following steps:

1. Parse data in a digital score. For this purpose we have used the music21 Python library (Cuthbert and Ariza [2010]), which allows for the use of several file formats. For the present work we have used MIDI and MusicXML files.

2. Segment the score in intervals of equal time length (given in number of bars or seconds). We consider different fixed lengths and frequencies of such time windows, which seem appropriate for capturing meaningful features.

3. For each time window, extract objects encoded in the score, such as pitch classes, vertical events (which we also refer to as chords), rhythm values, etc.

4. Define graphs and networks containing such objects as nodes, and edges which represent sequential or simultaneous occurrence of elements.

5. For the resulting graphs, compute several metrics and perform analysis algorithms: we consider mainly centrality measures, entropies and community detection. Repeat-

ing this for our whole sequence of time windows along the score yields a family of discrete time series.

6. Plotting and analysis of the resulting time series throughout the whole score.

7. Discussion of musical insights which can be drawn from this network-based analysis, mainly concerning style and form.

3 Some graphs associated with sequences of musical events.

For building and analyzing graphs from musical events, we run a Python script using two main libraries: Music21, to parse and manipulate digital music notation files, and NetworkX (Hagberg et al. [2008]). The code we have so far developed also incorporates some standard libraries for mathematical purposes, such as Numpy and Matplotlib.

After parsing a file in a symbollic digital format (such as .midi, .xml, .mxl, .abc, etc.), we focus on getting all the chords in a fragment of the score, together with their durations, and so we get a list of ordered pairs whose first entry is a tuple of pitch classes (without repetitions; that is, a pitch class appearing in two or more different octaves or instruments in the score will appear only once in the resulting tuple), and whose second entry is a decimal number which tells us the duration of that chord in quarters ('qL'=quarter length). We get one of these pairs for each "event" in the transcrpition or score; that is, one chord is added to the list everytime there is a new sound being registered in any of the staves making up the score. Next, we shorten that list by deleting all the repetead chords, adding up the time of the deleted repetitions, and also keeping count of the number of repetitions of each chord in the whole fragment being analyzed. In this way, we end up having a list of chords, all different, together with their cumulative durations and number of ocurrences. Later we will use these two numbers associated with each chord to indicate how "dense" a chord or pitch class is, by taking a node's weight to be its total duration, which will be proportional to its size when plotted, and its number of repetitions as the degree of transparency of its color (this gives us some information about texture). We will proceed analogously with the rhtyhms obtained in the original list (before adding up all durations of each chord), and asigning each of them a weight, proportional to its total number of occurrences. In this work we focus on pitch information.

3.1 Pitch-chord-rhythm (p-c-r) graph.

After getting this reduced list of chords expressed as tuples of pitch classes together with their cumulative durations and their respective number of repetitions, we proceed to build our first graph, which we call the **pitch-chord-rhythm (p-c-r) graph**, which encodes some meaningful relations among pitches of vertical events and the time intervals between them (the duration of each vertical event, delimited by the appearance, prolongation or removal of a note). It is a multipartite graph in which we have three types of nodes: pitch classes, chords (vectors of pitch classes) and lengths (decimal expression of duration in quarter note figures). When plotting the p-c-r graph we identify different types of nodes with different colors: pitch classes in turquoise, tuples of simultaneous events (which we also refer to as *chords*) in red, and rhythmic values in blue. We connect these nodes in the following way:

- Each chord-node is connected (or adjacent) to all of the pitch-class-nodes corresponding to its pitch classes, and so every pitch-class-node is connected to the nodes of all the chords it belongs to.

- Each chord-node is connected to the rhythm-nodes of all the durations with which it appears, and so every rhythm-node is connected to the nodes standing for all the chords played with that rhythm.

Note that we could also connect all pitch-class-nodes to rhythm-nodes, but we leave this out for the sake of clarity, focusing more on durations and harmony (not melody) in this part of the analysis (later on we shall incorporate a "more melodic" graph). We also consider a **p-c-i-r graph**, in which the "i" stands for *interval classes* and interval-nodes are connected to chords (in normal form) in which consecutive pitches delimit the corresponding interval.

In general, we can consider adding to our graph other kinds of nodes, coming from any other attribute of musical objects codified in the parsed file, like dynamics or instrumentation.

Nodes in the multipartite p-c-r graph are assigned weights in two different ways: by its number of occurrences along the given fragment (*i.e.*, its frequency), and by the sum of all its duration values along the fragment. In plots we respectively depict such attributes as the size of the node and the level of transparency of its color (see, for example, Figure 3.1). In the case of rhythm-nodes we only consider one weight: its number of occurrences.

Example 3.1. To exemplify the p-c-r graph, we take a well-know fragment of tonal classical music: the first eight measures of J. S. Bach's Contrapunctus I from The Art of Fugue BWV 1080.

3.2 Analysis of the p-c-r graph.

The metrics considered below for a graph, some of them with very similar behavior, measure in some way the distribution of information within the p-c-r graph associated with a music fragment. We consider different definitions of entropy, as well as centrality measure, density, modularity, number of communities, and average clustering. In our p-c-r graphs, this last metric, average clustering, becomes irrelevant (being $= 0$ always), since the graph is all connected.

As we saw in Figure 3.1, taking the p-c-r graph of a whole fragment we get a static map of all the events (tuples of pitch classes) in the fragment at once, together with their components (pitch classes and duration in this case). To incorporate in our analysis

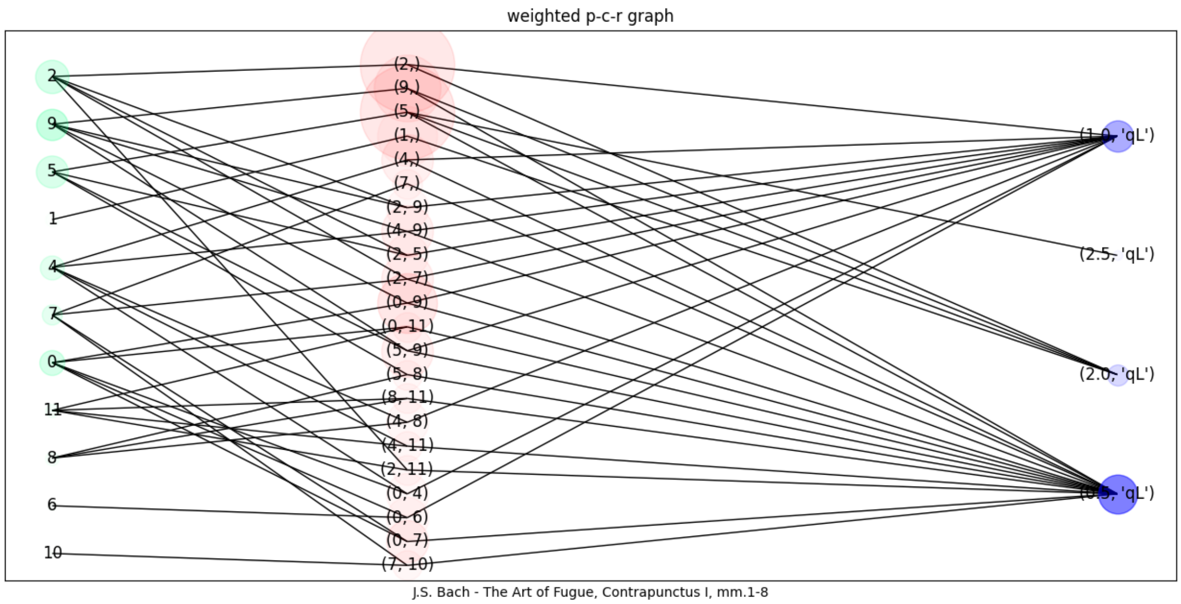

Figure 3.1: p-c-r graph for measures 1–8 of J. S. Bach's Contrapunctus I from The Art of Fugue BWV 1080.

the occurrence of events along a timeline, we consider moving time windows (of different sizes and regularities) which cover the whole fragment being analised. These windows are intervals of time of a fixed length which move along the score, covering the entire fragment, with or without overlapping. In this paper we consider time windows determined by a fixed number of bars taken every certain other fixed number of bars. These may be seen as an interval of bars of a certain radius around a the central barline of each window (for example, a window comprising mm. 1-8 of a score may be seen as an interval of radius 4 bars around barline 5.)[1].

We choose a fixed length for our time windows and displace them onward along the score in question, with a certain regularity, that is moving their center by a fixed distance at each stage. For each interval and center in this process we obtain a p-c-r graph, and obtain different metrics associated with it. This leads us to considering the discrete time series

[1]We assume for the time being that the scores we work with are divided in measures (though the music may not be strictly *metric*), and work upon subsequences of measures that cover a whole score (which may be either a complete piece of music, a complete segment or section of a larger work, or a fragment of a score).

determined by such numbers. We plot these as polygonal curves, which together form an ECG of the fragment being analyzed. Combining ECGs for time windows of different sizes and regularities we can get a more complete scheme of how information on our musical parameters is distributed, according to the associated p-c-r graph.

Of course one immediate issue when running this algorithm is how to choose the length and regularity of time windows. About this we can say the following: in order for this technique to be useful, the information grasped by each window must be neither too much nor too little, whatever that may mean in each musical context. Windows which are too narrow result too particular and yield too many points of measurement. On the other hand, windows which are too large imply all information is mixed and so measurements are less meaningful. Intuitively, our sliding window should capture enough information to build meaningful graphs that yield relevant and distinguishable measurements. Thus, the length of windows should be compatible with the average length of phrases, periods or other groupings of music elements in a specific type of music (for example, windows 4, 8 or 16 bars long seem reasonable for most music from the Common Practice Period).

The idea of a fixed length and fixed step for the moving time windows comes from considering that we may not know in advance how music is generally structured in a given sample. Also, meaningful groups of music elements (such as motives, phrases, etc.) are not necessarily grouped in sequences with the exact same length throughout a piece or fragment. Ideally, we would have a window englobing each of such units. This could be the result of fully determining the segmentation of the score into motives, phrases, etc., or establishing an algorithmic procedure for varying the lengths and positions of event windows. In Buongiorno Nardelli [2023], the author proposes the use of a change point detection algorithm from signal analysis in order to segment a score. In a future work we project to develop a segmentation algorithm proposal based on graph entropy or some other graph descriptor.

3.2.1 Communities of the p-c-r graph.

Communities in a p-c-r graph may include different types of nodes: chords, pitch classes, durations, interval classes, etc. Such music elements appear to "work together tightly", which means they are consistently coincident. It thus make sense, and has been indeed observed in samples, that communities of p-c-r graphs associated with tonal or modal music reflect the stronger harmonic regions and relations. It is usual to have communities "leaded" by a few pitch classes which group several chords around them. In the tonal context, these chords are mainly chords containing such pitch classes, together with passing notes or other ornaments.

Example 3.2. Following example 3.1, we show the five communities in p-c-r graph from Figure 3.1, associated with mm. 1–8 of Contrapunctus I from The Art of Fugue BWV 1080. See Figure 3.2.

The first community of the p-c-r graph clearly corresponds to the D minor chord (tonic) and some voice leading within and around it. Community #2 of this graph shows harmonic constructions around the third C-E. It includes also the fifth A-E, so it seems to reflect the A minor triad (minor V chord) and the tritone C-F$\sharp$, which leads to the subdominant chord (G minor). Besides this, the presence of the rhythm node tells us that movement involving these chords occurs mainly in quarter notes. The third community detected by the algorithm features the pitch-class-nodes 7 and 10, corresponding to G and B$\flat$, respectively. Looking at the node $(2, 7)$ we can relate this subgraph with the G minor chord (subdominant). Also, we notice the nodes in this community are highly coincident with the half-quarter rhythm. Next, community #4 is easily seen to correspond to the E Major chord, which is in this case the dominant of the dominant. The fifth and last of the communities in the p-c-r graph gives us more information about rhythms of specific pitch classes: D, F and C$\sharp$.

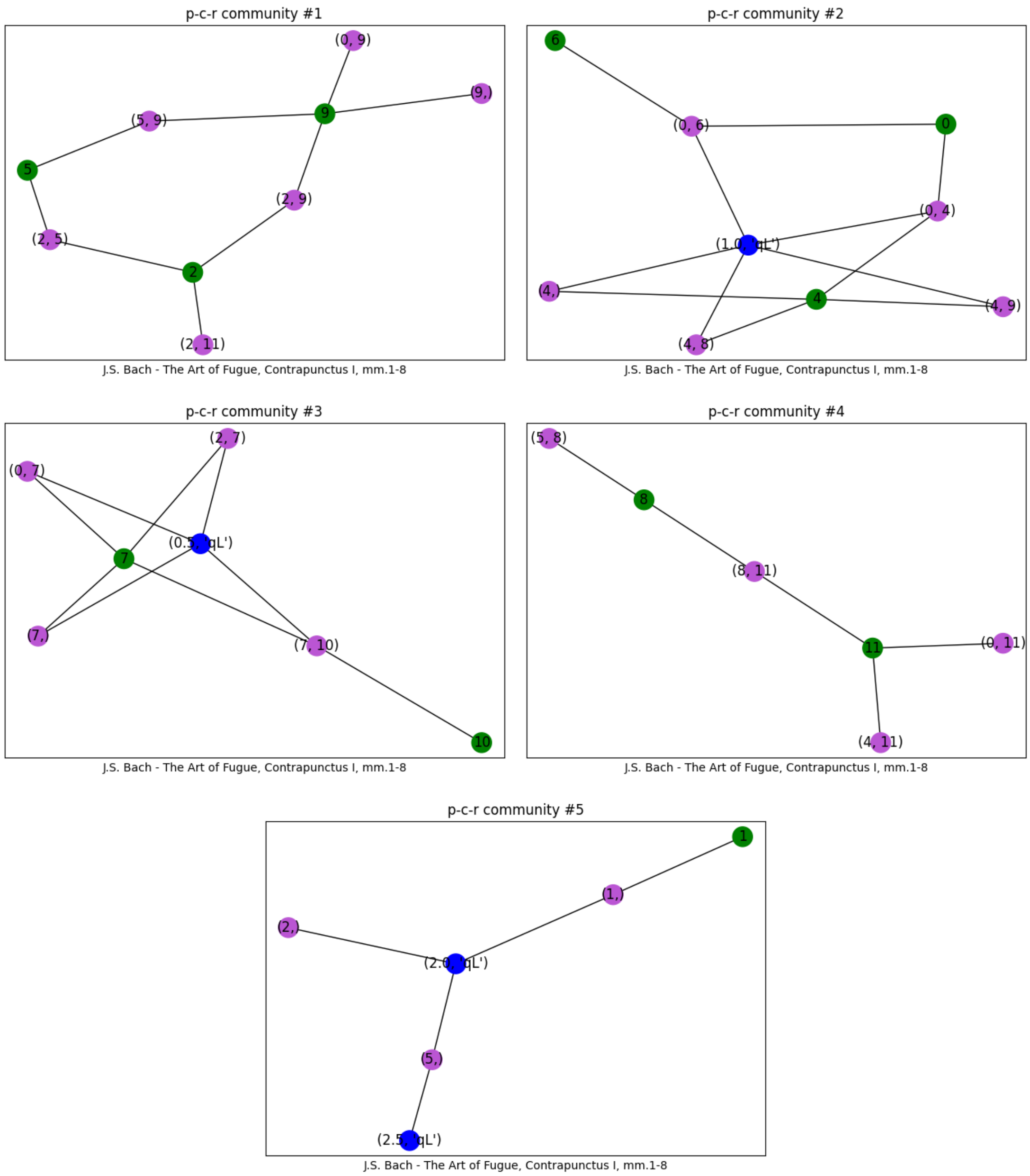

Figure 3.2: The five communities in the p-c-r graph for mm. 1–8 of J. S. Bach's Contrapunctus I from The Art Of Fugue BWV 1080.

3.2.2 Entropy of the p-c-r graph.

For our purpose, we consider different ways of defining the entropy of a graph, among the many possibilities. All of them tell us how likely it is to establish a "predominance" of certain nodes or edges (because they measure, more or less broadly, how sparse or condensed edges and weights in a graph are). Musically we may interpret this as a measure of the possibility to establish hierarchical relations among musical objects such as chords, rhythms, intervals or any other musical parameter to be found codified in the file being analyzed.

If the entropy of a piece of music is low (close to 0), it is reasonable that we may establish the particular "mode" or "scale" it is in, as well as hierarchies of both chords and rhythms. Moreover, this same principle may be applied to find out about the relative importance of sequences of pitches, chords, rhythms, etc. On the other hand, a higher entropy value means there is a more uniform distribution of musical elements. Thus, we expect a higher entropy in time windows which include a noticeable change in texture or harmony, which may lead to identifying tension/release passages, as well as a potential segmentation of the score.

We compute the Shannon entropy of pitches, chords and rhythms separately, taking into account different probability distributions, based on three different factors: total duration (weight) of nodes, number of occurrences, degree centrality (the proportion of how many connections a node has, with respect to the total number of edges in the graph), and Eigenvector centrality. We also compute the von Neumann entropy of the resulting graph. As we will see in the examples, several of these entropy measures share an almost parallel behavior, being most discordant when calculated for pitch classes. We will discuss this in depth in the "Results" section (section 4).

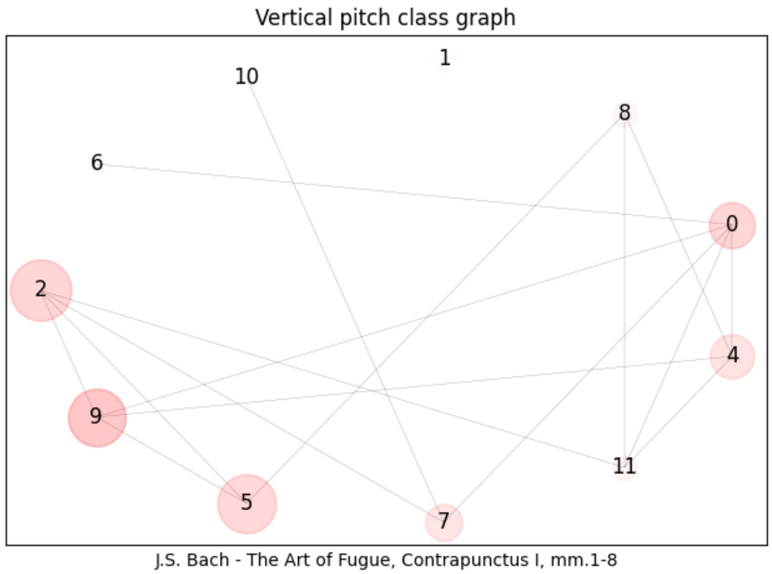

Figure 3.3: Vertical pitch class graph for mm. 1–8 of J. S. Bach's Contrapunctus I from The Art Of Fugue BWV 1080.

3.3 Pitch and chord graphs associated with sequences of events.

We now discuss some other graphs one may also associate with a music fragment, which model concurrences and sequences of specific music elements. As for the p-c-r graph, in all of them the size and degree of transparency are proportional to the number of occurrences and total cumulative duration of pitch classes. We keep on exemplifying the defined graphs with the initial fragment of Contrapunctus I from The Art of Fugue BWV 1080 (see Example 3.1).

3.3.1 Vertical pitch class graph.

In this graph, nodes are pitch classes, and edges describe vertical coincidences of notes in the score. That is, two pitch classes are connected if they appear together in some event from the sequence considered. The corresponding plot for our basic example is shown in Figure 3.3.

3.3.2 Horizontal pitch class graph.

A directed graph whose nodes are picth classes, in which arrows represent sequence in time. Given a sequence of vertical events there is one arrow from each pitch class in an event to every pitch class contained in the next event (see Figure 3.4.). Thus, paths in this network do not strictly correspond to melodies or harmonic/contrapuntal voices, but rather show pitches as elements of events which belong to events followed by events containing certain other pitches.

This model for horizontal pitch relations does not assume any particular voice leading scenario. Two consecutive events may contain different numbers of pitch classes, and in passing from one to another we might encounter voice crossings, unexpected leaps, or a melody "jumping" between registers and/or staves (*e.g.*, timbre melody, named *klangfarbenmelodie* in the 2nd. Vienese School). In order to analyze voice leading computationally, it is necessary to establish how to locate and define melodic movement, which is not univocally defined in general. For example, keeping track of voice-leading in a post-tonal setting can be tricky. Hence the choice of not assuming any general voice-leading criteria, and considering every possible horizontal relations among pitches from consecutive events.

3.3.3 Event or chord sequence graph.

In this network we merely represent the sequence of vertical events (*chords*), considering a weight of the arrow, given by the number of occurrences of the corresponding event progression. Cycles in this graph represent harmonic cycles which are *feasible* according to chord progressions score. That is to say, they can be considered as valid harmonic paths. It is not always the case that these cycles appear as an actual progression in the score, since it may happen that such chord connections do not follow each other sequentially in the score. Nevertheless, we can grasp this by looking at smaller time windows and keeping

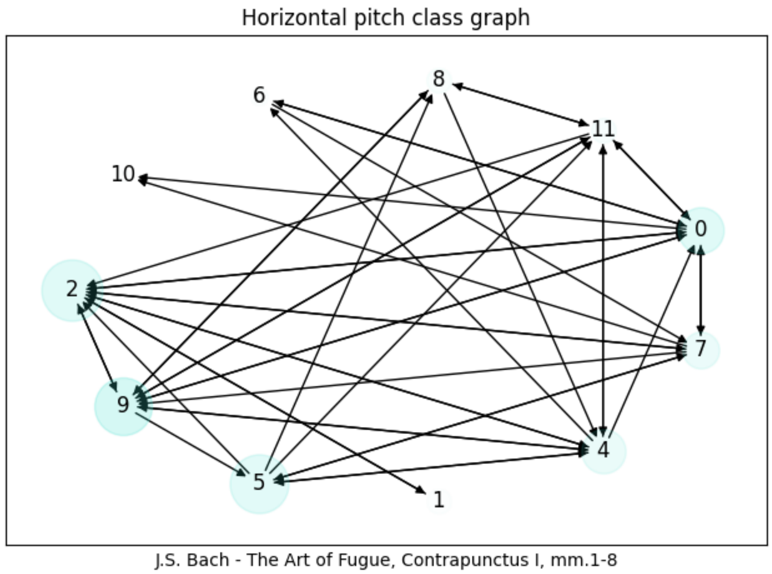

Figure 3.4: Horizontal pitch class graph for mm. 1–8 of J. S. Bach's Contrapunctus I from The Art Of Fugue BWV 1080.

track of how cycles arise. The chord graph for our example is depicted in Figure 3.5. This graph is considered also in Buongiorno Nardelli [2023], where it is then considered itself as a time series.

These are just examples of graphs and networks which can be universally associated with a fragment of any music score, each modeling relations within a specific aspect of music. Further, we may consider graphs encoding the relations between changes in rhythm, instrumentation, dynamics or texture, which we expect exploit, in future work, under the same approach here exposed.

3.4 Time series and ECG plots from metrics of associated networks.

For each length and frequency of our moving time windows, the values for each metric (centralities, entropies, etc.) of a graph associated to the corresponding music fragment

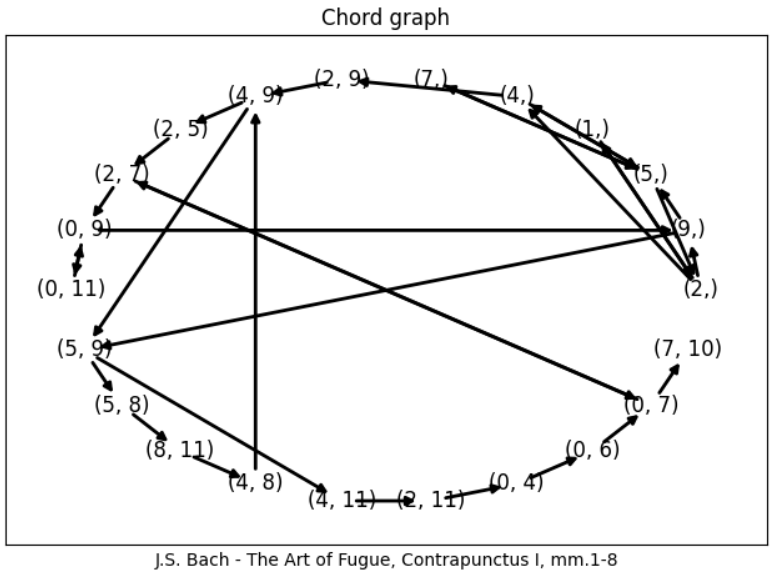

Figure 3.5: Event sequence graph for mm. 1–8 of J. S. Bach's Contrapunctus I from The Art Of Fugue BWV 1080.

yield a discrete time series. Given a length and frequency for time windows covering a score, plotting together all of these time series for the p-c-r graphs associated with the given music fragment, results in an ECG type of plot (see Figures 3.6–3.10). As we said before, a common way of comparing time series is dynamic time warping, which we will apply to several samples in the next section.

Following our example from The Art of Fugue (Contrapunctus I) we briefly review the ECG type plots obtained from time series of several metrics, for different time window sizes and frequencies. The metrics computed here are:

1. Shannon entropy of notes according to their accumulated duration throughout the fragment.

2. Shannon entropy of notes according to their number of occurrences in the fragment.

3. Shannon entropy of chords (pitch class vectors) according to their accumulated duration throughout the fragment.

4. Shannon entropy of chords according to their number of occurrences in the fragment.

5. Shannon entropy of duration values according to their number of occurrences in the fragment.

6. Shannon entropy of the p-c-i-r graph according to the degree centrality of nodes.

7. Shannon entropy of the p-c-i-r graph according to the eigenvector centrality of nodes.

8. Von Neumann entropy of the p-c-i-r graph.

9. Density of the p-c-i-r graph.

10. Number of communities of the p-c-i-r graph.

11. Modularity of the p-c-i-r graph.

The first five entropies computed are not graph-specific, but rather the Shannon entropy of the distributions of durations and frequency of elements, which are viewed as weights of nodes in the graphs we have described. The rest of descriptors describe graph-specific properties of the p-c-i-r graph. This lets us keep track of the evolution of entropy for each music element considered (pitch classes, pitch class vectors and duration values), while looking at how the entropy of the whole multipartite graph as well as its communities, density and modularity change.

ECG plots can be interpreted as a map of the whole score that shows the change in variety of several types of elements (in this case, duration values of events, chords, pitch classes, communities, etc.).

Example 3.3.

We can make the following general remarks about all of the above plots:

- The values of entropy calculated with respect to degree centrality and eigenvector centrality are very close to the von Neumann entropy of the p-c-r graph, usually with the former being bounded by the other two.

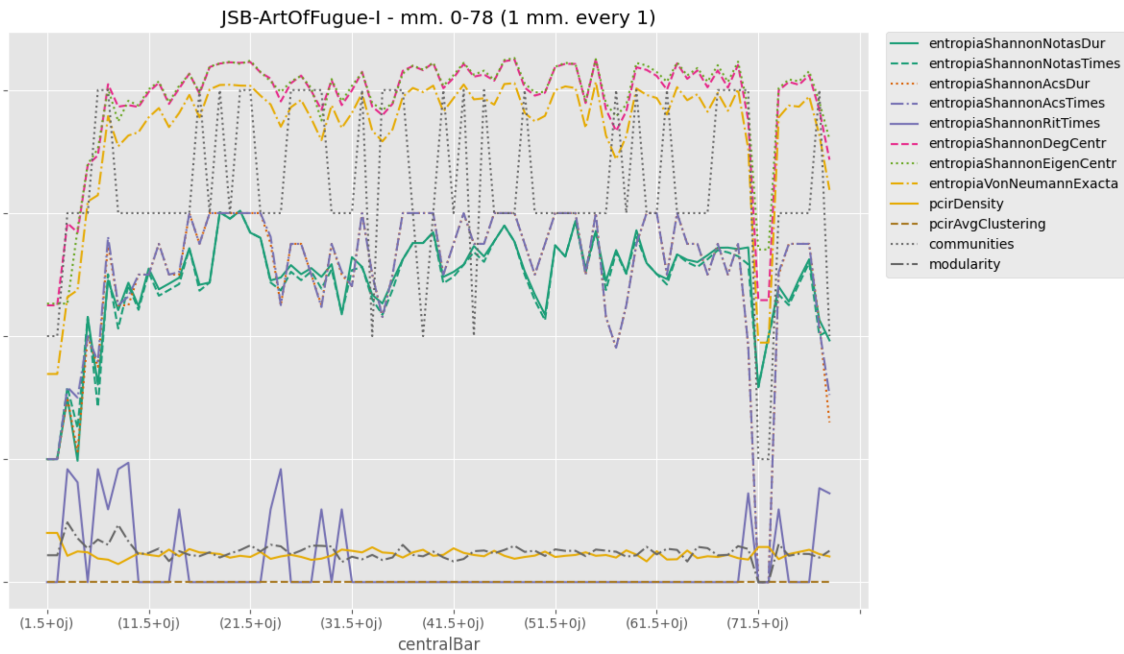

Figure 3.6: ECGs for metrics from the p-c-r graph for mm. 1–8 of J. S. Bach's Contrapunctus I from The Art Of Fugue BWV 1080, taking time windows of length 1 measure, without overlapping.

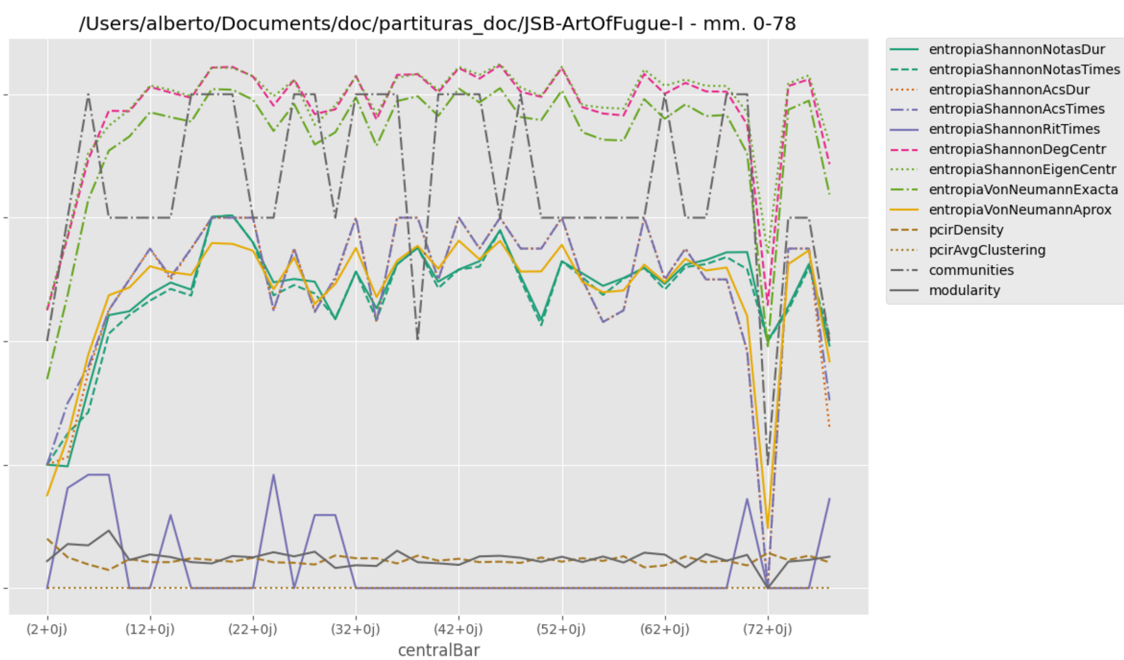

Figure 3.7: ECGs for metrics from the p-c-r graph for mm. 1–8 of J. S. Bach's Contrapunctus I from The Art Of Fugue BWV 1080, taking time windows of length 2 mm., without overlapping.

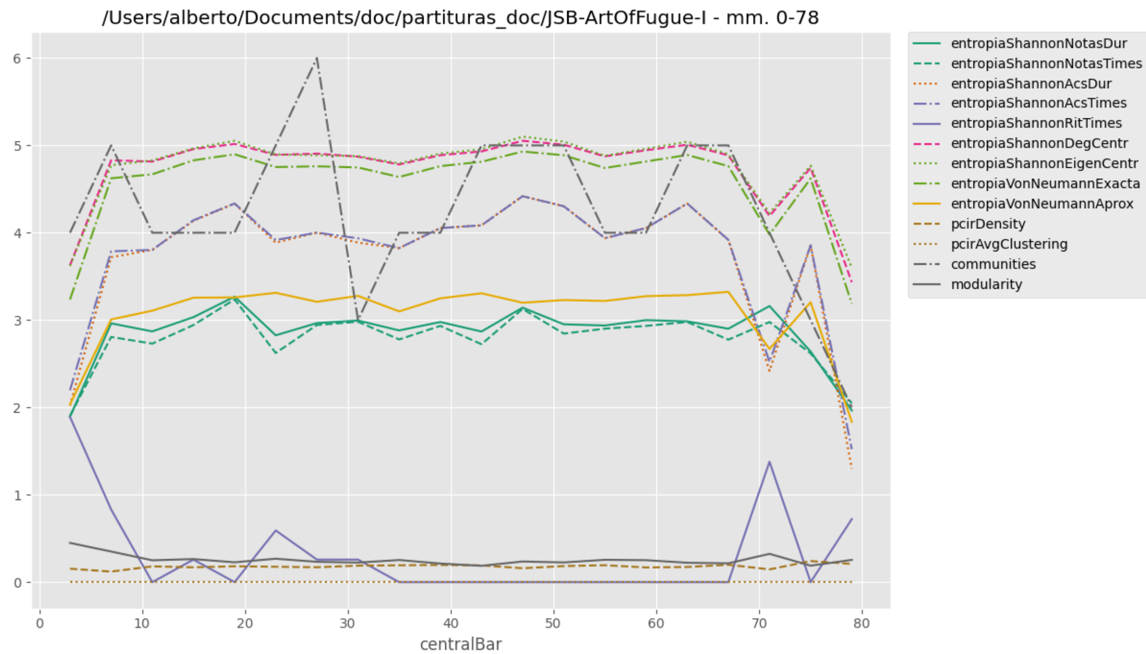

Figure 3.8: ECGs for metrics from the p-c-r graph for mm. 1–8 of J. S. Bach's Contrapunctus I from The Art Of Fugue BWV 1080, taking time windows of length 4 mm., without overlapping.

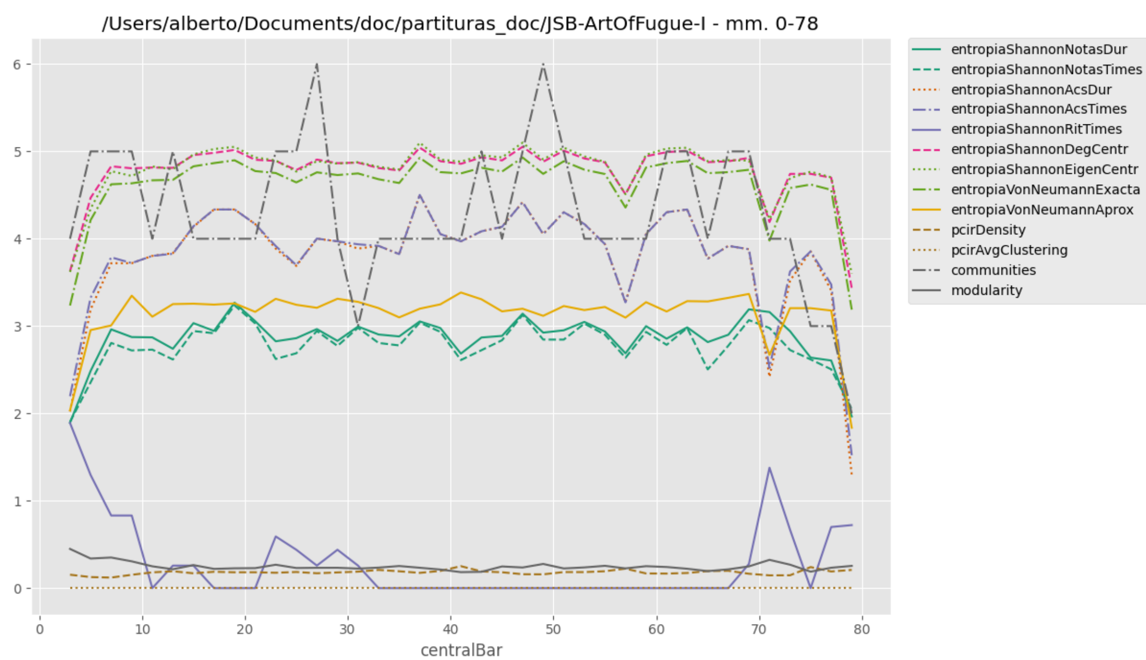

Figure 3.9: ECGs for metrics from the p-c-r graph for mm. 1–8 of J. S. Bach's Contrapunctus I from The Art Of Fugue BWV 1080, taking time windows of length 4 mm., every 2 mm..

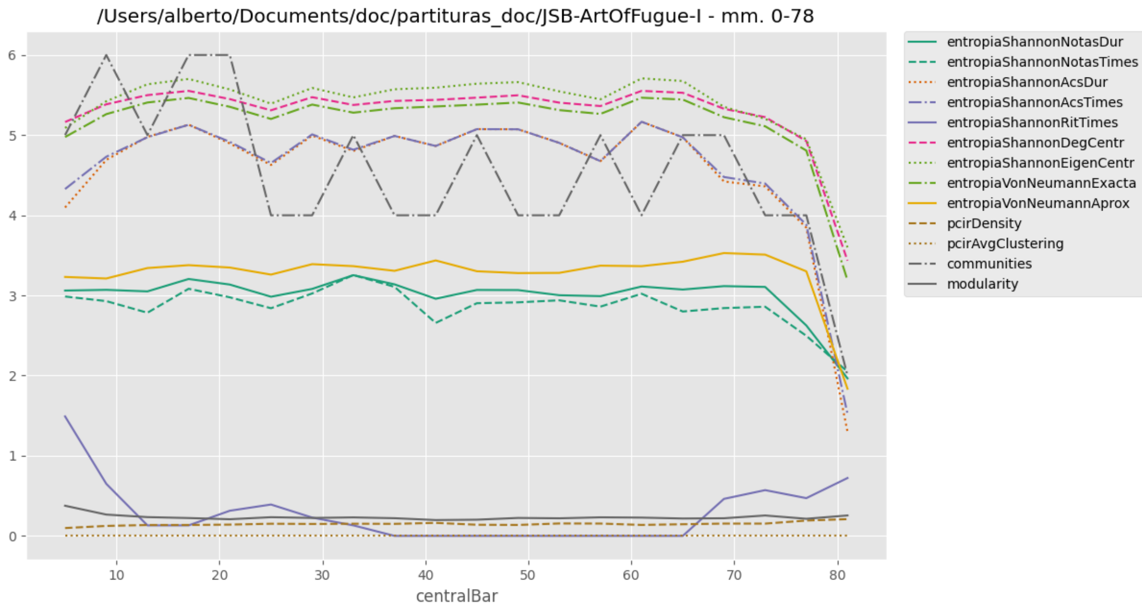

Figure 3.10: ECGs for metrics from the p-c-r graph for mm. 1–8 of J. S. Bach's Contrapunctus I from The Art Of Fugue BWV 1080, taking time windows of length 8 mm., every 4 mm..

- Shannon entropy of chords according to their number of occurrences is almost equal to entropy with respect to their total duration.
- The five entropy metrics mentioned in the first two points above are almost parallel to each other.
- As to the values of Shannon entropy of pitch classes with respect to occurrences and duration, they are near each other and show to have very similar behavior, they differ more than the groups of entropy computations discussed above.
- Entropy of rhythms (taking their frequency distribution) follows a very different contour, sometimes contrary and in occasions independent from other metrics. It shows a long flat region with value 0, corresponding to a long section of the score in which events occur at even intervals of time, resulting in only one rhythmic value (therefore their entropy equals 0 throughout those bars).

- There are some evident common peaks or valleys in the ECGs for several metrics at a certain time or around a certain time. In the next section we will see these often coincide with section changes.

The fact that different entropy measures have a very similar behaviour may lead to keeping only a few of them. Nevertheless, the slight crosses observed between some of them could be meaningful for style classification. This issue will be addressed in future papers.

Besides this general considerations, we are interested in contrasting plots corresponding to different time window settings, and what can that analysis tell us about the music. This involves translating plots. ECGs can be regarded as a general map of how information in the score changes, from either a statistical and graph-theoretical point of view. That is, how much variety or monotonicity of elements is found in each time window. Thus, when looking at this plots for a whole score, rather than looking at values of music elements themselves, it is the change in their distribution and relations.

4 Results.

We begin analyzing a fragment taken from a well-known example of European classical music: the first counterpoint of J.S. Bach's 'The Art of Fuge'. We get its principal attributes and show its associated graphs.

J.S. Bach - The Art of Fugue, Contrapunctus I, mm.1-8.

#eventos: 34

#eventos/totalDurAcs= 32.0 da numero promedio de eventos por cuarto

totalDurNotas= 60.5/totalDur= 32.0 da numero promedio de voces por cuarto

totalTimesNotas= 100/totalDur= 32.0 da densidad promedio por cuarto

Entropï¿œa de Shannon de clases de altura c.r. a la duraciï¿œn: 2.999988562883508

Entropï¿œa de Shannon de clases de altura c.r. a las repeticiones: 2.9564684817727676

Entropï¿œa de Shannon de Acordes c.r. a la duraciï¿œn: 4.045816369049408

Entropï¿œa de Shannon de Acordes c.r. a las repeticiones: 4.3149727675300324

Entropï¿œa de Shannon de Ritmos (c.r. a las repeticiones): 1.5830435300531966

Entropï¿œa de Shannon de la grï¿œfica c.r. a la centralidad de grado: 4.969583379448451

Densidad de la grï¿œfica de alturas-acordes-ritmos: 0.1036036036036036

Aglomeraciï¿œn promedio (average clustering) de acordes: 0.0 # comunidades: 5

Aglomeraciï¿œn promedio (average clustering) de clases de alturas horizontales: 0.5536966149182225
Densidad de la grï¿œfica de clases de alturas horizontales: 0.4727272727272727

Aglomeraciï¿œn promedio (average clustering) de acordes: 0.0256993006993007 Densidad acordes: 0.07142857142857142

Tamaï¿œo de una base de ciclos de acordes: 6

Luna: Bach (cc. 1-16), Bartok (cc. 1-19), Brubeck (cc. 1-15) y Haydn (cc. 1-18).

vs

Bach (art of fugue, Contrapunctus I, cc.1-8, 8-16cc., 1-22),

de 8 en 8 cc., cada 4:

cc.1-8: 5 coms.

- Com. 1: la-do-mi ; tercera mayor: do-mi, mi-sol#, fa-la; negra
- Com. 2: sol#-si-fa ; tercera menor (segunda menor): fa-sol#, re-fa, sol#-si, si-re ;
- Com. 3: sol-sib ; cuarta justa: si-mi, re-sol, sol-do ; corchea
- Com. 4: do#-re ; re-la, fa ; blanca
- Com. 5: fa#; cuarta aum: do-fa#

cc.5-12: 6 coms.

- Com. 1: do-fa-(fa#) ; (2a. m) do-mi-fa, si-do , (4a. A) do-fa# ; do-fa-sol;
- Com. 2: re-sib ; 3a. m: re-fa-la, re-fa, sol-sib, si-re ; SibM, sib-re, re-sol ; corchea

- Com. 3: mi-la ; 4a. J mi-la, si-mi, re-mi-la, re-sol-la , (2a. M)
- Com. 4: (sol#) ; 3a. M mi-sol#, fa-la, LaM, do-mi ; negra
- Com. 5: do#-sol ; LaM7, do# dim ,sol dim
- Com. 6: si; SolM7 sol#-si

cc.13-20: 6 coms.

- Com. 1: sol#-si-fa ; sol# dim FaM, SolM7, MiM, re-fa, sim ; corchea
- Com. 2: sol-sib ; 3a. m: Solm, SolM, Lam7m, Mim, DoM, mi dim;
- Com. 3: do# ; 3a. M, 2a. m: LaM7, MiM (FaM, SibM) ; (negra)
- Com. 4: re-la; 4a. J, re m, re-la, mi-la
- Com. 5: mi; 2a. M la m, MiM7 , si dim, DoM7
- Com. 6: do-fa# ; ReM7, fa# dim

The set of graphs associated with a music fragment discussed in this text yield both a quantifiable as well as a visual tool for music analysis in a general musical context.

Also, the present work points toward a possible definition of *musical meaning* from the point of view of network and time series analysis and visualization: we may consider each type of music element together with its attributes as an element of different networks, which includes, for example, its neighbors, the paths or cycles from/to/through it, as well as their attributes, and how these change along a certain lapse of time or interval of events.

From the plots of graphs associated to a sequence of vertical events we can trace some principal elements of Schenkerian analysis (the reader may refer to Forte and Gilbert [1982]):

- Looking at how an associated graph mutates along the fragment, and filtering by weight we can recover certain elements of the different levels of deepness of musical structure (again, surface/subjacent level).

- Form determined by clear changes in texture: apparently the analysis of the ECGs obtained can help track certain changes in form; smaller/bigger time windows -> surface/subjacent level.

On the other hand, from the analysis of ECG plots for moving time windows along a fragment we obtain results relatable to both Schenkerian analysis and the generative analysis perspective (Lerdahl and Jackendoff [1996]). In a way, the obtained ECGs share certain features with grouping, time-span reduction and prolongational reduction, as defined in the Generative Theory of Music: we are considering a fixed interval of time (which could alternatively be a fixed number of events) which transits along a fragment of a score, measuring at each stage how relevant musical information is organised. This way, certain critical points arise, at which the form could be segmented, or perhaps find tension or release points.

For the specific cases we have examined, analyzing a music score together with its resulting series of ECG plots we can conclude that indeed there are several relevant musical features that are somehow reflected in one or more network metrics. Given a series of plots, there seems to be a strong correlation between the frequency of certain peaks or valleys, and the *magnitude* of the change in the music. By this, we mean there is a more drastic change in one or more music features.

As "control" examples we have chosen some representative works of European classical, though it is not our intention to restrict ourselves to that repertoire. In fact, we are more interested in applying the present data analysis methods to music scores outside the Western corpus of the so-called common practice period. Yet we believe it is pertinent to contrast our proposal with traditional tonal analysis.

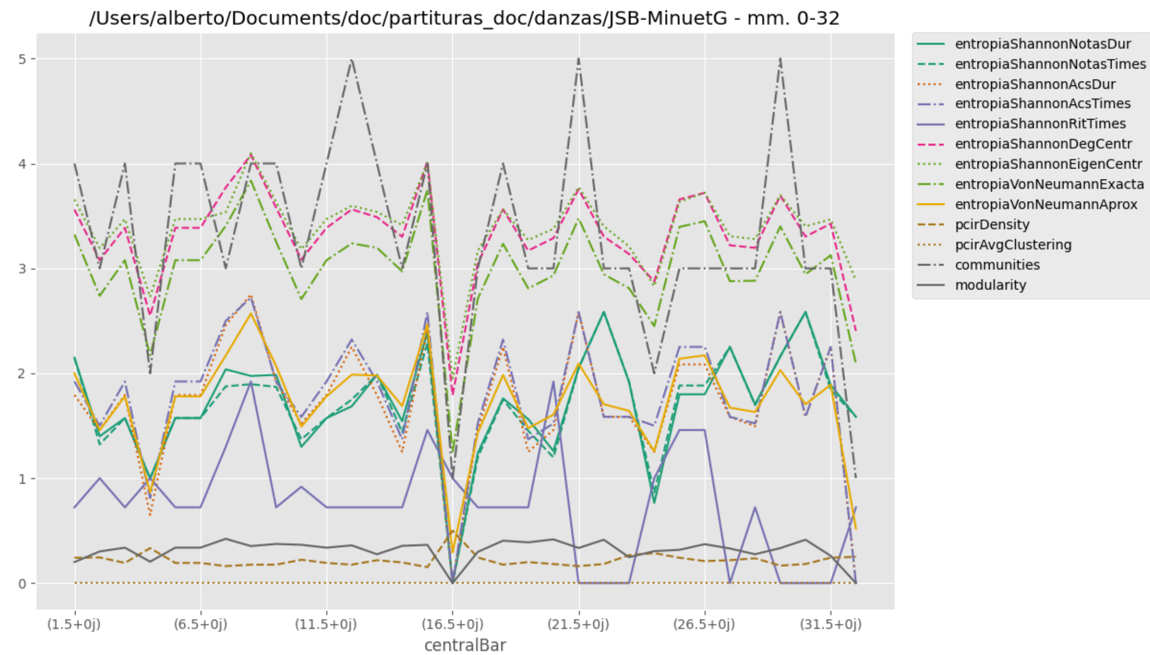

Figure 4.1: ECGs for metrics from the p-c-r graph for J. S. Bach's Minuet in G from the Notebook for Anna Magdalena, taking time windows of length 1 measure, without overlapping.

Let us look closely at a simple example: the well-known Minuet in G Major from the book for Anna Magdalena Bach. This piece shows a stereotypical minuet form in a major key: thirty two bars, with the first section englobing bars 1–16, a phrase in the dominant in bars $17-24$, and the repeat of the second phrase of the first section.

Example 4.1. As we stated before, in the above plots we may observe several points with recurrent presence of valleys or peaks. For example, around measures 8 and 16, coincident with the end of the first and second phrases. Around m. 21 we also find what seem to be relevant peaks in the plots, this time within a segment where we find a modulation to D Major. Finally, we also notice a persistent peak around mm. 25–27, coincident with the last phrase of the piece and a modulation back to G Major (from D Major).

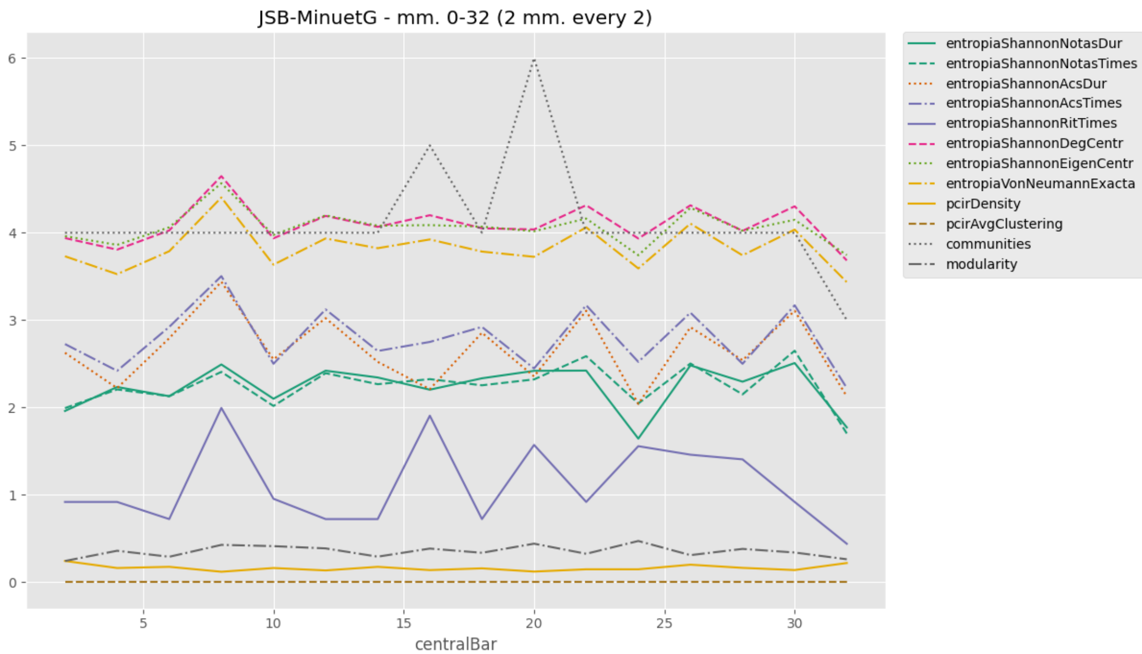

Figure 4.2: ECGs for metrics from the p-c-r graph for J. S. Bach's Minuet in G from the Notebook for Anna Magdalena, taking time windows of length 2 mm., without overlapping.

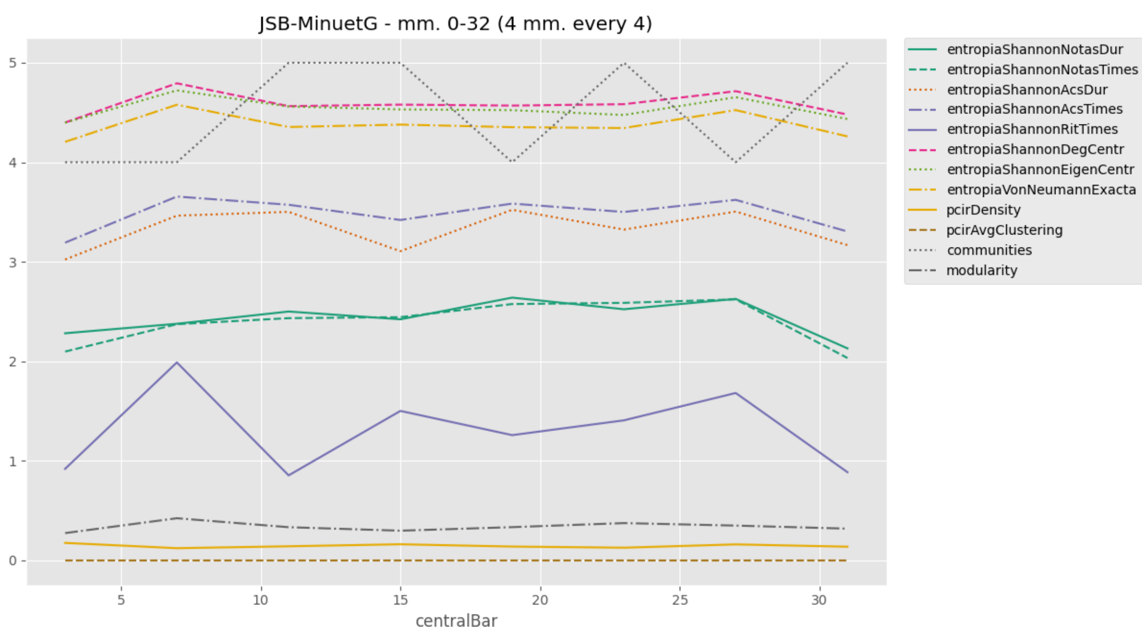

Figure 4.3: ECGs for metrics from the p-c-r graph for J. S. Bach's Minuet in G from the Notebook for Anna Magdalena, taking time windows of length 4 mm., without overlapping.

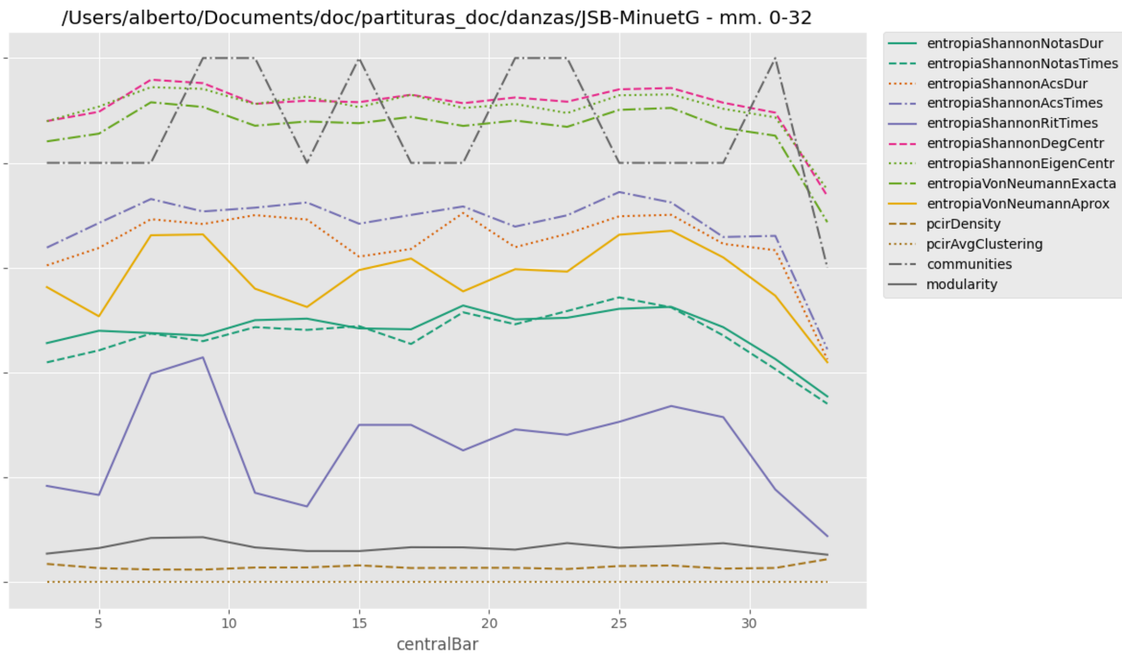

Figure 4.4: ECGs for metrics from the p-c-r graph for J. S. Bach's Minuet in G from the Notebook for Anna Magdalena, taking time windows of length 4 mm., every 2 mm..

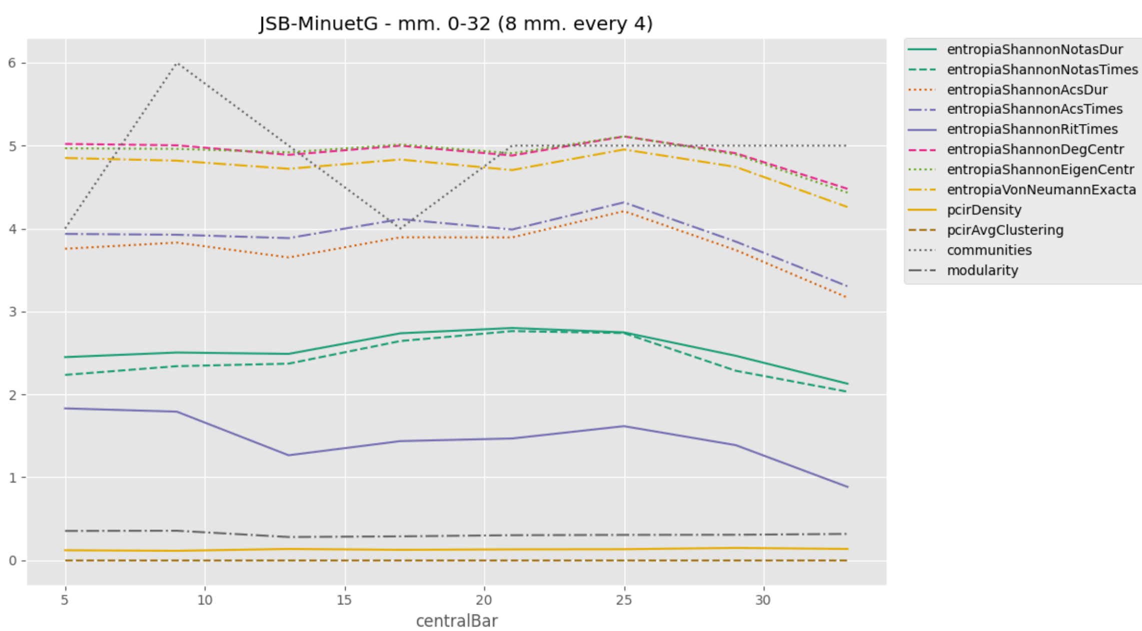

Figure 4.5: ECGs for metrics from the p-c-r graph for J. S. Bach's Minuet in G from the Notebook for Anna Magdalena, taking time windows of length 8 mm., every 4 mm..

5 Conclusions and future work

Besides looking deeper into emerging communities in music-defined graphs and networks, we want to study their motifs and their possible musical interpretation.

We will seek a deeper analysis of the time series obtained from a larger score corpus. Additional to the use of dynamic time warping to compare time series, we are interested in also using their visibility graphs for such task.

It is projected to use hierarchical clustering algorithms for sample classification.

We are currently seeking a mathematical criteria which could lead to an automatic music form segmentation tool, based on entropy and other graph properties. It would be interesting to also apply signal processing methods, as in Buongiorno Nardelli [2023] for the time series defined by the metrics computed for the graphs and networks hereby discussed.

Inquire in the behavior and utility of networks containing other music parameters and relations: rhythm, dynamics, texture, etc.

Inquire about the convenience or meaningfulness of keeping just some of the metrics here considered (droping those which are too close to each other.... or... are they subtle but meaningful for style distinction?).

References

Jason I Brown and Ian George. Modulation graphs in popular music. *arXiv preprint arXiv:2306.13691*, 2023.

Marco Buongiorno Nardelli. Tonal harmony and the topology of dynamical score networks. *Journal of Mathematics and Music*, 17(2):198–212, 2023.

Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6), dec 2004. doi: 10.1103/physreve.70.066111. URL `https://doi.org/10.1103%2Fphysreve.70.066111`.

Michael Scott Cuthbert and Christopher Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. 2010.

Matthias Dehmer and Abbe Mowshowitz. A history of graph entropy measures. *Information Sciences*, 181(1):57–78, 2011.

A. Forte and S.E. Gilbert. *Introduction to Schenkerian Analysis*. Norton, 1982.

Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

Edward Casey Kenley and Young-Rae Cho. Detecting protein complexes and functional modules from protein interaction networks: A graph entropy approach. *Proteomics*, 11 (19):3835–3844, 2011.

Fred Lerdahl and Ray S Jackendoff. *A Generative Theory of Tonal Music, reissue, with a new preface*. MIT press, 1996.

Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

Mark Newman. *Networks*. Oxford University Press, 2018.

Johan Sejr Brinch Nielsen. Statistical analysis of music corpora, 2009.

Iris Yuping Ren. Complexity of musical patterns. *University of Warwick*, 2014.

Wai Man Szeto and Man Hon Wong. A graph-theoretical approach for pattern matching in post-tonal music analysis. *Journal of New Music Research*, 35(4):307–321, 2006.

Adrian Walton. A graph theoretic approach to tonal modulation. *Journal of Mathematics and Music*, 4(1):45–56, 2010.