Erasmus Mundus Masters in Complex Systems Science Second Year Summary Report and Two Papers

Iris Yuping Ren[1,2]

[1] Complexity Science Centre, University of Warwick
[2] Inria Saclay Ile-de-France
yuping.ren.iris@gmail.com

Abstract. This report includes a summary of the work I have done during my M2 project, including but not limited to two papers in the appendices. In the first paper, we extend techniques from topological data analysis (TDA) to networks. These techniques include persistence homology, Vietoris-Rips filtration, Betti-number barcode, persistent diagram, bottleneck distance, and our original method *edge weight filtration* and *persistence time distribution*. We apply them to small world and scale free networks, and the networks derived from pieces of music which I presented in my previous work, in which we represent a piece of music as a network, with nodes corresponding to pitch and duration pairs, and edges connecting occurrences adjacent in time. Using network statistics and TDA methods, we can produce new music and recover facts about music which humans also appreciate. We also propose future applications of this methods for music creation, analysis and educational purposes. In the second paper, we created an agent-based model of music group playing under four different interaction mechanisms. Based on real music data, added randomness and simplifying assumptions, we examine how agents synchronize and deviate from the original score. We find that while music can make synchronization complex, it also helps reducing the total deviation. By studying the simulation process, several conclusions on the relationship between different growing speeds of total deviations and different interaction schemes are drawn. With interpretation from a musical point of view, we find that, in a music ensemble, listening to neighbors helps the players end up in sync. However, if people do not listen carefully enough, the deviation becomes larger than when people do not listen at all. On the issue of whom one should listen to, the results show no significant differences between listening to the immediate neighbors and to the whole group. Finally, we also observe that large deviations can be reduced by making the musicians move while playing.

Summary

During my M2 internship, I have been working on the "topological data analysis on Music Networks" project, extended from my M1 project. In comparison to the M1 project, I put my focus on the topological data analysis part which I didn't have the time go deep last year. I have also detailed and improved the results from my M1 project, thanks to the opportunities to many visits to different institutes, making presentations and having feedback during my internship. Besides this M2 poject, one of my mini-projects done in the first semester at Ecole Polytechnique has been published by the conference CMMR2015 and won a best master student paper award (in appendices). I wish to publish this M2 project paper (in appendices) later on as well. Also since my main subject of my M2 internship is still the "TDA on Music Networks", I will be mainly talking about this subject in the following sections in the summary. I will give an introduction to the subject first, and then give results overview and make a contribution discussion. [3]

Introduction

Why this is important? The relationship between music and shapes have generated considerable interest since the Pythagoreans. Generally the marriage of mathematics and arts has produced many more interesting results [1–3]. They are not only aesthetically pleasing, but also useful for artists and mathematicians, both theoretically and practically: people have succeeded at automating art and music production; resolving cases of uncertain authorship; and improving arts, music and mathematics education; and there are too many more examples to be listed exhaustively. The applications of more tools from mathematics are needed and destined to thrive this interdisciplinary study of arts and science. One day, we may be able to make machines automatically write and improvise music and tell the musical finger-print of a music piece or composer.

We provide some answers to the questions by investigating the prominent connections between notes in music by finding the shape of music pieces using complex network and topological data analysis (TDA). Algebraic topology is a branch of mathematics that uses tools from abstract algebra to study topological spaces. Statisticians started to combine some theories of it with data analysis and gave birth to the field called topological data analysis.

The traditional approach of studying networks is to draws on theories and methods including graph theory from mathematics, statistical mechanics from physics, data mining and information visualisation from computer science, inferential modeling from statistics, and social structure from sociology. And they mostly focus on locally defined quantities of nodes and edges, such as node degrees, edge weights and correlations between neighboring nodes, and so on.

[3] For the sake of succinctness, only some figures are shown here. Some more visualisations can be accessed at `https://sites.google.com/site/irisyupingren/documents`.

However, these methods become cumbersome when dealing with many-body properties and do not capture the precise mesoscopic structure of complex networks [5]. As we can see from the following sections, one solution to this problem is using the TDA methods, because it takes the dynamic and evolution of a network into considerations. Moreover, this new method creates the bridge between pure and applied math, which allows many more import of toolset from pure mathematics.

A melody is a time sequence of notes. Therefore, the connections between notes are interesting and essential to investigate, and we can pose questions more specifically: what are the notes with most connections? Can we recover music structures out of these connections? How are the connections in different music pieces differ from each other? Furthermore, can we summarise music in a more visual way using the connections between notes?

What tools should be used To answer the questions, the first step is to construct the music networks.Music emerges from the non-trivial connections and interactions of the sounds. In almost all musical works, especially classical western music, each musical instrument or singer voice should emit a limited number of such sounds as written in sheet music. In other words, we work with musical pieces which consist of the repetition of a discrete set of sounds, which differ in pitch and duration. Although this is in its simplest form, music is discretisable in terms of pitch and duration, which in turns are the backbone of music, and therefore allows us to use a network based approach to analyse. This discretised music consideration also motivated many relevant works from other topological and geometrical perspectives [6–10], generating many useful results for both musicians and mathematicians.

The second step, after obtaining the networks, is to analyse the network degree distribution and apply TDA tools. Network statistics reveals general information of the structures of our networks. TDA provides topological information on all scales, which is also the philosophy of persistence, the core of TDA. The details of the methods and the results of analysis will be given in section 2 and 3 respectively. Detailed introduction can be found in [11, 12].

Data and miscellaneous The musical pieces we used for the analysis are Beethoven's String Quartets No.01 - No.09, Mozart's String Quartets No.01 - No. 04, Haydn's String Quartets No.1 - No.2. The data format we start with is musicxml. Similar to .xml files, musicxml consists of different layers of text cells: from parts, measures to notes and attributes, which form a tree-like structure of musical information. One thing to notice is that musicxml is not a sequential encoding format. For example, backup and forward commands are used to encode single parts with multiple staves or multiple voice. This necessitates care in creating a parsing algorithm to convert musicxml to numerical data structure. The way we did it is first to create a list of elements including pitches, time in the music, duration, and dynamics, for each note in the musicxml sequentially.

Second, sort them according to their time element. Only after obtaining a time series of notes, can we construct our sequential network thereafter.

As for the literature in this area of study, there has been some research on this type of musical networks, [13] for example. In our paper, we contest to their results on the power-law degree distribution. Although there is much other research on music and networks, it mostly concerns social networks but not music itself [14–16]. Some other network constructions take a few notes or a chord as a node, for example in [17, 18]. However, our approach is different in that we incorporate duration information and all the melody, rather than just the harmonic progression. Moreover, none of the above works implemented the TDA techniques. Thus, using the TDA and network, two novel methods thriving in mathematical and complexity researches, not only can we exploit a vast body of existing theoretical results, but also can introduce a new contribution, since the classic network models of friendship, social relations, power grids and emails and phone communication are of a very different nature compared to the music networks we build.

Why it has not been done yet The network method itself, which is part of the complexity system science, is a relatively new field with a lot of new tools and applications. The area of study of TDA has only been examined by statisticians and pure mathematician for around ten years. There are some works on this network and TDA topic. Nevertheless, there has not been anyone who applied them into the research of arts and science.

Why I am in the position to do it After finishing my study at Warwick and my first semester here at Ecole Polytechnique, I had already gained insights and hands-on experience in this field. Furthermore, in this area of study, I have established contacts with Institute for IRCAM (Music Acoustic Research Coordination, Paris), Centre for Digital Music (QMUL), SONY CSL (Japan), INRIA (Ecole Polytechnique Paris), and Institut fur Mathematik (Technische Universitat Berlin), and many PhD students from different institutes. The discussions we had really helped me broaden my view in this area of study, and helped me realized that there is still a lot of research potential in this field.

Personally speaking, the benefit of doing a research with music is that it combines my passion for maths and my joy of music giving me constant motivation to learn and try new methods. I have been playing the violin for more than ten years. Also having some experience in vocal training and piano playing, I constantly give performances with student orchestras and choirs. From a more academic perspective, I studied mathematics and physics in addition to statistics during my undergraduate study and I have seen how powerful mathematics can be in physics and I am fascinated by it. Furthermore, the training from the Erasmus Mundus Masters in Complex Systems Science prepared me better to tackle the problems in this field. In sum, I am very passionate about statistical musicology and have given it a lot of thought, and my training in science also puts me in a position to investigate this field.

Results Overview

My results can be divided into two categories: one on small world and scale free network, and the other on the music networks. Also, I employed two different tools (filtrations, introduced in section 3.2) from TDA to look at the problem with different point of view.

Using the first filtration, I made an analysis of the speed of topological change in a network when gradually adding strongly connected edges to weakly connected edges. In the typical network section 6.1, I explored the parameter spaces of scale free and small world networks and found different properties of their topological changes and the speed of the topological changes when adding edges. In the music network section, we apply the same method to music networks and found some potential application in music education.

Using the other filtration, we also explored a range of parameters in small world and scale free networks, and found we could reconstruct their different topologies: the hub topology and the ring topology. This gives us confidence in the application of this filtration to networks. From music networks, we gained the point cloud shapes of each music piece, and also tried clustering methods to classify according to different instruments and composers.

As an improvement on my first project, it is possible now to create music out of the music networks, further consider the implications of a linear in- and out-degree relation, and a set of statistical methods has been used to fit the network degree distribution.

Detailed construction and results can be found in section 3.2 and section 6.1.

Contribution Discussion

On the music side, I have given contribution in the creation, the analysis and the education aspects. Samples of music creation using networks are given. Analysis are performed on three different levels: the note level, (Section 6.2 and Section 6.2); the tonal level (Section 6.2); and instrument and composer level (Section 6.2 and Section 6.2). Music education considerations have been given from the perspective of the speed of topological changes. On the network side,I have questioned the results of another paper [13], which contents power-law degree distributions in music networks.

What follows is a paper detailing the above mentioned results and contributions.

Topological Data Analysis on Music Networks

Iris Yuping Ren[1,2], Frederic Chazal[2], and Charo I. Del Genio[1]

[1] Complexity Science Centre, University of Warwick
[2] Inria Saclay Ile-de-France
yuping.ren.iris@gmail.com

Abstract. We extend techniques from topological data analysis (TDA) to networks. These techniques include persistence homology, Vietoris-Rips filtration, Betti-number barcode, persistent diagram, bottleneck distance, and our original method *edge weight filtration* and *persistence time distribution*. We apply them to small world and scale free networks, and the networks derived from pieces of music which I presented in my previous work, in which we represent a piece of music as a network, with nodes corresponding to pitch and duration pairs, and edges connecting occurrences adjacent in time. Using network statistics and TDA methods, we can produce new music and recover facts about music which humans also appreciate. We also propose future applications of this methods for music creation, analysis and educational purposes.

Keywords: topological data analysis, music, complex network, small world network, scale free network, algorithmic composition

1 Introduction

The relationships between music and shapes have generated considerable interest since the Pythagoreans. Here, we study the connections between notes in music by constructing shapes of music pieces using complex networks and examining those shapes with topological data analysis (TDA).

A melody is a time sequence of notes. In elementary improvisation, the player think of one note after another. Therefore, the connections between notes are interesting to investigate: what are the notes with most connections? Can we recover music structures out of these connections? How do the connections in different music pieces differ from each other?

To answer the questions, the first step is to construct the music networks. As one of the methods in complex systems science widely used in studying social, biological, technological and economic systems, complex networks are abstract entities which can efficiently describes the relationships between individuals. On the other hand, music emerges from the non-trivial connections and interactions of the sounds. Thus, the interdisciplinary approach of complex systems science, using methods from mathematics, physics, statistics, computer science is a natural choice to produce this type of quantitative analysis of musical pieces.

Moreover, in most musical works, especially classical western music, each musical instrument or singer voice is usually asked to emit a limited number of such sounds as written in sheet music. In other words, we work with musical pieces which consist of the repetition of a discrete set of sounds, which differ in pitch and duration. Although this is in its simplest form, music is discretisable in terms of pitch and duration, which features backbone of music, and therefore allows us to use a network based approach to analyse. This discretised music consideration also motivated many relevant works from other topological and geometrical perspectives [6–10], generating many useful results for both musicians and mathematicians.

The second step, after obtaining the networks, is to examine network degree distribution and apply TDA tools for analysis. Network statistics reveals general information of the structures of our networks. TDA provides topological information on all scales, which is also the philosophy of persistence, the core of TDA. The details of the methods and the results of analysis will be given in section 2 and 3 respectively. A detailed introduction to the field can be found in [11, 12].

The musical pieces we used for the analysis are Beethoven's String Quartets No.01 - No.09, Mozart's String Quartets No.01 - No. 04, Haydn's String Quartets No.1 - No.2. The data format we start with is musicxml. Similar to .xml files, musicxml consists of different layers of text cells: from parts, measures to notes and attributes, which form a tree-like structure of musical information. One thing to notice is that musicxml is not a sequential encoding format. For example, backup and forward commands are used to encode single parts with multiple staves or multiple voice. This necessitates care in creating a parsing algorithm to convert musicxml to numerical data structure. The way we did it was to first create a list of elements including pitches, time in the music, and duration, for each note in the musicxml sequentially. Second, sort them according to their time element. Only after obtaining a time series of notes, can we construct our sequential network thereafter.

There has been some research on this type of musical networks, [13] for example has the same construction of music networks. In our paper, we contest to their results which based on the assumption that the music networks degree distribution can be modeled by power-law degree distributions. There are other network constructions take a few notes or a chord as a node, for example in [17, 18]. However, our approach is different in that we incorporate duration information and all the melody, rather than just the harmonic progression, which gives considerations to all music elements in a music piece. Moreover, none of the above works implemented the TDA techniques. Thus, using the TDA and network, two novel methods thriving in mathematical and complexity researches, we can exploit a vast body of existing theoretical results and give new insights to music researches.

For sake of succinctness, only some figures are shown here. The rest of the visualization products can be accessed at https://sites.google.com/site/ irisyupingren/documents

2 Complex Networks

In this section, to establish a base line and compare with music networks later on, we give an overview of the types of networks upon which we used TDA techniques upon.

2.1 Small World Networks and Simulation

The small world networks, also known as the Newman-Watts-Strogatz graph, are characterised by its small average path length, in spite of the presence of a large number of nodes. In addition to social networks, small-world networks are common in biology, physics, computer science, and many other fields [19]. For example, road networks, brain networks and gene networks all have some small-world properties.

To generate/simulate this type of network, we use the package provided in [20]. Two key parameters are k, the number of connected nearest neighbors, and p, the rewiring probability. The process of generation is two step: first, put nodes in a ring topology and then connect their k nearest neighbors; second, switch edges according to the rewiring probability p. As for the weights of edges, we use a range of different distributions to assign the weights: uniform, normal, powerlaw, etc. We use this type of network as a base line before applying our TDA techniques to the music networks. Relevant results are shown in Section 5.

2.2 Scale-free Networks and Simulation

Scale-free networks, as suggested by the name, have the property of self-similarity so that they look similar at different scales, and therefore do not have a fixed scale. This property is realised by having a power-law degree distribution $P(k) \sim k^{-\gamma}$. As a consequence of this distribution, unlike the topology of small world networks, there can be nodes with very large degrees, which are also called hubs.

We also use the package provided in [20] to generate/simulate this type of network. The mechanism is to progressively add nodes to the existing network, and then introduce edges with preferential attachment, which is, roughly, the nodes with a larger degree get more links. This method has three main parameters:

- α: Probability for adding a new node connected to an existing node chosen randomly according to the in-degree distribution.
- β: Probability for adding an edge between two existing nodes. One existing node is chosen randomly according the in-degree distribution and the other chosen randomly according to the out-degree distribution.
- γ: Probability for adding a new node connected to an existing node chosen randomly according to the out-degree distribution.

Because we have to choose one of the three options at each step, $\alpha+\beta+\gamma = 1$. Like the small-world networks, we use this type of network as a base line before applying our TDA techniques to the music networks. Relevant results are shown in Section 5.

2.3 Construction of Music Network

To build our networks, we consider the adjacency matrices of the music pieces. To map notes to nodes, we use the lexicographical order of pitch and duration. For a larger piece, such as 4 movements of a string quartet, this normally yields from a few hundreds nodes up to more than a thousand. For smaller pieces, with a duration of 2–3 minutes, the nodes are usually less than a hundred. To standardise our approach, we label the nodes with the commonly accepted midi number for pitches(C4=60, D4=62, etc.), and assign them a duration in seconds, converted from the logic value in musicxml using tempo information. Rests are denoted by zeros, while grace notes and other musical decorations are ignored. The octaves are not identified, because we are not looking for particular harmonic topological structures, and from the point of the melody, the same note in different octaves are two very different notes.

As for the entry in the adjacency matrices, that is, the edges in the network, we will add one whenever one node is followed by the other in the music. For example, if we have a scale C-D-E-F-G-A-B, each note lasting for 1 second, then there will be 7 nodes, the first one is (C, 1s), (D, 1s) the second, (E, 1s) the third, ect.. Also, there will be 6 edges, which are (C, 1s)-(D, 1s), (D, 1s)-(E, 1s), etc. When there is a chord in the musical sequence, we connect all the nodes in the chord to the next one, and all the previous nodes to all the nodes in the chord.

In the end, we obtain a network with the number of nodes equal to the number of all pitches multiplied by the number of all durations in the piece. We take the giant connected components of this network for further studies, and leave out the unused pitch and duration combination. As the music progress, it defines a path in the network.

3 TDA

As we introduced in the first section, we can explore new network properties using TDA. To formally introduce the subject, we provide the mathematical definitions and intuitions for each new concept.

3.1 Simplex and Simplicial Complexes

Simplices are mostly known as points, lines, triangles, etc., up to arbitrary dimensions. Mathematically, suppose we have a set of points $p_0, p_1, ...p_n$ satisfying the condition that: $p_1 - p_0$, $p_2 - p_0$, ..., $p_n - p_0$ are linearly independent. The n-simplex set is $C = \{\theta_0 p_0 + ... + \theta_k p_n \mid \theta \geq 0, 0 \leq i \leq n, \sum_{i=0}^{n} \theta_i = 1\}$.

A simplicial complex is a set of simplices satisfying two conditions:

– Any convex hull of the non empty subsets of the n+1 points in a n-simplex (called the face of a simplice) of the simplices from this set is also in this set
– The intersection of any two simplices from this set is a face of both simplices

For a more thorough introduction on the subject, we refer the book in [21].

3.2 Filtration

Simply put, a filtration is a nested sequence of increasing subsets. Mathematically, a filtration is an indexed set F_i of subobjects, with the index i running over an index set I that is a totally ordered set, the subobjects should satisfy that: if $i \leq j, i, j \in I$, then $F_i \subseteq F_j$. Intuitively, one can imagine a filtration acts on simplicial complexes and gives a collection of new complexes, with the index depending on a distance function. Therefore, the index i is going to control the level of the growing, which is also known as filtration time. We use two different filtration methods on the networks for different purposes:

Vietoris-Rips Filtration Vietoris-Rips filtration has been used widely in topological data analysis. It is a sequence of simplicial complexes built on a metric space to add topological structure to a disconnected set of points.

Formally, we first introduce the Vietoris-Rips complexes: Let X be a subset of a metric space and let d be its metric. Pick an $\epsilon > 0$. Construct a simplicial complex as follows:

– Add a 0-simplex for each point in X
– For $x_1, x_2 \in X$, add a 1-simplex between x_1, x_2 if $d(x1, x2) \leq \epsilon$
– Similarly, for $x_1, x_2, x_3 \in X$, add a 2-simplex with vertices x_1, x_2, x_3 if $d(x_1, x_2), d(x_1, x_3), d(x_2, x_3) \leq \epsilon$
– Generally, for $x_1, x_2, ..., x_n \in X$, add an n-simplex with vertices $x_1, x_2, ..., x_n$ if $d(x_i, x_j) \leq \epsilon$ for $0 \leq i, j \leq n$

After these steps, we obtain the Vietoris-Rips complex, and the Vietoris-Rips filtration is the collection of Vietoris-Rips complexes at all scales.

Construct Vietoris-Rips Filtration with Network: a Metric To apply the Vietoris-Rips filtration to networks, we need to define a metric. We define the distance between nodes in the network to form a distance matrix D_{ij}:

$$D_{ij} = \sum_{w \in g_{ij}} w^{-1}$$

where w represents the weights on edges, and g_{ij} is the shortest path (geodesic) in network between the nodes i and j. The metric is taken this way because we want to construct a space where the more closely connected nodes in networks are closer in the space. Therefore, we took the inverse of the weight since we look at networks where larger weights indicate more closely connected relationships between nodes. In our music networks, this is exactly the case. For instance, if in a music, there are ten transitions between a C4 quiver(also known as a quarter-note) and a G4 quiver, and four between a C4 quiver and a F$^\sharp$4 quiver, the edge weights will be 10 and 4 respectively between the nodes; so the distances will be 0.1 between the C4 quiver and the G4 quiver, 0.25 between the C4 quiver and the F$^\sharp$4 quiver, and it is natural to do this musically, too. However, when the edge weights are equivalent to the distance between nodes already, for example in many road networks, it should not be taken reciprocal when constructing the filtration.

Edge Weight Based Filtration Besides the widely used Vietoris-Rips filtration, we propose a novel filtration for the network based on its edge weights. In a network setting, since we want to investigate first the strong links and then the weak links, it is natural to use the edge weights to determine whether an edge will be added early or late, as shown in Fig. 1: the edges are added at different time with $t(edge) = max(weights) - weight(edge)$. In this way, the network at different time steps can be seen as the superlevel subsets from the network in which we will see the stronger connections appear first. Due to this thresholding of edge-adding, our networks are evolving with the filtration time. With inspection in dimension zero and dimension one, it allows us to see more mesoscopic structures and many-body properties of complex networks' components and loops formation. There has been similar investigation in social and biological networks such as [5].

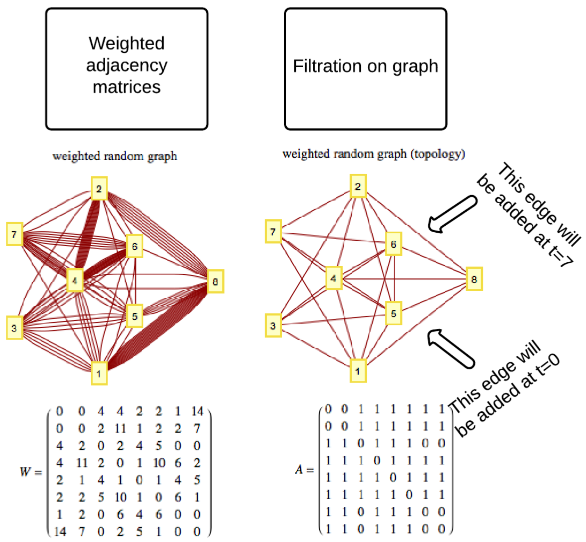


Fig. 1. From weighted adjacency matrix to the construction of persistence homology

3.3 Homology and $Betti_k$ Number

Homology is a certain procedure to associate algebraic objects with topological objects. It can be described by $Betti_k$ numbers. Roughly speaking, $Betti_k$ numbers are the number of k dimensional holes. In this project, we only look at $Betti_0$ and $Betti_1$, corresponding to the component and loop structures in the networks. Mathematically speaking, it is the rank of homology group , which is defined as:

$$H_k := ker\partial_k / im\partial_{k+1}$$

$$\text{where } \partial_k : C_k \longrightarrow C_{k-1} \text{ is the boundary operator}$$

in which C_k is group consists of the set of all k-chains and the operation of addition, and k-chain is a sequence of simplices $\sigma, \sigma_1, \sigma_2, \cdots, \rho$ s.t. any 2 consecutive ones share a k-face. [21] provides more systematic materials on this subject.

3.4 Persistent Homology

Persistent homology is at the core of topological data analysis. The philosophy of persistence is to avoid selecting a fixed value of the threshold, instead to obtain a useful summary for all the different values of at once.

Mathematically, given a space of simplicial complex, a distance function corresponds to a filtration of the simplicial complex $\emptyset = F_0 \subseteq F_1 \subseteq ... \subseteq F_n = F$, when $0 \leq i \leq j \leq n$, the inclusion $F_i \hookrightarrow F_j$ induces a homomorphism $f_k^{i,j}$: $H_k(F_i) \rightarrow H_k(F_j)$ on the simplicial homology groups for each dimension k. The k persistent homology groups are the images of these homomorphisms. Then the interesting values are the $Betti_k^{i,j}$ numbers we discussed before, which are the ranks of the persistent homology groups.

3.5 Barcode

Plotting the barcode is a standard way to encode the persistence homology in topological data analysis, which represents the topological information of a growing set strata. We plot the $Betti_k^{i,j}$ numbers in the following way to generate the barcode: with the x axis being the filtration time, the y-axis is the arbitrarily ordered homology generators H_k, we plot $Betti_k^{i,j}$ line segments from i to j.

A fast way to read a piece of information from the barcode is to look at the cross-section of a certain time: the number of bars is equal to the number of our interested components in the certain dimension, that is, the Betti number. For example, when $k = 0$, the barcode cross-section number tells how many components there are; $k = 1$, the barcode cross-section number tells how many loops there are, etc. A Betti interval with long filtration interval can be interesting because it is exhibiting some long-existing topological structure.

3.6 Persistence Time Distribution

After obtaining the barcode for edge weight filtration, if we want to investigate the speed of topological features' change, we can plot a distribution of the persistence time. The x axis has values of filtration times x_i. The heights indicates the first-order difference of the Betti numbers with specific x_i values. For dimension zero, the heights are $\mid Betti_k^{x_0,x_{i+1}} - Betti_k^{x_0,x_i} \mid$. For dimension one, the heights are the heights are $\mid Betti_k^{x_i,x_\infty} - Betti_k^{x_{i+1},x_\infty} \mid$.

Intuitively, they are the distributions of the filtration times of dying topological dimensional zero features, and distributions of the filtration times of birth in dimension one features. In a network setup, it visualises how fast the components are connected and how fast the loops are forming. This is interesting because it shows the topological changes of networks with gradually connecting edges according to weights. For example, given a weighted social network, weights being the friendship strength, if one wants to investigate how the components of community changes under different friendship strength, the persistence time distribution can give a summary. One such result is shown in Fig. 3 and Fig. 5. We can easily see the speed of topological change at each filtration time using this distribution.

3.7 Persistence Diagram

Persistence diagrams are multi-sets representing of barcodes, or the $Betti_k^{i,j}$ number intervals we introduced before. It is also commonly used to summarise the barcode. Simply put, it takes the birth and death times of a barcode, and plots them in pairs on $\mathbb{R}^2$, with each point associated with multiplicity equal to the number of such features.

Formally, having the $Betti_k^{i,j}$ numbers, the diagram $D(f) \subset \mathbb{R}^2$ is the set of points (i, j), counted with multiplicity $Betti_k^{i,j}$, with all points on the diagonal, counted with infinite multiplicity.

In this construct, a point closer to the diagonal would be more likely to be noise, because its birth and death time would be very close to each other. More salient features are the ones more off-diagonal. It is a summary of the persistence construction, and we use it for clustering across different networks.

One of advantage of the persistence diagrams is their stability [22]. Stable here means that small changes in the input imply only small changes in the diagram. More will be introduced in the next subsection using bottleneck distance.

3.8 Bottleneck Distance

Bottleneck distance is used to measure how different two persistence diagrams are. Mathematically,

$$d_B(Dgm_p(f), Dgm_p(g)) = \inf_{\eta} \sup_{x \in Dgm_p(f)} \|x - \eta(x)\|_\infty$$

where η is the set of bijections between the multi-sets $Dgm_p(f), Dgm_p(g)$, and $\|p - q\|_\infty = max(\mid x_p - x_q \mid, \mid y_p - y_q \mid)$. Intuitively, it matches the off-diagonal points in two persistence diagrams in a way that makes the match the most plausible, and then take the distance of the largest value in x- or y-coordinate. The near-diagonal points will be canceled by the points on the diagonal, and this is why the definition of persistence diagrams gives infinite multiplicity to diagonal points.

As mentioned in the last subsection, it has been proven that the persistence diagrams are stable using the bottleneck distance, and therefore appropriate to be used in detecting topological features in data sets with noises. Formally, with mild assumptions, we have: $d_B(Dgm_p(f), Dgm_p(g)) \leq \| f - g \|_\infty$. More results can be found in [22].

4 Clustering

We use the following clustering concept to get a better understanding of our topological data analysis results: to compare between networks using bottleneck distances introduced in section 3.8, also to clustering within networks using generated distance matrices introduced in section 3.2, and its multi-dimensional scaling, and finally to evaluate the clustering results. This section is an overview of the clustering tools we use.

4.1 Hierarchical Clustering

Hierarchical clustering is a standard clustering method to be used on distance matrices for n objects. The main goal is to build a hierarchy of clusters, and there are many algorithm variations to achieve this goal. The procedures we use are as follows:

- First, assign each object to its own cluster
- Second, join the two most similar clusters. Similarity here is described by the distance between clusters: $D(X,Y) = \min_{x \in X, y \in Y} d(x,y)$
- Iteratively continuing the second step until there is just a single cluster

This method is also known as the single linkage method.

The results of the clustering will be visualised as dendrograms in section 6, which give the sequence of cluster fusion and the distance at which each fusion took place [23].

4.2 K-Medoids

K-means and k-medoids are widely used in real world data clustering. Unlike hierarchical clustering, they provide a k-partition of the data set. Both these partition methods try to minimize the distance between points in the same cluster and a point selected as the cluster center. K-means is one of the very basic in machine learning, and its successor, K-medoids, is known as a more stable version of K-means. The main difference between the two is that K-medoids chooses datapoints as centers and works with distance matrices. We only introduce and use the Partitioning Around Medoids (PAM) algorithm here to realise K-medoids later on:

- Initialisation: randomly select k of the n data points as the centers
- Associate each data point to the closest medoid using the distance matrices
- For each center m, we iterate through each non-center data point p to calculate the total cost of swapping m and p
- Select the lowest cost configuration
- Repeat step 2 to 4 until there is no change in the partition

4.3 Clustering Similarity

We use a value called the Jaccard clustering similarity to evaluate our clustering results:

$$J = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}$$

where M_{11} represents the number of observation pairs where both observations are in the same cluster in both clusterings; M_{10} represents the number of observations where the observations are in the same cluster in the first clustering but not the second; similarly, M_{01} is the other way around; finally, M_{00} represents the number of observation pairs where neither pair are in the same cluster in either clustering.

This quantity gives a measurement as to how similar the clustering is from the view of the comemberships, hence very suitable to unsupervised learning such as k-medoids and hierarchical clustering.

5 Considerations for Realisation

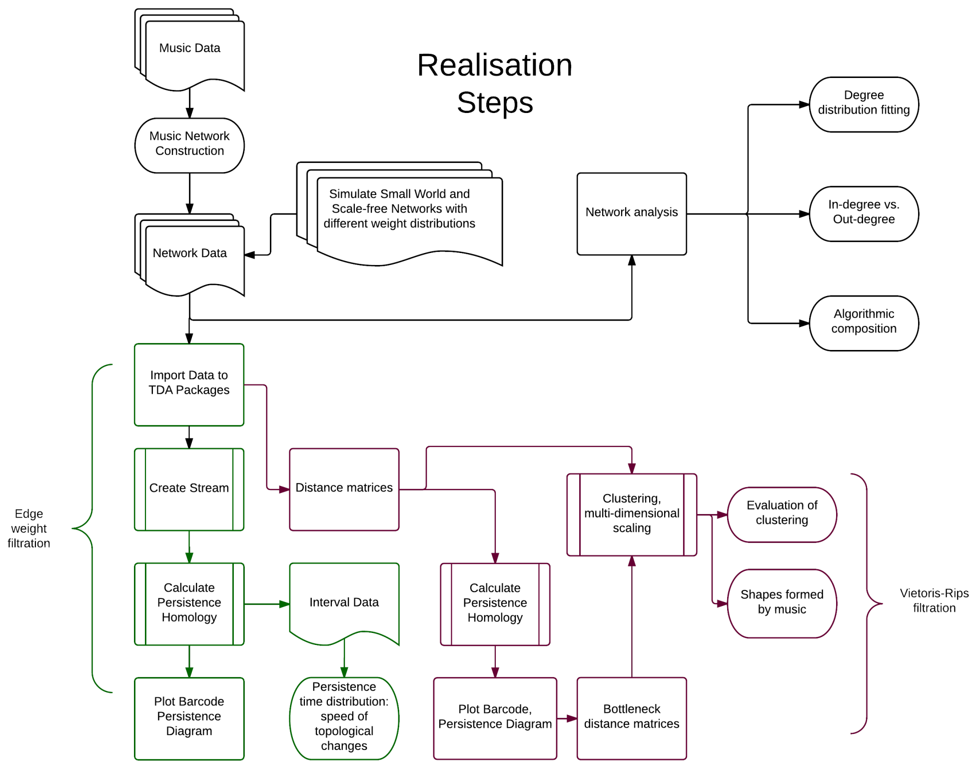


Fig. 2. A flow chart of our realisation

This section is dedicated to our realisation procedures. The process starting from the music data to the results is show in the flow chart: Fig. 2. After extracting music information using the `minidom` package, we create and analyse the network using [20]. Degree distributions are fitted to power-law and exponential distributions using the package provided in [24]. We also plot in-degree vs. out-degree figures using [20]. Designed random-walks are created using the networks to generate music by employing the package [25].

Then, we output the network adjacency matrices and node names, and import them into Matlab and R for TDA analysis. In addition, random, complete, scale-free and small world networks of different parameters with different weight distributions are created and processed in the same way for later use.

Next, we use Javaplex [26] to compute the results of edge weight filtration. In Javaplex, the filtered simplicial complexes are also called streams. We first add all nodes (vertex) to the stream (filtration). The edges are added as seen once before in Fig. 1. Subsequently, we compute the intervals of persistence, then store them in Matlab matrices form. We also output the annotated intervals, which tell us what are the nodes in the intervals of persistence. For example, in the dimension one case, the annotated intervals are consist of the components in the loops generated in the process of filtration. Finally, we plot barcodes, persistence diagrams and persistence time distributions based on the intervals of each network.

The distance matrices are calculated using Python, and then imported into R to create Vietoris Rips filtration and for another analysis purposes. As done in Javaplex, we also calculate the persistent homology, plot the barcode and persistence diagram. The package we use is a new TDA package in R [27] linking many other existing TDA package. The Vietoris Rips filtration process can be computationally costly when go up to dimension two, and that is why we have some second dimensional results on the 100 nodes simulated networks, but not on the 400-1000 nodes music networks. In comparison to the Javaplex, we found many advantages of the TDA package in R, including costing less memory and does not have a pre-defined resolution.

We also accompany the plot with multi-dimensional scaling results from the distance matrices to give a 2-d visualisation. After, we measure the distance between the persistence diagrams using bottleneck distance. So far, we have obtained two layers of distance matrices: the first from the nodes in network (notes in music), the second from between-network (between-music). The multi-dimensional scaling and clustering methods can be used in both layers. Within the network, the multi-scaling gives us a map showing how close the nodes are and clustering gives us a suggestion of a division of community in the networks; between the network, the multi-scaling visualise how close the networks are to each other and clustering suggests which networks are of the same type.

Finally, we use package [28] to calculate the Jaccard clustering similarity and evaluate our clustering results against random guesses, and give visualisations by the corresponding multi-dimensional scaling results.

6 Results

6.1 Typical Network Results

Barcodes and the speed of topological change We start with establishing a series of baseline results of the small world networks. The obtained barcode results for the edge weight filtration are shown in Fig. 3.

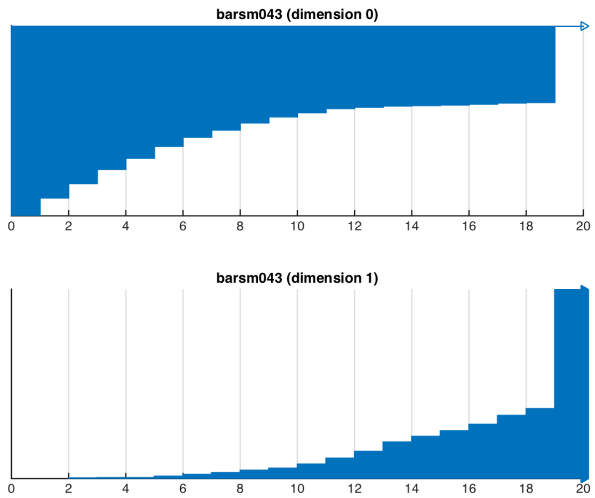
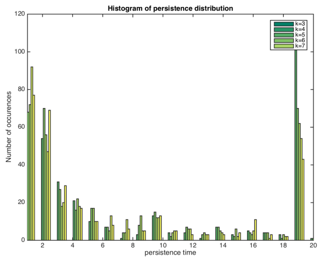


Fig. 3. Barcodes of small world network with 100 nodes. Each node is connected to 4 nearest neighbors in a ring topology. For each edge u-v in the underlying "100-ring with 4 nearest neighbors" with probability 0.3 add a new edge u-w with randomly-chosen existing node w. The edge weight distribution is the uniform distribution with the maximum value chose to be 20. As introduced in section 3, the barcode indicates the number of components (top) and the number of loops (bottom) changing with filtration time.

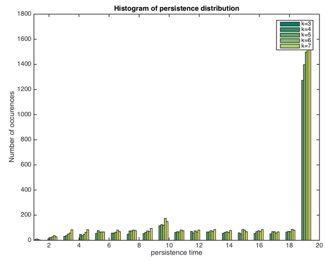
The trends of the barcode growth are straightforward from our filtration introduced in section 3.2: the connected components of the network decrease as the links are added, and the loops in the network increase as more edges are added. The big discrepancy that appears in the one dimensional barcode between $t = 19$ and $t = 20$ can be explained as the sudden increase in the connections between nodes, which is caused by the constraints introduced by the edges which have been already added.

As we can observe in Fig. 3, because our calculations are based on large networks, the barcodes are densely packed. This gives a problem that, having tested different parameter configurations of the small world networks, they all looked the same at the first glance. Mostly because, as expected, in dimension zero and one, they should have similar topological structures but just with minor differences. Therefore, as we need better representation to compare between the networks of different parameters, the persistence time distribution construct introduced in section 3.6 become natural to use.

In Fig. 4(a) and 4(b), the persistent time distribution of the length of intervals are plotted. Each color corresponds to a varying number of the connected nearest neighbors and re-wiring probability. Again: the height of each bar can be interpreted as the thickness of each barcode layer, since they are the count of how many intervals are there (the number of the blue lines in the barcode). The connections with the barcode results are: in the $dim = 0$ case, the persistence times on the x axis is essentially the times of death of the features. Similarly, the x axis represents the times of birth in the $dim = 1$ case.



(a) Distribution histogram of the number of occurrences of a certain persistence time of Betti number 0 in small world networks, with varying number of connected nearest neighbors k and probability 0.9 of adding the new edges. The edge weight distribution is the uniform distribution with the maximum value chose to be 20

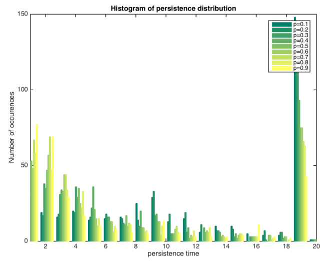


(b) Distribution histogram of the number of occurrences of a certain persistence time of Betti number 1 in small world networks, with the same parameters as Fig. 4(a)

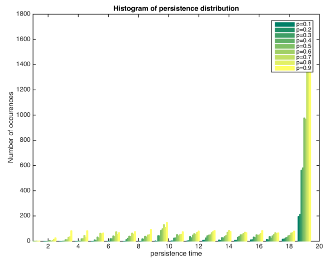
Fig. 4. Distribution histogram of small world networks

Having a closer look at the distribution histograms, we can see they indeed share the same trend and that the behaviors before the last step are quite noisy. Nevertheless, it is clear that, first, we have a "swing" of the thickness of the barcode in dimension zero, which means as we lower the threshold of linking, the disappearances of individual components slows down. The fact the last step is steep suggests that consideration of the smallest weighted edge is important, since in this case, it can have significant effects on the change of topology. Second, with a larger number of connected nearest neighbors in the ring topology, we have a bigger discrepancy in the last step in dimension one, and smaller discrepancy in dimension zero. This implies that, when adding the smallest weighted edges, the speed of holes generation is faster when there are more connections within the ring topology, and on the other hand, the speed of making connections is slower.

To have a more comprehensive understanding of the parameters, we also plotted the parallel results for varying probability of adding new random edges with seven connected nearest neighbors in a ring topology, as shown in Fig. 5(a) and Fig. 5(b).



(a) Distribution histogram of the number of occurrences of a certain persistence time of Betti number 0 in small world networks, with fixed number of the connected nearest neighbors = 7, and varying number of probability of adding the new edges. The edge weight distribution is the uniform distribution with the maximum value chose to be 20



(b) Distribution histogram of the number of occurrences of a certain persistence time of Betti number 1 in small world networks, with the same parameters as Fig. 5(a)

Fig. 5. Distribution histograms of small world networks

Similar to the diagrams with varying number of the nearest neighbors, we observe the same trend and discrepancies at the last step, which leads us to the conclusion that a larger probability of rewiring gives a better connection of the network. It is still hard to tell if there is a generalisation of the results of the changing parameters of dimension zero. Nonetheless, we can see some interesting behavior in dimension one, where it is clear that the bigger the rewiring probability, the larger the speed of the generation of loops at the same level.

Multi-dimensional scaling and persistence diagrams In this section, we present the results from Vietoris-Rips filtration. As shown in Fig. 6, we observe the ring topology from the small world network and the hubs in scale free network, which verifies our distance construction. Moreover, the corresponding persistence diagrams are showing their advantages in summarising these topologies concisely.

Further comparing between Fig. 6(c) and Fig. 6(d), we observe that the edge weight distribution has a significant effect on the distance space of networks: although preserving the ring topology to a certain degree, it creates outliers. Comparison between Fig. 6(c) and Fig. 6(b) tells us the rewiring process eliminates the outliers and homogenised the network.

In addition to bring light on the interdisciplinary subject of topological data analysis and complexity science, we will be using the results of this section to establish some properties of music networks and their uniqueness.

6.2 Music Networks

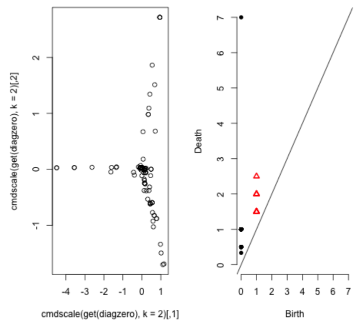
In this subsection, we will show all the results we obtained from performing our network and TDA analysis. Mainly can be divided into creation and analysis, the analysis part includes three layers, which are

- Low Layer: Pitch, duration, etc. (Section 6.2 and Section 6.2)
- Mid Layer: Tonality, phrase, melody, etc. (Section 6.2)
- High Layer: Composer, genre, emotion, etc. (Section 6.2 and Section 6.2)

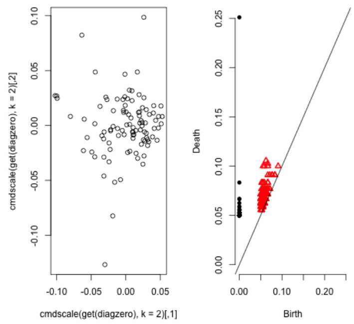
The creation part is in section 6.2. We also mention a potential education application in section 6.2.

Network degree analysis One typical degree distribution is shown in Fig. 7. Number one in the legend denotes violin I, two denotes violin II, three denotes viola and four denotes cello. These will stay the same whenever there are only four numbers in the legend throughout the paper. When there are more than four groups of data, they will be indicating different piece numbers instead of instruments. It is hard to tell if they are long-tail or exponential distributions. We also observe similar in-degree and out-degree distributions, as shown in Fig. 8.

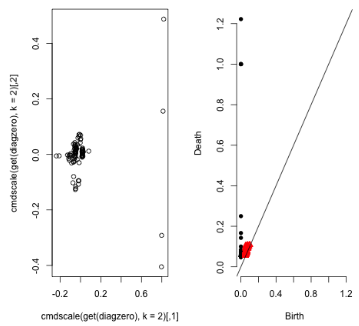
To scientifically decide which distribution is a better fit to our music network, we employ the statistical framework for discerning and quantifying power-law behavior in empirical data mentioned in [24]. This method combines the maximum-likelihood fitting methods with goodness-of-fit tests based on the Kolmogorov-Smirnov statistic and likelihood ratios. One typical result of fitting is shown in Fig. 9. Combined with likelihood ratio values, it is obviously that the exponential distribution is a better fit in comparison to the power law distribution.



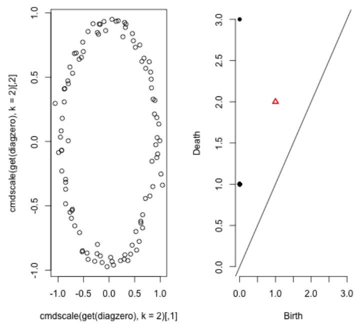
(a) Multi-dimensional scaling and the corresponding persistence diagram of scale-free network with $\alpha = 0.2$, $\beta = 0.3$ (details in section 2.2). Edge weights are all equal. We see the presence of hubs in the scale free network.



(b) Multi-dimensional scaling and the corresponding persistence diagram of small world network with fixed number of the connected nearest neighbors = 8, and rewiring probability = 0.9. The edge weight distribution is the uniform distribution with the maximum value chose to be 20. With a large rewiring probability and uniform edge weight distribution, we see homogeneous structures of the network.

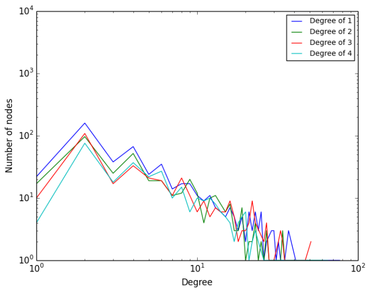


(c) Multi-dimensional scaling and the corresponding persistence diagram of small world network with fixed number of the connected nearest neighbors = 8, and rewiring probability = 0.1. The edge weight distribution is the uniform distribution with the maximum value chose to be 20. With a small rewiring probability and uniform edge weight distribution, we see outliers in the multi-dimensional scaling.

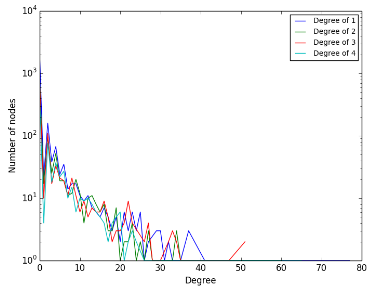


(d) Multi-dimensional scaling and the corresponding persistence diagram of small world network with fixed number of the connected nearest neighbors = 8, and rewiring probability = 0.1. Edge weights are all equal. With a small rewiring probability, we see the ring topology which was put in when constructing the network.

Fig. 6. Multi-dimensional scaling (a visualisation of distance spaces) and persistence diagrams of small world and scale-free networks' distance spaces

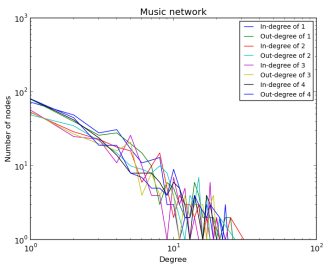


(a) Degree distribution of Beethoven String Quartet No. 09 in C major Opus 59, indicating power-law-like structure

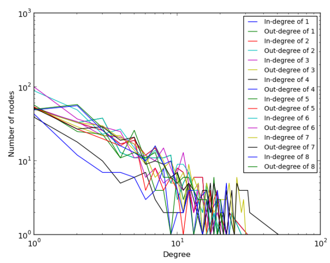


(b) Degree distribution of Beethoven String Quartet No. 09 in C major Opus 59, indicating exponential-like structure

Fig. 7. Degree distributions plotted in y-log and log-log coordinates. It is hard to determine by eye if it should be power-law or exponential degree distribution.



(a) In/out degree distribution of Beethoven String Quartet No. 05 in A major Opus 18



(b) In/out degree distribution of the viola part of eight pieces of Beethoven String Quartets

Fig. 8. In/Out degree distribution of different pieces and instruments

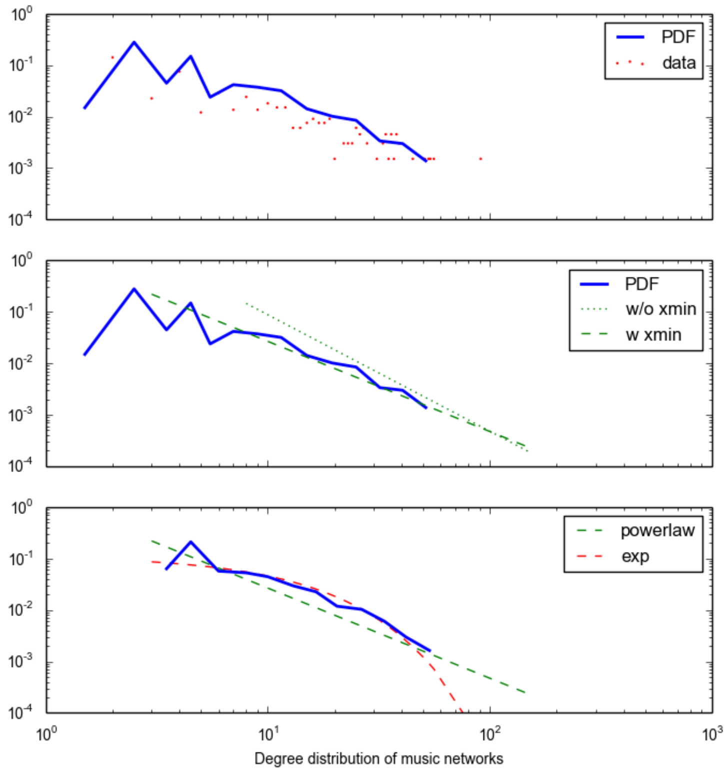


Fig. 9. Fitting of degree distribution.

These exponential-like distributions suggests that there are very important notes in the music which are indispensable. Without these notes, the structure of the music would be destroyed. Moreover, in accordance with music theory, the high degree nodes are mostly the rest notes and tonal notes in the piece, as shown in Table 1. The results also entail that a network can recognise tonality of a music quite easily.

First violin		Second violin	
Degree	Note	Degree	Note
88	Rest, 0.3	77	Rest, 0.3
77	Rest, 0.6	61	Rest, 0.6
51	F6, 0.3	46	F5, 0.3
44	E6, 0.3	43	C6, 0.3
42	D6, 0.3	39	A5, 0.3
38	C6, 0.3	38	C5, 0.3
37	G6, 0.3	37	E5, 0.3
37	B6, 0.3	37	D5, 0.3
36	F6, 0.6	35	G5, 0.3

Table 1. Table of important nodes in the network of Beethoven String Quartet No.01 in F Major

Also, since we do not see the exponential distribution in mere pitch statistics nor duration alone [29], we can conclude that the pitch and duration recover the most essential parts to music.

Notably, the exponential degree distribution has been rated as the "more normal than normal" network degree distribution, some examples include the network of power-grid, email users relationships [30] and so on.

We have also plotted in-degree against out-degree in Fig. 10, which is largely linear in normal scale. This is suggesting the homogeneity of the music network. Because a melody such as "C-E-F-C-E-G-C-E-D" with recursions can produce a small network with one node has 3 out-degree and 1 in-degree, which is a counter-example to our case. That is, this kind of small melody is not explicit shown in the network, and the network is encoding a more homogeneous structure of music.

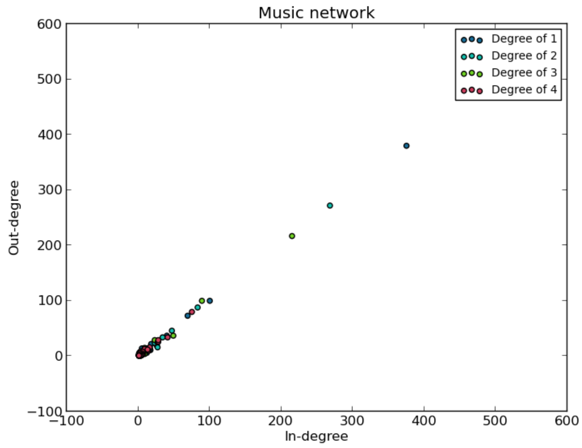


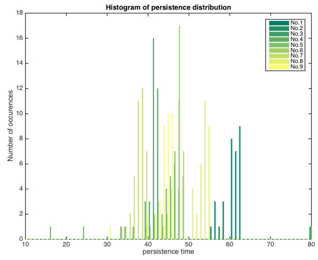
Fig. 10. In- vs. Out-degree.

Creating music using network Since the original music is a predetermined walk within the music network, it is natural to consider a design path in the networks to generate music. Also applied in [13], we generated music by taking a weighted walk in the network using edge weights, and artificially avoiding gap between pitches larger than an octave. One exercept is shown in Fig. 11. Although having the same network structure as a Haydn string quartet piece, the generated music is more contemporary-like music, which remind us the importance of other music structures other than mere connections: chords progression, phrases of music, global rhythm, etc.

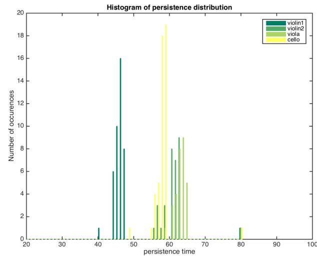
The speed of topological change and difficulty of pieces In Fig. 12(a) and Fig. 12(b), the distribution histograms are of similar constructs as the small world networks, with the parameters changed to the different No. of music pieces and different instruments. Compared with results from random (small world networks with large rewiring probability), complete (small world networks with large connected neighbors), scale-free and small world networks, the differences between the persistence are clear. First, we do not have the continuous persistence spread over time. In contrast, we sometimes have outliers, and most of the time, the bars show concentration around a certain value. Second, the counts of a certain persistence times are significantly less than the ones in the small world networks, although we have almost the same, and sometimes even more nodes than the small world networks. This shows us there are certain topological patterns in music networks that are not shared by random, complete, scale-free and small world networks.

Fig. 11. Music generated by using the music networks. With seed music being the second violin of the second movement of String Quartet No. 57 in C Major, Op. 74, No. 1, FHE No. 28, Hoboken No. 72

In addition, these figures can also provide a complexity measure of the music: when we have higher bars in large x-value, it indicates more presence of weak connections, which means more variety in music and can be harder to practice in terms of repetition.



(a) Distribution histogram of the number of occurrences of a certain persistence time of Betti number 0 in music networks, featuring the second violin with different music



(b) Distribution histogram of the number of occurrences of a certain persistence time of Betti number 0 in music networks, featuring Beethoven String Quartet No.1 with different instruments

Fig. 12. Distribution histograms of music networks

Annotated intervals In addition to the statistics we have done with the persistence intervals, one more result from the edge weight filtration is the annotated intervals. Simply put, as introduced firstly in section 5, the annotated intervals tell us what the nodes in the intervals of persistence are. For example, in the dimension one case, the annotated intervals are consist of the components in the loops generated in the process of filtration.

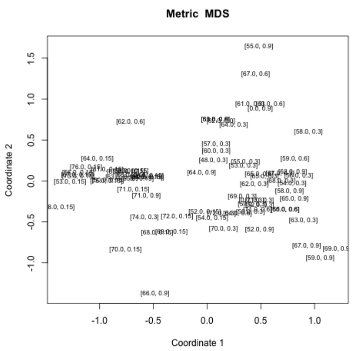
An excerpt is shown as follows (The musical notation is replaced from numerical notation we originally have):

- $[78.0, \infty)$: [F4,G4] + [0,G4] + [C4,F4] + [0,C4]
- $[78.0, \infty)$: [0,C3] + [B2,C3] + [B2,C4] + [0,C4]
- $[78.0, \infty)$: [0,B3] + [G3,B3] + [G3,D4] + [C $\sharp$4,D4] + [0,C $\sharp$4]

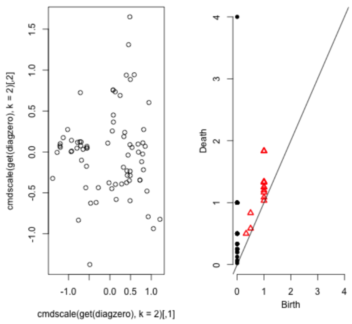
The intervals before the column are the time intervals of our persistent homology, followed by the actual nodes which constructed the intervals. This tells us the combination C-F-G (C sus4), G-B-C $\sharp$-D and B-C-C are of importance in the music. For each instrument in each piece of music, we obtain hundreds of such data, which might be useful for future research in algorithmic composition.

Multi-dimensional scaling, note maps, and persistence diagrams Here, we present the multi-dimensional scaling and persistence diagram results on the note level. As shown in Fig. 13(a), we can give note maps of music. It is essentially a point cloud of music notes with a metric defined in section 3.2. Although they are overlapping in the figure here, we have made it zoomable with R. This shape formed by an entire piece of music can be interesting for further musicology study.

In Fig. 13(b), we show the original point cloud without the labeling as in Fig. 13(a). Although we have concluded that the degree distribution is not power-law, which defines a scale-free network, we can observe from this figure that, in the distance space we constructed, there are properties shared by scale-free networks and music networks, as shown in Fig. 13(b) and Fig. 6(a).



(a) The multi-dimension scaling on the note level: it forms a note map of the whole piece.



(b) An exemplary persistence diagram and a raw multi-dimensional scaling figure, which will be used to calculate the bottleneck distance and gives clustering results in the next section

Fig. 13. Multi-dimensional scaling and the note map of Haydn String Quartet, No.2, viola

Clustering and its evaluation After obtaining the persistence diagram on the note level as discussed in the last section, we now use the bottleneck distance of persistence diagrams to try to classify the music. We used the hierarchical clustering and k-medoid clustering introduced in section 4. Fig. 14 is showing the results from the hierarchical clustering calculated in the dimensional zero case. We observe that it shows distinct classification of the shape of the music.

To evaluate the classification in terms of instruments and composers, we calculate the clustering similarity (4.3) of each clustering method and dimension.

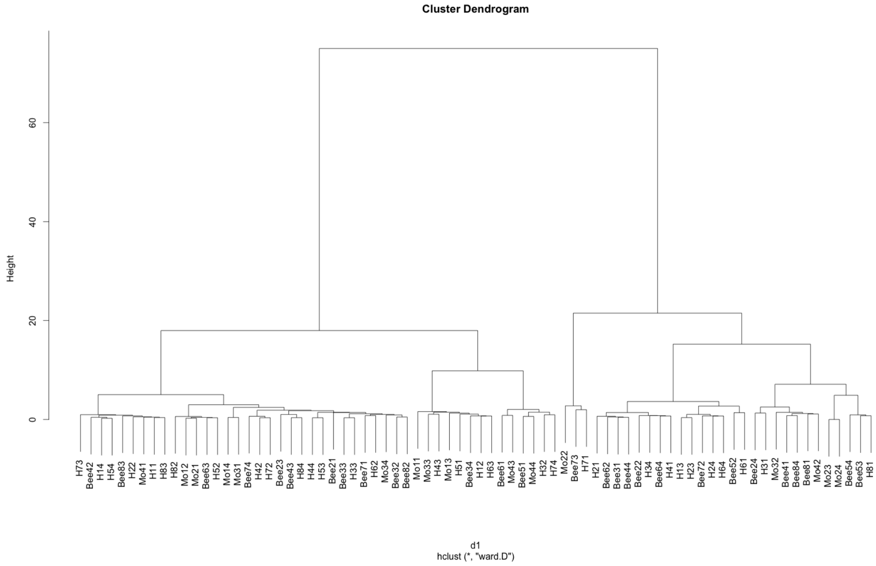
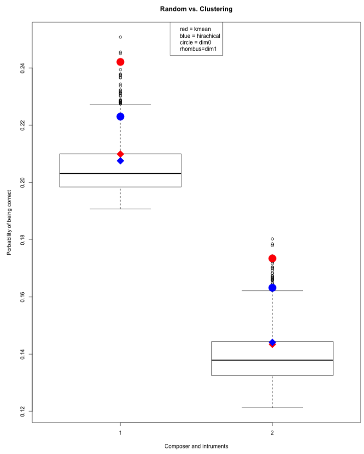


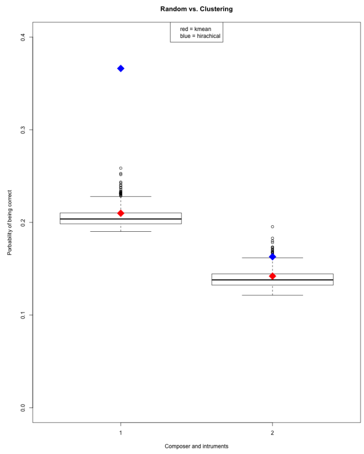
Fig. 14. The result of hierarchical clustering

The results are shown in Fig. 15(a). Some observations can be made, for example, the topological data analysis gives a better results when clustering by different composers than by different instruments, by dimension zero than by dimension one, mostly by k-medriod than by hierarchical methods. In Fig. 15(b), we are making a comparison between our topological data analysis results with a matrix norm taken from the distance matrix on the note level. Although it is doing better when applying hierarchical clustering to the composer clustering, the TDA results still prevail in other configurations.

In general, our results are better than the random guesses but not satisfactory comparing to the music "truth". However, we should take into consideration that we are comparing between Haydn, Mozart and Beethoven, whose relationships have been known as "Beethoven imitating Mozart, and behind Mozart, Haydn". Without knowledge for each piece, it can be hard for musicians to distinguish between them, too. Also, in a string quartet, although it is easy for the human ear to distinguish between different instruments, without the timbre of each instrument as is the case in our construction, it can be hard to identify between instruments, too, solely based on pitch and duration. Therefore, our results, which are based on a very limited data set, although are not as accurate as big-data cross-genre music classification, produce distinctive insights based on the shape of music formed by its notes.



(a) topological data analysis clustering similarity results. Composers-wise (clustering into three groups) is on the left, instrument-wise (clustering into four groups) is on the right). The box graph is the results of 1000 random guesses. As shown in the legend, the red points are the results of the k-medoid clustering, and the blue points are the results of the hierarchical clustering; the circle points are the results of dimension zero, and the rhombus points are the results of dimension one.



(b) Matrix norm clustering similarity results. The box graph is the results of 1000 random guesses. As shown in the legend, the red point is the k-medoid clustering, and the blue point is the hierarchical clustering.

Fig. 15. Clustering evaluation

7 Discussion: Future Work and Applications

In this project, we gained our results by implementing network and TDA methods on a very limited dataset. Obviously, more research can be done using more and different genres of music data. Also, if we keep track of the information from more dimensions, such as dynamics and timber, it might show the real power of persistent homology in high dimensional spaces. From a network and statistical point of view, we can also continue examining more implications of the statistics, and verify them by testing on more pieces of music using ensemble averaging. The extensions of the TDA methods are numerous, too. For instance, we can apply different edge metrics rather than the weights, different thresholding methods, and different dimensions of filtering quantities [5].

As for applications, the results of this project have the potential to assist algorithmic composition and music analysis, and be used for educational purposes and to evaluate the complexity of pieces. By combining the annotated intervals obtained with certain inorganic algorithms, such as genetic algorithms, cellular automata, chaos self-similarity theory, and artificial neural networks mentioned in [31–33], we might be able to improve and bring life to the current research. Furthermore, among many other possible applications of network and topological data analysis, we can exploit the analogy often draw between music and language [34]. Due to presence of the famous power-law: Zipfs law, it is very likely that the network of language can show similar behaviours as the musical ones, and therefore such an investigation might bring more interesting insights into the interdisciplinary study of art and science.

8 The References Section

References

1. Terry Allen and Camille Goudeseune. Topological considerations for tuning and fingering stringed instruments. *arXiv preprint arXiv:1105.1383*, 2011.
2. Christina Anagnostopoulou, Miguel Ferrand, and Alan Smaill, editors. *Music and Artificial Intelligence*, volume 2445 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, August 2002.
3. Athanase Papadopoulos. Mathematics and group theory in music. July 2014.
4. Complex network. `http://en.wikipedia.org/wiki/Network_science`.
5. Giovanni Petri, Martina Scolamiero, Irene Donato, and Francesco Vaccarino. Topological strata of weighted complex networks. *PloS one*, 8(6):e66506, 2013.
6. Dmitri Tymoczko. *A geometry of music: harmony and counterpoint in the extended common practice*. Oxford University Press, 2011.
7. Kenneth Jinghwa Hsü and Andreas J Hsü. Fractal geometry of music. *Proceedings of the National Academy of Sciences*, 87(3):938–941, 1990.
8. Dmitri Tymoczko. The geometry of musical chords. *Science*, 313(5783):72–74, 2006.
9. R. Budney and W. Sethares. Topology of Musical Data. *ArXiv e-prints*, July 2013.
10. M. G. Bergomi and A. Portaluri. Modes in modern music from a topological viewpoint. *ArXiv e-prints*, September 2013.
11. Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.
12. Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, and Clément Maria. Introduction to the r package tda. *arXiv preprint arXiv:1411.1830*, 2014.
13. Xiao Fan Liu, Chi K. Tse, and Michael Small. Complex network structure of musical compositions: Algorithmic generation of appealing music. *Physica A: Statistical Mechanics and its Applications*, 389(1):126 – 132, 2010.
14. Nicole B Ellison, Charles Steinfield, and Cliff Lampe. The benefits of facebook friends: social capital and college students use of online social network sites. *Journal of Computer-Mediated Communication*, 12(4):1143–1168, 2007.
15. Nancy K Baym and Andrew Ledbetter. Tunes that bind? predicting friendship strength in a music-based social network. *Information, Communication & Society*, 12(3):408–427, 2009.

16. Nicholas J Bryan and Ge Wang. Musical influence network analysis and rank of sample-based music. In *ISMIR*, pages 329–334, 2011.
17. Shalev Itzkovitz, Ron Milo, Nadav Kashtan, Reuven Levitt, Amir Lahav, and Uri Alon. Recurring harmonic walks and network motifs in western music. *Advances in Complex Systems*, 9(01n02):121–132, 2006.
18. Florian Gomez, Tom Lorimer, and Ruedi Stoop. Complex networks of harmonic structure in classical music. 438:262–269, 2014.
19. Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.
20. Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA, August 2008.
21. Allen Hatcher. *Algebraic topology.* 2002.
22. David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.
23. Pierre Legendre and Loic FJ Legendre. *Numerical ecology*, volume 24. Elsevier, 2012.
24. Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.
25. Michael Scott Cuthbert and Christopher Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. 2010.
26. Andrew Tausz, Mikael Vejdemo-Johansson, and Henry Adams. Javaplex: A research software package for persistent (co)homology, 2011.
27. Brittany T. Fasy and Fabrizio Lecci. *TDA: Statistical Tools for Topological Data Analysis*, 2014. R package version 1.1.
28. John A. Ramey. *clusteval: Evaluation of Clustering Algorithms*, 2012. R package version 0.1.
29. Tuomas Eerola and Petri Toiviainen. Midi toolbox: Matlab tools for music research. *University of Jyväskylä, Jyväskylä, Finland*, page 14, 2004.
30. Danijela Horak, Slobodan Maletić, and Milan Rajković. Persistent homology of complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(03):P03034, 2009.
31. J. D. Fernandez and F. Vico. AI Methods in Algorithmic Composition: A Comprehensive Survey. *ArXiv e-prints*, February 2014.
32. Gerhard Nierhaus. *Algorithmic composition: paradigms of automated music generation.* Springer, 2009.
33. Jeremy Leach and John Fitch. Nature, music, and algorithmic composition. *Computer Music Journal*, pages 23–33, 1995.
34. Aniruddh D Patel. Language, music, syntax and the brain. *Nature neuroscience*, 6(7):674–681, 2003.

Synchronization in Music Group Playing

Iris Yuping Ren[1], René Doursat[2], and Jean-Louis Giavitto[3]

[1] Erasmus Mundus Master's in Complex Systems Science, Graduate School, Ecole Polytechnique, Paris, France
[2] Complex Systems Institute, Paris Ile-de-France (ISC-PIF), CNRS (UPS3611), Paris, France
[3] Institut de Recherche et Coordination Acoustique/Musique (IRCAM), CNRS (UMR9912), Paris, France
yuping.ren.iris@gmail.com

Abstract. In this project, we created an agent-based model of music group playing under four different interaction mechanisms. Based on real music data, added randomness and simplifying assumptions, we examine how agents synchronize and deviate from the original score. We find that while music can make synchronization complex, it also helps reducing the total deviation. By studying the simulation process, several conclusions on the relationship between different growing speeds of total deviations and different interaction schemes are drawn. With interpretation from a musical point of view, we find that, in a music ensemble, listening to neighbors helps the players end up in sync. However, if people do not listen carefully enough, the deviation becomes larger than when people do not listen at all. On the issue of whom one should listen to, the results show no significant differences between listening to the immediate neighbors and to the whole group. Finally, we also observe that large deviations can be reduced by making the musicians move while playing.

Keywords: synchronization, collective behavior, agent-based modeling, deviation, music playing

1 Introduction

Many questions have been asked about the rhythmic complexity of music. Is it more difficult to synchronize over a melodic rhythm or a drum beat? Is it better to listen to people around you or just play as written in a music ensemble? How can we obtain better synchronization? Several research papers and books have addressed synchronization problems in biological and social/human interaction systems [1–5], but few have answered this line of questions. In this project, we simulate music ensembles using agent-based models, a method known for its ability to produce complex behaviors from simple rules. Although it is not possible for simple models to accurately represent every interaction among musicians, it is still possible to gain valuable insights from abstractly simulated music ensembles.

Two important concepts embedded in this project are derived from the well-known "firefly" model of synchronization [5]. Like this model, we define *phase*

Fig. 1. Initial configuration of the music group program, with a conductor symbolized in red and four different groups of musicians in "white", "green", "yellow", and "blue".

and *frequency* variables to characterize the system. Herre, the frequency of each agent will be called "tempo" and the phase lag, the "the waiting time". Important differences with the firefly model are the incorporation of actual music data and conditional interactions between musicians. Another important concept is the unavoidable deviation of the played stream from the written music, which has been investigated in [6–8] and experimentally proven. Although we do not have such a small time resolution, the implementation can be justified with amateur music players.

2 Model

In this model, we use real music data in the form of duration datasets (without pitch), extracted from Beethoven's quartets. The players follow these durations and different interaction schemes among themselves. We simulate music ensembles consisting of four sections, "white", "green", "yellow", and "blue" (Fig. 1), so that we can observe the differences between schemes applied inside each section. Musical interactions between two sections are ignored for simplicity. Some amount of spatial interaction between players will be introduced at a later stage.

2.1 Parameters

The parameters of the model are the following (Table 1):

- Number of agents: how many agents there are in one musical section.
- Music sheet: 10 different music datasets (rhythmic parts only, no pitch); 1-8 are the Beethoven string quartets Nr. 1-8; 0 and 9 are drum beats with intervals of 1 and 3 seconds.
- Avg freedom: mean of the freedom of agents (with default standard deviation, modifiable from the program itself).
- Tempi std: standard deviation of the tempi of agents (with default mean value, modifiable from the program itself).

Parameter	Range	Notation
Number of agents	1-28	x
Music sheet	0-9	N/A
Avg freedom	1-100	F
Tempi std	0-20	α
Max reaction	0-100	R
Confidence	0-8	C
Waiting resolution	1-1000	N
With/Without Conductor	true/false	–
Move/No move	true/false	–

Table 1. Table of model parameters with the range of acceptable values and mathematical notations

- Max reaction: maximum value of the reaction skills of agents (where actual skill is a random integer number under this cap, modifiable from the program itself).
- Confidence: how many actively playing neighbors one musician must have, in order to be confident that s/he is playing at the right time.
- Waiting resolution: a normalizing factor controlling in part how much time resolution a musician has.
- With/without conductor: this is just for the yellow group; the tempo will be set uniformly to 100 if this is on and the players became aware that they are playing "wrongly".
- Move/no move: agents will move randomly if this is on, as shown in Fig. 2. Their neighbors will therefore also change.

To have a concrete view of the effect of these parameters, we explain the dynamics of the model in the next section.

2.2 Dynamics

The mechanism used to synchronize the musicians is based on the music. For every note duration in the dataset, we approach it using a timer, which is reset at the beginning of every step. The value of the timer is denoted by $t(i)$, where i is the step number in the process, which is equal to the number of duration values in the dataset. Then, once the timer's value and the note's duration $m(i)$ are sufficiently close, we ask the agents, which are by default in color gray, to change to the color belonging to their group (white, green, yellow, blue), hence achieve an effect of "playing" the event. We will also use the word "recoloring" to denote music playing. We denote each agent by x, as mentioned in the parameter table. For describing the relation between a parameter and the turtles-owned value controlled by it, we use a functional notation. For example, each turtle's reaction skill will be denoted by $R(x)$. Considering all the parameters we used above, this part of the dynamics can be expressed as:

$$\text{if } m(i) - \frac{\alpha(x)}{100} \times t(i) > \frac{F(x)}{N}, \text{ wait for } R(x)/N, \text{ and set } \mathit{color} := \mathit{gray},\ t(i) := 0$$

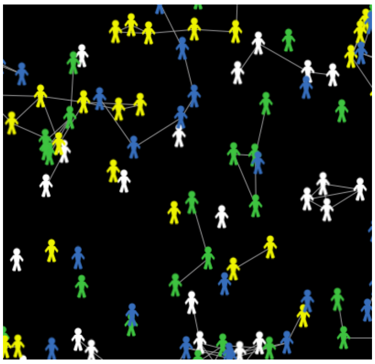


Fig. 2. Typical motion dynamics.

otherwise, recolor. The next time the agent becomes gray can happen at the next step, when the timer discovers that there is still a certain amount of time until the end of the next duration.

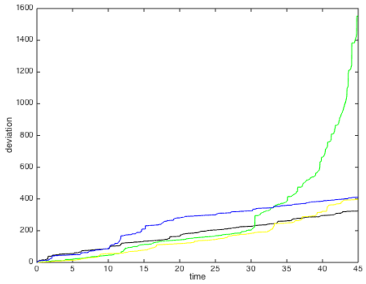
Finally, we add interactions among players to the process, asking agents to "look" whether there are enough players around them who are playing. If that number is larger than the confidence level of a player, C, then s/he must change the tempo according to the mean of the active neighbors, denoted by $\{x_k\}_{k\in[0,28]}$ (explained in detail in the next paragraph). We denote the number of the gray linked-neighbors by n_x, and write this part of the dynamics:

$$\text{if } n_x \geq C,\ tempo(x) := \overline{tempo(x_k)}$$

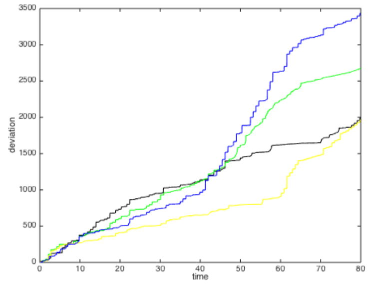
After all this decision making, we record the actual difference between the waiting time and the duration, and plot this deviation. Differences between the four group reside in how they react to other players' tempi, i.e. the differences between the $\{x_k\}$:

- Players in the white group listen to other neighboring white players and take the mean tempo from them.
- Players in the green group listen to other neighboring green players, but follow a normal distribution whose mean is equal to the average tempo of the neighbors.
- Players in the yellow group have two choices: when the conductor option is on, they sync to the conductor, i.e. adopt a uniform tempo; otherwise, they listen to all other players in all groups.
- Players in the blue group listen to all other blue players and take the mean tempo from them.

We also introduce a motion dynamics, while the "Move" option is on, we ask the players to move randomly, including changes in their links; that is, their neighbor will change according to where they are.



(a) Time series of the total deviations of the white (shown in black), green, yellow and blue groups, featuring the large deviation of the green group. Other groups have similar lower total deviations. The “conductor switch” for the yellow group is on. Other different growing patterns between the white, yellow and blue groups are caused by the specificties of the music at hand.



(b) Time series of the total deviations of the four groups when musicians are moving. Here, the blue group is strongly influenced by the bad tempi of the green players. In other runs, the group that gets most influenced might change. In general, however, there is no outlier curve of total deviation like the green one in (a).

Fig. 3. Time series of total deviations: (a) static players; (b) moving players.

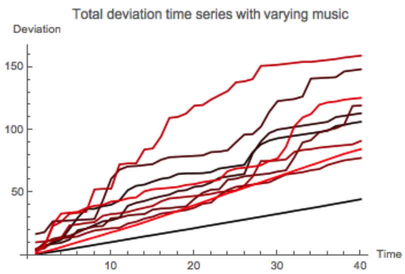
2.3 Statistics

The following statistics are used to measure the outcome of our model:

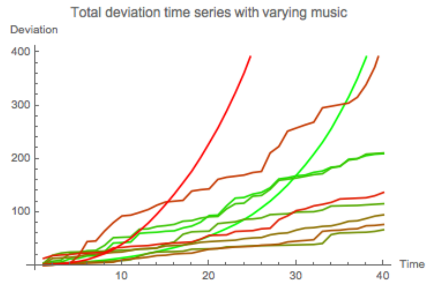
- Each group’s total deviation from the music, called “total deviation 1”, etc.
- Each player’s deviation from the music (because the total deviation loses the information about whether individual players are lagging or leading).
- The tempo distribution of the players over each group; synchronization among players can be observed when these distributions converge.
- The deviation distribution of the players; most are centered around zero, others account for the cumulative deviation that we show in the total deviation window.

3 Results

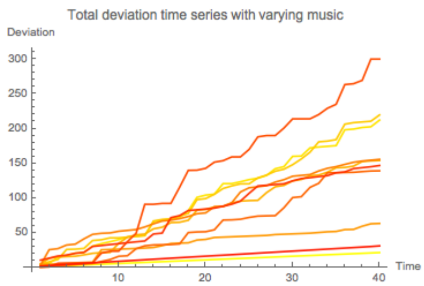
In the beginning of the simulation, tempi are scattered in all four groups, and total deviations grow with time in a similar manner. We can also see the convergence of tempi in certain groups. After observing the process for a while, we find different growing speeds of the total deviation between different groups. The green group exhibits a particularly big deviation as shown in Fig. 3(a). After running for a period of time, the program slows down. This should not matter



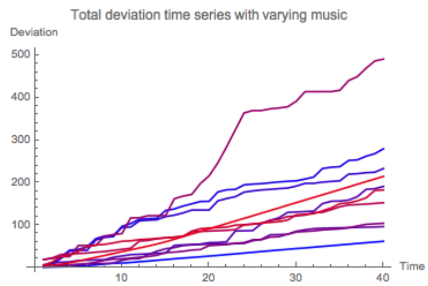
(a) Time series of the total deviations of the white group for different music pieces. Real music is adding complexity to the drum-beat music. Listening to neighbors results in larger deviation than the group with the conductor for drum-beat music, but smaller deviation for real music.



(b) Time series of the total deviations of the green group for different music pieces. Compared to the Beethoven music, the drum-beat music leads to fast growth in deviation. The complexity of music prevents the generation of deviation upon deviation. The curve with larger curvature is the drum beat of 3-second intervals.



(c) Time series of the total deviations of the yellow group for different music pieces. Real music is adding complexity to the drum-beat music. With the conductor, the drum-beat music has the least deviation. The linear growth rates are close, also resulting from having a conductor in lead.



(d) Time series of the total deviations of the blue group for different music pieces. The linear growth of the drum-beat music is at around the same level of the Beethoven music. The linear growth with larger slope is the drum beat of 3-second intervals.

Fig. 4. Total deviation time series of all four groups with different music pieces

much for the project because when deviations become large, the ensemble usually stops playing. However, there are cases when musicians sight-reading new music are not able to know for a while whether they are playing out of step or not. So it is also useful to look at the dynamics for a longer period of time, and record observations of large deviations.

One way to improve on large deviations is by actually making the players move (Fig. 3(b)). By “improve”, we mean that the slopes of total deviations in the four groups are more or less similar, so there is no more single large deviation (as the green curve in Fig. 3(a)), which is the most disruptive to music group playing. This can happen because, as the completely off-beat players encounter the right players, they can exchange information, not through tempo, but just by looking at whether their linked neighbors are playing or not. However, this method has an obvious flip side, which is that bad influence can be proliferating, too, as can be seen in Fig. 3(b).

Given the above results and diagrams, we can already draw musical conclusions such as: listening to your neighbors helps the ensemble end up in sync; furthermore, if people do not listen carefully (as in the green group) the results can be a disaster. In the case of the yellow group, it is safe to say that they should not listen to people who do not listen; instead, they should look at the conductor. Finally, for the blue group, the lesson we can learn is that listening to the whole group or only to your neighbor does not make much difference, therefore it is sufficient to listen to a small number of people around you.

Besides running the simulation and observing statistics under a given set of parameter values, we also explored the "music" parameter axis. The total deviation time series of all four groups with different music pieces are shown in Fig. 4(a)-4(d). There are two regular-looking curves in each graph, because music Nr. 0 and music Nr. 9 are drum-beat intervals of 1 second (the line corresponding to the group color) and 3 seconds (the red line), not music. In the green group case, the growth is fast in comparison with the other linear growth of deviation. We can also see one common feature out of the drum-beat cases: the smaller the intervals are, the easier they are to sync.

Excluding Fig. 4(b), in most of the cases, we can see that music definitely makes it harder for people to minimize their deviation, especially as shown in Fig. 4(c). However, in Fig. 4(b), it is actually helping with a reduction of the total deviation. If we recall the phenomenon of many people trying to clap in a certain tempo but unavoidably just getting faster and faster, this fast-growing curve may bear some resemblance to that phenomenon. A plausible explanation of the seemingly helpful function of music would be that the varying interval lengths are suppressing further growth of the deviation during the process.

4 Conclusion and Future Work

We have presented a model consisting of different mechanisms of synchronization, which was able to tell us some non-trivial facts about music group playing. In future work, we can implement minor modifications such as changing the distribution of different parameters in addition to their values; different neighbor selection strategies can be used, since musicians are not necessarily just listening to their immediate neighbors in the ensemble.

However, the most important factors omitted here are the many musicological nuances which are no doubt used by individual musicians; for instance, the fact that a certain amount of rest in the music will help synchronization, or that off-beat notes are harder to sync, etc., are not considered. Moreover, the model did not account for musical interactions across the four groups, although they clearly influence musical interpretation and synchronization, too. Therefore, we will take introducing musical rules in the agent behavior as a priority in future work. While such projects are mostly based on subjective observation and rather non-exhaustive, they also open the door for more critical inquiry and opportunities for interesting discoveries at the same time.

References

1. Keiko Yokoyama and Yuji Yamamoto. Three people can synchronize as coupled oscillators during sports activities. *PLoS computational biology*, 7(10):e1002181, 2011.
2. Peter J Beek. Timing and phase locking in cascade juggling. *Ecological Psychology*, 1(1):55–96, 1989.
3. Arkady Pikovsky, Michael Rosenblum, Jürgen Kurths, and Robert C Hilborn. Synchronization: a universal concept in nonlinear science. *American Journal of Physics*, 70(6):655–655, 2002.
4. Arthur T Winfree. Biological rhythms and the behavior of populations of coupled oscillators. *Journal of theoretical biology*, 16(1):15–42, 1967.
5. Steven Strogatz. *Sync: The emerging science of spontaneous order*. Hyperion, 2003.
6. H Hennig. Synchronization in human musical rhythms and mutually interacting complex systems. *Proceedings of the National Academy*, 2014.
7. Holger Hennig, Ragnar Fleischmann, Anneke Fredebohm, York Hagmayer, Jan Nagler, Annette Witt, Fabian J Theis, and Theo Geisel. The nature and perception of fluctuations in human musical rhythms. *PloS one*, 6(10):e26457, January 2011.
8. H Hennig, R Fleischmann, and T Geisel. Musical rhythms: The science of being slightly off. *Physics Today*, 2012.