Jonathan B. Moore. Evaluating the spectral clustering segmentation algorithm for describing diverse music collections. A Master's Paper for the M.S. in L.S degree. May, 2016. 104 pages. Advisor: Stephanie Haas

This paper presents an evaluation of the spectral clustering segmentation algorithm used for automating the description of musical structure within a song. This study differs from the standard evaluation in that it accounts for variability in genre, class, tempo, song duration, and time signature on the results of evaluation metrics. The study uses standard metrics for segment boundary placement accuracy and labeling accuracy against these song metadata. It reveals that song duration, tempo, class, and genre have a significant effect on evaluation scores. This study demonstrates how the algorithm may be evaluated to predict its performance for a given collection where these variables are known. The possible causes and implications of these effects on evaluation scores are explored based on the construction of the spectral clustering algorithm and its potential for use in describing diverse music collections.

Headings:

Music libraries

Sound recordings

Information storage & retrieval systems – Audiovisual materials

Library Automation

Signal Processing

EVALUATING THE SPECTRAL CLUSTERING SEGMENTATION ALGORITHM FOR DESCRIBING DIVERSE MUSIC COLLECTIONS

by
Jonathan B. Moore

A Master's paper submitted to the faculty
of the School of Information and Library Science
of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements
for the degree of Master of Science in Library Science

Chapel Hill, North Carolina
May 2016

Approved by

_____

Stephanie Haas

Table of Contents

I. Introduction

As new technologies have opened up new ways to study music, they likewise have created new applications for this research. Among these is Music Information Retrieval (MIR), an interdisciplinary field of study that examines methods of providing access to musical data. Modern research in MIR seeks innovative ways of indexing a digital collection of music that can be applied in search engines, recommendation services, and scholarly databases. These strategies are built upon our ability to describe musical similarity and which songs are similar in certain ways to other songs. There are multiple things one can take into account for this task, including but not limited to the following factors. 1) The bibliographic information that accompanies a piece of music: its title, composer, lyricist, date of composition, etc. 2) The social component: what are the listeners of this piece of music like and how can we use that to predict who else might be interested in the piece of music? 3) The subjective qualia of the music: how might one describe the experience of listening to this piece of music and does that make it more suitable for certain moods or activities? 4) The aural qualities of the music itself: the relationships between notes, harmonies, rhythms, and other qualities that are revealed in the content of the music. The subject of this particular study is musical structure, which fits within that fourth factor and comprises that quality of a piece of music which allows us to identify themes and sections which repeat within the piece. The study of musical structure has long played a major role in music theory and analysis, alongside harmony,

melody, rhythm, and timbre. Structure has historically been a primary component in determining how to classify a piece of music; terms like rondo, fugue, and sonata double as both structural and genre descriptors in the Western Classical tradition, and even more modern genre classifications like *blues* and *pop* often imply a defined structure. This study of musical structure is certainly central to any debate about musical description.

Naturally, we cannot expect every digital collection of music to be structurally analyzed by a music theorist, but there are nascent methods of automating the task. These processes, known as structural segmentation algorithms, are able to parse a digital audio file and discern its repeating sectional components. Usually, these algorithms are evaluated in an adversarial way, i.e. competing algorithms are tested against one collection and ranked according to the which algorithms maximize or minimize the values returned by standardized evaluation metrics that measure, for instance, the accuracy of the placement of sectional boundaries or the agreement between sections that an algorithm determines are alike and those determined by a human expert to be alike.. This study proposes another kind of evaluation using just one subject algorithm against the same standardized evaluation metrics. This method of evaluation is not to determine which algorithm among many receives the highest average evaluation score, but instead to determine what variables within a collection have a significant effect on the evaluation scores that a single algorithm receives.. In this case, I will examine the performance of the spectral clustering algorithm proposed by McFee & Ellis [1] against ground truth segmentations in a collection in which broad genre classification (referred to as class), narrow genre classification (referred to as genre), the division of beats within a bar (referred to as time signature), the frequency of beats (referred to as tempo), and the song

duration for each song are known. . It could be used to identify the weaknesses of an algorithm or to determine particular types of collections to which an algorithm is or is not suited. The basic research questions may then be stated as follows:

1) Do either broad *class* or narrow *genre* of a song have a significant effect on the accuracy of the spectral clustering algorithm for the structural segmentation of that song?
2) Is there a significant correlation between the *duration*, *tempo*, or *time signature* of a song and the accuracy of the spectral clustering algorithm for the structural segmentation of that song?

II. Review of the Literature

As a field of study, MIR rests within the larger field of Information Retrieval (IR). Therefore, its beginnings lie in research first pioneered for text-based documents in traditional search contexts. The earliest music search systems developed under this standard relied on textual metadata for retrieval. Fields that you might find as part of a traditional card catalog formed the basis for access: searching for music by a known composer, or with a known title, or from a known album, or published in a particular year. This model is sufficient for the kinds of basic searches done by reasonably informed users on collections that are well described. Even modern applications largely rely on the prototypical formula. As an example, the music used in this study was purchased from the iTunes store using keyword searches that returned those pieces of music in which the title, artist, or some other field matched the given keywords. From an academic context, where searches might require more rigorous methods of retrieval, consider the description of the catalog of *Naxos Music Online*[1], which maintains a vast and comprehensive controlled vocabulary across dozens of metadata fields to provide effective access to its users. The methods of description vary in complexity and efficiency, but have in common a dependency on a well-trained cadre of catalogers accurately describing large collections. Not only is this a time- and resource-intensive process, it is also one with limited effectiveness outside of bibliographic information. Even trained experts disagree

on the largely subjective terminology of genre and style, and explicitly musical qualities (including tempo, key, or chord progressions) require exhaustive analysis to evaluate even in the few cases where they are unambiguous. As the number of pieces of music continues to grow, and services seek to provide access to ever-larger numbers of them, searching based on some measure of musical similarity using traditional means of musical analysis by experts is just impractical.

One popular workaround is to crowd-source description tasks. Organizations like MusicBrainz[2] draw on the efforts of a community of enthusiasts where many amateur users may each submit their own metadata as tags. In this system, the agreement of many of these users on a set of metadata substitutes for the pronouncement of one or a few expert catalogers. Referred to as social-tagging, this metadata arrives unrestricted and unstructured; the process democratizes the descriptive task on the predicate that popularity is a predictor of accuracy. Given their lack of editorial influence, these tags can potentially cut across any and all potential categories of description. Genre, key, time signature, lyrical subjects, the appropriate mood for listening, even the internal memes of the tagging community may coexist as relevant labels. [2] shows us that this approach has real advantages over bibliographic search, but disadvantages as well. The social interest across all pieces of music is not evenly distributed, with a small minority of music receiving the lion's share of attention and tags, and the long tail of music that remains being so meagerly described as to make meaningful tags hard to distinguish from meaningless noise. In cases where one is primarily concerned with only the most well-known music this might be sufficient, but the effectiveness of social-tagging for a piece of music wanes as a function of its obscurity. When one is concerned with providing

access, penalizing obscurity can be counterproductive. To consider all pieces of music within a collection uniformly, researchers have considered ways to incentivize a user base to consider tracks they might not have otherwise considered. For instance, [3] discusses how the process of tagging might be gamified to encourage users to tag more evenly and comprehensively. Yet attempting to control the behavior of large crowds is never a simple or reliable process.

A similar hurdle once faced text-based IR. That field has since benefitted from volumes of research focused on discerning the semantic qualities of documents without relying on human description. The potential of this research for internet applications, where personal examination of the vast quantities of documents on the web would be inconceivable, injected a new urgency into the field in the 80s and 90s that ultimately gave us the modern search engine and with it the ubiquity of Google (whose searches-per-year surpassed 1 Trillion in 2011[3]). The web-search renaissance was built on the basic ability to parse and estimate the semantic relevance of textual documents digitally, but such a task is not so easily replicable with musical content. Text is divisible into letters: well-defined elements that can be represented as a standard string of binary data, the collection of which can represent a word. Words exist as independent entities with relationships that we can categorize in dictionaries and thesauri. Without needing to understand the meaning of words, larger semantic concepts like topic can be algorithmically estimated based on relatively straightforward functions like word occurrences across an index. Music must work with a different kind of data that cannot be understood in the same semantic way that text is. This is not to say that automated text-based content analysis is easier, only that music content-based analysis must use unique

methods to achieve similar results. [4] identifies two techniques that persist in popularity in the MIR community. The first is based on the foundational work previously discussed in tagging music, with the distinction that the tagging is not done socially but rather algorithmically based on the feature analysis of the signal. The second focuses on creating relevance judgements using patterns in the time and frequency representation (TFR) of the music itself without the mediation of a semantically meaningful tag. [4] labels these as the "Bag-of-Features" approach and the "Sequence-based" approach. The following section will detail the ways digital music data is used for content description and the strengths and weaknesses of major approaches that have informed the creation of the structural segmentation algorithm. The final section will discuss approaches to evaluating structural segmentation algorithms and how this study differs from those approaches.

Bag-of-Features

The bag-of-features (BoF) approach is so called because it bears some similarity to the bag-of-words conceptualization of a document used in text-based IR. Like the bag-of-words approach does with words, the bag-of-features approach to content analysis separates some kind of identifiable element from within the piece of music and considers it as a solitary unit outside of the particular context in which it appears. Which kind of feature being considered usually gives the particular implementation of this approach its name; researchers call it alternatively by the names bag-of-features, bag-of-frames, bag-of-audio-words, bag-of-systems, and so on to distinguish the particular qualities being bagged in their approach [5]—[10]. Common to all approaches is a set of tags tied to a probabilistic model that identifies some features of the TFRs of the pieces of music in the

collection commonly associated with those tags. A machine-learning process of some variety is typically employed to improve the accuracy of the probabilistic tagging feature [9][10]. Tags are often pre-defined according to some controlled vocabulary, although this approach can be combined with social tagging to develop an unbound dictionary of tags tied probabilistically with identifiable features as in [2]. Accordingly, these automatically generated tags may theoretically take any form: genre, instrumentation, mood, etc. and are therefore well-suited to search systems in which an end-user is searching for music based on a keyword query-by-text. While the variety of approaches that fall within the BoF framework are too numerous to go into in fine detail, one can nonetheless outline the common process of generating a BoF representation as [8] did in fig. 1.

Figure 1. From [8]

The first few steps, from audio signal to feature extraction, are just as previously outlined. Preprocessing refers to any step that must be taken to prepare a digital audio file for signal analysis, including changing the file format or other such tasks. To disambiguate, in fig. 1 the term “segmentation” refers to the process of segmenting an audio signal into frames by a window function. Once a TFR for an audio signal is created, the BoF approach must quantize the vectors of the TFR. In other words, they must identify some number *n* of relevant values per some subdivision in the TFR and map them to a vector of *n* dimensions. This vector is then predicted to belong to some defined feature vis-à-vis a probability model in that vector space, where probability is determined based on an initial sample set of data for which both vectors and feature tags are known.

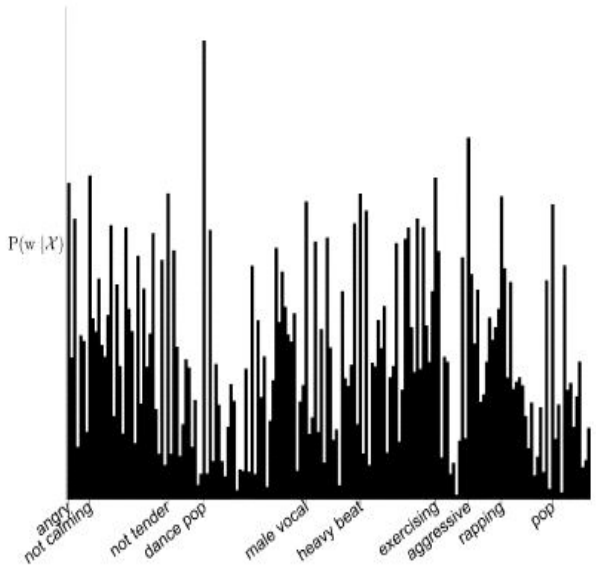

Figure 2. A histogram of tags, w, *in the song* Give it Away *by the Red Hot Chili Peppers and their cumulative probability,* P*, of occurrence in the bag of feature vectors,* X*. The 10 most probable tags are labeled. From [9]*

A common approach is to employ a nearest-neighbor calculation between the given vector and the nearest known sample vector. While difficult to visualize in vector space of more than 3-dimensions, the idea is that one can predict which feature is represented by a new vector simply by determining which feature is represented by the nearest known vector (by Euclidian distance). This is the basic approach used in [8][11]. This approach has drawbacks, notably that one must compute each new vector against all known vectors as they are generated. An alternative proposed in [9] is to define some parametric function in the vector space for which the output is the probability that the vector belongs to a certain feature. Using a parametric probability function requires less computational effort, making it more suited to large collections. These probability functions are most often defined using a Gaussian Mixture Model (GMM), in which multiple Gaussian distributions of probability simulate a continuous probability function[9][10][12], although alternative or modified constructions are not uncommon[6]. The audio signal is thus reduced to a collection (or bag if you will) of these feature vectors for which each is said to belong to a probable feature. That feature for each vector is taken from the vocabulary of tags that is determined by the sample data. The data can then be represented by a histogram of all

tags in the vocabulary and their cumulative probability of occurrence in the song as in fig 2.

While research continues to make progress in improving BoF approaches to music content-analysis, [5] identifies persistent problems that limit its general application at least to polyphonic music (music with many voices or sources of sound that do not always produce sound in unison). One, BoF methods usually cannot improve beyond a maximum precision (about 70%) that is not affected by extenuating factors, which [5] calls the *glass ceiling*. Two, means of modeling dynamic changes in the audio signal in BoF approaches offer no improvement over static models despite their significance in the perception of a listener. Three, intriguingly there seems to exist a class of polyphonic songs which are found to be consistently returned as false positives in BoF MIR tasks regardless of the circumstances of the search; these songs are called *hubs.* Additionally, [4] identifies the more general weakness that, while BoF approaches may succeed in identifying features accurately, they ignore the context in which the features exist and the behavioral relationships between them within the song. For instance, in principle one could use tags to identify the number major and minor chords are present in a song, but not the movement between major in minor chords across a song. Likewise, one could use tags to identify a saxophone in a song, but not where in the song it plays a solo. These descriptions and those like them, while they may not be useful for the lay searcher, are imperative in a musician's conceptualization of a piece of music. For these tasks, we must look beyond the paradigm of traditional metadata description using text established by the practices of card catalogues. One must be able to describe temporal and structural patterns within a song directly. [4] calls these approaches "sequence-based."

Sequence-Based Analysis

When describing temporal and structural patterns, one is not looking merely for the presence of some values, but the relationship of those values to the values around it. This requires knowledge of the order of values; in other words, we must examine not just the values but the *sequence* of values. For example, consider these three sequences of integers: “1,2,3,4,5”, “1,2,3,5,4”, and “2,5,3,4,1”. Approaching this with a BoF framework would allow us to identify the equal occurrence of the same values in each sequence, and therefore equal similarity among all sequences. However, given “1,2,3,4,5” as a query, one nonetheless would likely want to identify ”1,2,3,5,4” as being more similar or relevant than “2,5,3,4,1”. The body of sequence-based approaches to retrieval depends therefore on the ability to quantify the degree of similarity between two comparable sequences of values [4]. This is called *sequence alignment*. In general, sequence alignment seeks to generate an alignment score between sequences, such that the highest scoring sequence can be said to be the most similar to the query sequence. The specific process, however, depends on the class of values that make up the sequence being examined. Certain important features in music can be understood only as a sequence. For instance, a *melody* is a sequence of pitches; a *chord progression* is a sequence of harmonies; even a piece of music itself can be considered a sequence of repeating sections. The accuracy of sequence alignment as a process then depends on how accurately one can identify the values that make up these sequences: pitches, harmonies, sections. Significant progress has been made in the field of melodic transcription of polyphonic audio, but the task has not yet advanced to the degree that it may be consistently applied to recorded music as opposed to MIDI-based audio examples

[4]. The reader is referred to [13] for a review of pitch tracking systems in melodic transcription and to [14] for a comprehensive overview of digital melodic transcription research. Although melodic transcription is not yet applicable in writ large, significant advancements have been made in sequence-based analysis based on the final two examples, chord sequence and structural sequence. This review will only discuss the research into the latter, although a discussion of chord sequence estimation and its foundational work in identifying musical "states" can be found in appendix 5. The following will refer to TFRs known as the Mel Frequency Cepstral Coefficients (MFCC) and the Constant-Q Transform (CQT) in some detail. See appendix 4 for a full definition and discussion of these types of TFRs that are used in musical content analysis and specifically in the spectral clustering algorithm.

Structural Sequence

The analysis of structural sequence is a way to identify and order the states that are emergent within the signal, the musical *form*. While computationally difficult, this is a process that even lay listeners perform almost subliminally when listening to a piece of music. It is the process by which a listener can infer, for instance, that the chorus of a song has moved to the verse. These states within the music, which may colloquially be referred to as a section or part, are conditional on their relationship to other states; that is to say, you cannot logically have a song that is all chorus and you cannot have a bridge without the two sections that it bridges. It is the repetition, or lack thereof, and order of these emergent states which allow us to classify them. Despite how naturally a human listener may be able to identify these sections, the computational equivalent referred to as *segmentation* has proven to be a challenge. [15] proposes that sections within a piece of music are defined by 3 fundamental relationships: homogeneity, novelty, and repetition.

Homogeneity refers to those consistent elements within a section that allow us to say that it is one single unit; novelty is the contrast in elements that marks a break in homogeneity and thus a new section; and repetition is that feature that marks the recurrence of a previously-occurring section. These relationships must be determined by some features that can be represented in a TFR, although which features most clearly establish the relationships may vary. Accordingly, approaches to the structural sequence problem use a variety of TFRs with a variety of specialized uses as a starting point, and there is as yet no one TFR that is clearly best-suited. [16] established in 2001 that the MFCC generally outperformed other TFRs if the focus of segmentation rested on timbre; however, new research and new applications since then have broadened the horizons. While the MFCC continues to be used in many studies, it appears in the corpus alongside, and indeed often in conjunction with, chroma features and to a lesser extent the CQT as well as many less common TFRs [15].

Regardless of the TFR used, a specialized representation is used for segmentation that has not yet been discussed. Rather than visualizing the frequency against time in a signal, segmentation requires some method of measuring homogeneity, novelty, and contrast. For this, we do not need to know the specific values of frequency features, but rather some measurement of the relative similarity and distance of these features to one another. [17] proposed a metric known as the self-similarity matrix (SSM) that allows for this. Given two vectors, which in this case represent the values of two frames of some TFR, [17] asserts that the scalar product of the two vectors may be used as a similarity metric. [15] notes that Euclidean or cosine distances between the vectors are also commonly used. By comparing each frame of the given TFR pair-wise with every other

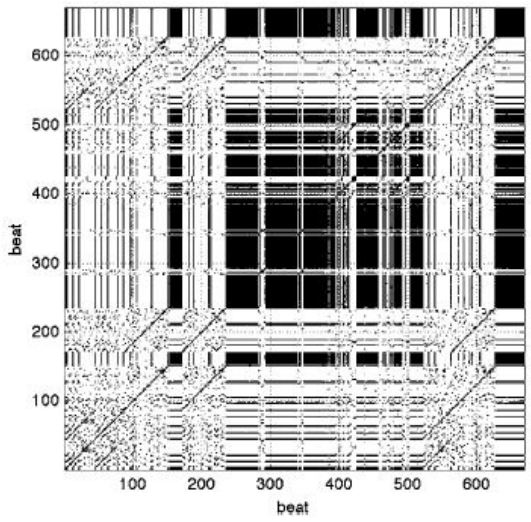

Figure 3. A self-similarity matrix representation for Mozart's Symphony #40, Mvt. 3. Low distance is denoted by darker shades. Frames are divided by beat. From [52]

frame, one can construct a square matrix with the values representing the distance between every combination of vectors. If represented as a heat map, one should find that the values are lowest along a center diagonal of the matrix, representing the distance between each frame and itself. Diagonals parallel to the center diagonal represent low distances between one succession of frames and a separate succession of frames. This can be used to identify repetition. Square regions of lower values along the central diagonal represent a localized section of frames that have low distances among themselves. This can be used to identify homogeneity. Regions of high distance values near the central diagonal represent frames that are near to each other in time but have a large distance metric. This can be used to identify novelty. These features can be seen in fig 3.

With a given self-similarity matrix, it follows that the next undertaking is to describe some computational method of identifying these relevant repetition, homogeneity, and novelty features. Different algorithmic approaches often prioritize one of these three qualities [15]. Additionally, within these three categories of approach, there are two goals to which an algorithm might aspire. The first is boundary detection, in which the aim is to identify the points in time in the signal that delineate where sections begin and end. The second is labeling, which focuses on grouping sections by the

likelihood that they are alike. Segmentation algorithms typically accomplish either boundary identification only or both boundary and label identification.

In early boundary identification experiments, [18] attempted boundary identification without a full SSM representation, simply by calculating the Mahalonobis distance between vectors of successive frames in multiple TFRs; however, this method suffers in that the scope of frame-to-frame novelty does not take into account the context of the frames. That is to say, sometimes, a boundary cannot always be identified as change in a single instant. [19] builds on the original SSM research by introducing a boundary algorithm that prioritizes novelty of a region. In this method a checkerboard-like kernel with a Gaussian radial function (a visualization can be seen in fig. 4) with a given size, or duration, iterates across the central diagonal of the SSM. The correlation between the values in the kernel and the values in the SMM are measured and plotted against the duration of the song producing a *novelty curve*. Where the regions of high and low similarity conform closest to the checkerboard shape of the kernel, the correlation and thus novelty is high. This shows the location of box corners of high and low distance, where regions of high similarity are separated by regions of high distance. High values in the novelty curve suggest that a location is a logical boundary point between sectional regions.

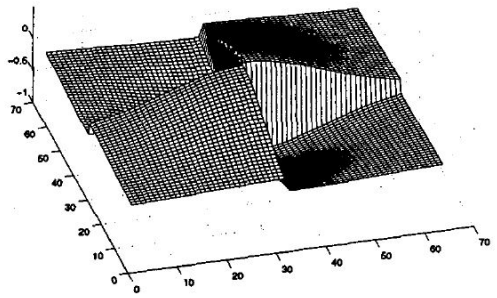

Figure 4. The kernel with Gaussian radial function used to measure audio novelty in a self-similarity matrix in [19] and [17].

[20] proposed an improvement to the standard novelty curve for boundary identification that includes both local novelty and a model of global novelty. In the proposed

method, each vector of a given TFR is concatenated with the values of preceding vectors according to some duration parameter. This produces a series of high-dimensional nested vectors that retain a kind of "memory" of the recent past. A novelty curve against a time-lag transformation of these vectors yields boundaries that more accurately captures transitions between inter-homogenous sections rather than simply the highest points of local novelty. In [21], the authors build upon the work of [20]. They do away with the novelty curve altogether and instead reduce the memory-informed self-similarity matrix to a fixed-dimensional (i.e. duration-independent) matrix of latent repetition factors that capture transitions between repetitive and non-repetitive sections. [22] formulates an alternative method of regional boundary identification from homogeneity rather than novelty. A cost function is utilized that computes the sum of the average self-similarity between successive frames of a signal. The task of the function is to group as many frames together as possible given a cost parameter that penalizes grouping frames with low self-similarity. By increasing the value of the cost parameter, the number of possible segments decreases. This allows control over the function in its implementation that prevents spurious or too-frequent boundary identification.

[18] and [19] both note that boundary identification alone can be useful for example by facilitating audio browsing (where the listener may want to jump between meaningful sections rather than attempting to locate them via traditional fast-forwarding and rewinding). However, simply identifying boundaries does not provide meaningful descriptions of the relationships between sections. Repetition-based methods have approached labeling visually, as a task of identifying the diagonal stripes parallel to the central diagonal in the SSM. These techniques have been prone to noise-based errors and

false stripe identification, and they rely on the assumption that all repetitions occur in the same tempo (changes in tempo result in a distortion in the shape of a stripe, angling it towards or away from the diagonal) [15]. In the first approach to some kind of label identification, [23] sought to combine the novelty-seeking kernel boundary algorithm of [19] with a homogeny-seeking clustering algorithm that compares the Singular Value Decomposition (SVD) of the regions between novelty-identified boundaries. The SVD is computed as a function of the relationship between the empirical mean and covariance of the spectral values in the TFR for each segment. This SVD takes a value between 0 and 1. These SVD values are then used to create a segment-indexed similarity matrix (seen in fig. 5). High SVD values indicate that two segments should be grouped under the same label. This is a highly versatile method that [23] notes could be used to identify structural similarity even in image or video data; however, the process is computationally intensive. Furthermore the similarity between segments determined by SVD is unable to account for changes in key that do not affect the underlying structure. In other words, where a human might recognize a section in one key with a certain melody, and a section in another key with the same melody as belonging to the same section, the SVD cannot. [24] proposes a novel alternative to clustering by SVD using a calculation of the 2D-Fourier Magnitude Coefficients (2D-FMC) of each segment. This has the dual advantage over the SVD in that it is computationally simpler and key-invariant. In this implementation, the 2D-FMC used can

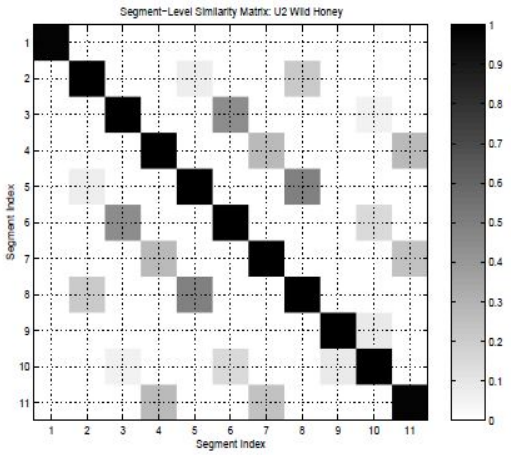

Figure 5. A segment-indexed similarity matrix of the SVD values of a piece of music determined to have 11 segments.

be described as a segment-length-normalized matrix of values that measures in two dimensions the frequency-amplitude of some segment of chromagram the way that a DFT measures in one dimension the frequency-amplitude of a signal. The distance relationship between the 2D-FMCs for each segment can be plotted against each other in a similar segment-indexed similarity matrix.

These methods are largely successful in applying labels but they are nonetheless dependent on an independent boundary identification algorithm. [25] the “constrained clustering” algorithm holds that the identification of recurring sections can more efficiently be done using an E-M trained HMM to identify section states similar to the models used in chord identification (see appendix 5). The process by which this occurs uses what is essentially a continuous, adapted BoF approach that reduces some frame within a TFR to a histogram of feature probabilities. These probabilities are related to some probability model of states for which the states involved correspond to expected structural sections types in the vein of “chorus,” “verse,” “intro,” etc. Unlike [23], this method has the ability to describe sections meaningfully even if there is no repetition of them within a song. Additionally, the process can optionally be refined with the addition of an independent novelty-seeking boundary identification mechanism which can be used to introduce “cannot-link constraints” that define frames which should not be linked with the same label. Unfortunately, the general applicability of the method is limited given that it requires foreknowledge of the type of music to which it is being applied in order to precisely define the states-as-sections.

Spectral Clustering

The spectral clustering algorithm proposed in [1] offers a dual-purpose alternative that is more generally applicable: a method of both a boundary and labeling identification based on the graphical interpretation of a transformation of the SSM proposed based on concepts in spectral graph theory. The algorithm does not generate a novelty curve across the central diagonal and relate the sections that fall within these boundaries. Rather, it seeks to explicitly identify nested or hierarchical sections by analyzing the SSM at narrowing levels of granularity and relates the identified sections via a combination of local timbre features and long-term harmonic features. One way to express the practical implications of this narrowing granularity is that it seeks at each step to separate a given signal that is assumed to be completely homogeneous into two identifiably distinct divisions. The first step divides the most distinct, like sections from the rest of the song; the second divides the most distinct, like sections from what remains; the third from what remains of that; and so on and so on up to some parameter of steps set by the algorithm. This process is visualized in fig. 6, where the parameter, *m*, is set at 10. The process by which it arrives at this solution is explained subsequently.

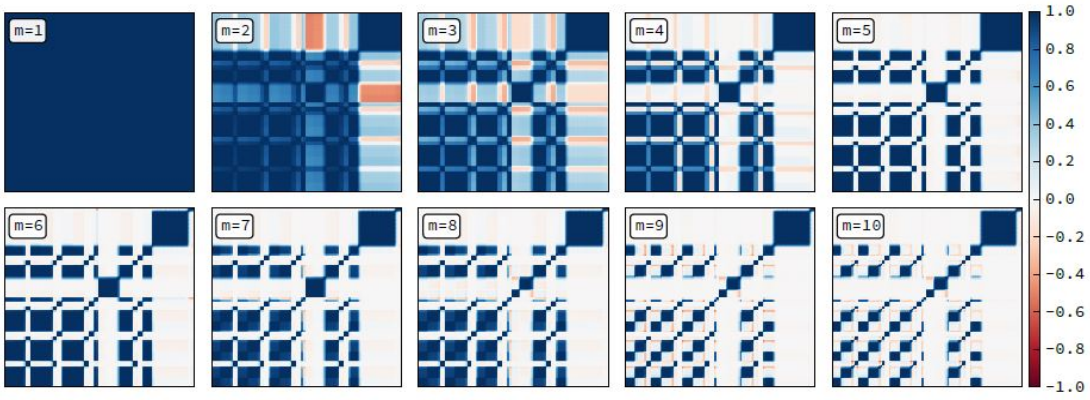

Figure 6. The song Come Together, *by the Beatles is viewed as entirely homogenous at m=1 and progressively divided. At m=2, one sees the 'outro' identified as distinct from the rest of the song in the upper-right corner. At m=3, the 'solo' roughly in the center is revealed. As m increases, the repetitive structures of the verse and chorus become more evident. Finally, as m approaches 10, even individual measures can be identified.*

The procedure can be divided into 2 parts: the construction of a graph suitable for analysis and the analysis of the graph. First, a CQT of a signal is generated. This TFR is chosen based on its ability to capture long-term harmonic patterns. The CQT is mean-aggregated to the beat level; that is to say, the frames of the CQT that fall within a single beat are collected together as a single frame for which the spectral vector values are a mean of the vector values of the frames that fall within that beat. A memory-informed CQT is constructed according to the precepts of [20] from the beat-synchronous frames where each frame is concatenated with the frame that immediately precedes it. A specialized SSM is constructed from the resultant modified CQT according to a nearest-neighbor calculation. In this form, the values between each pair of frames is not a linear distance metric. Instead it is binary: 1 for two frames that are determined to be nearest neighbors in the vector space and 0 for all other frames. This produces an SSM that enforces representation of only the strictest similarity; however, the representation produces a field of points rather than smooth lines. The representation must then be filtered to more clearly show patterns of similarity. This is done with the aid of an MFCC representation of the signal, chosen because it more accurately represents local patterns in timbre. A beat-synchronous MFCC is constructed from the first 13 mel frequency cepstral coefficients. The relationship between the CQT and the MFCC representation are used to generate a filtered representation of the SSM intended to capture the *affinity* between local and global sequences of similarity called the *affinity matrix*. The full calculation of this affinity matrix is outlined in the proposal of the method in [1].

Once this affinity matrix is generated, the concept of the *Laplacian* from spectral graph theory informs the subsequent analysis. The Laplacian is a differential operator in

the graph in that it measures the diffusion of points in the affinity matrix. Clustering occurs based on this diffusion of points, where regions of less-diffuse points can be said to constitute a region of homogeneity in the graph. This is done progressively according to the *m* smallest eigenvectors of the Laplacian. These eigenvectors measure the rates of diffusion such that they progressively increase in complexity. The first eigenvector encodes membership in the complete set. The second encodes the clearest differential in diffusion. The third encodes the clearest differential in diffusion of the result, and so on up to the eigenvector *m* which is a parameter. The higher the value of *m*, the more the diffusions will be differentiated and thus the more segments will be identified; however, higher *m* attempts to measure granularity in the affinity matrix so fine that it becomes sensitive to errors in the representation. There is also the simpler problem of over-segmentation. As [1] notes, the challenge of the spectral clustering algorithm is in defining the parameter *m* without "a priori knowledge of the evaluation criteria," or in other words, without some information about the required level of granularity. One possible application of this study is to determine the effect of certain known qualities of diversity in a music collection that may be known a priori, whether by expert knowledge or estimation by some independent automated system, on the performance of the spectral clustering algorithm. This will reveal which of these qualities are correlated with worse segmentation accuracy, whether by over- or under- segmentation, and thus suggest which qualities may require adjustment of the algorithm.

Evaluation

Before moving on to the evaluation of spectral clustering performed in this study, it will be useful to describe the various evaluation metrics commonly used in the segmentation

task and exactly what they are supposed to evaluate. Segmentation evaluation falls under the purview of the Music Retrieval Evaluation eXchange (MIREX), a community of MIR researchers that organize a yearly presentation of evaluation results of state-of-the-art MIR algorithms. MIREX, which began officially in 2005, has since been the primary conduit through which MIR evaluations are conducted [26]. In order to provide robust evaluations that can be said to be generalizable across multiple tools and algorithms, MIREX seeks to standardize 3 components of the evaluation process: 1, standardized tasks or queries to be made of collections; 2, standardized evaluation metrics that measure success at these tasks; and 3, test collections of significant size to allow for these tasks and evaluations to be run [27].

The simplest evaluations concern only the boundary identification task and are focused on measuring the difference between the boundaries estimated by an algorithm and known boundaries. There are two accepted ways of doing this which may be used together. The first is *hit rate*, which considers the accuracy of estimated boundaries by detecting whether or not they fall within some window of time surrounding a known boundary. Common windows are 0.5 seconds for strict accuracy, explained in [28], and 3 seconds for more lenient accuracy, explained in [25]. There are 3 values associated with hit rate corresponding to precision, recall, and the F-measure of the two. Precision measures the percentage of estimated boundaries that fall within a known boundary's window; recall measures the percentage of known boundary windows that include an estimated boundary; and the F-measure the harmonic average of the two rates. The second boundary evaluation is known as *median deviation*. [28] defines this metric as well. Deviation refers to the value in seconds that separates an estimated and known

boundary. The median is determined based on the total collection of deviations between near boundaries. There are two values associated with median deviation: that between known boundaries and their nearest estimated boundaries and that between estimated boundaries and their nearest known boundaries. These are respectively referred to as the Median Deviation E to R and Median Deviation R to E. Both the hit rate and the median deviation may be *trimmed*. To trim the metric means to ignore the values generated by the first and last boundaries. This can be useful when one does not particularly care about an algorithm accurately labeling the point at which silence ends and the music begins (and vice versa). When these values are less than a second from the beginning and end of the track, that accuracy is not particularly informative for meaningful segmentation.

There are also two metrics associated with labeling identification accuracy. The first is known as *pair-wise frame clustering*, defined in [25]. This value compares labeled frames in an estimation against known labels. All frame pairs are considered against each other. The pairs that are assigned to the same label in the estimation form the set P_E, and the pairs that are assigned the same ground-truth labels form the set P_A. There are three values that make up this metric corresponding again to precision, recall, and F-measure. These can be calculated according to these equations (ex. 12[25]) where PWF_P measures possible under-segmentation and PWF_R measures possible over-segmentation. Labelling success can also be evaluated with the metric known as *normalized conditional entropies*, described by [29]. This is a rather more complex metric that measures the amount of missing and spurious information in a labeling estimation. The conditional entropy measures the number of disagreements between estimated frame labels and known frame labels; however, this is value is simply a count. In other words, even between two

estimations that each have full disagreement with their known labels, the song with the most segments receives a worse conditional entropy score simply because it has more to disagree about. The normalized conditional entropy is, aptly, a normalization of this count that makes it segment-count-agnostic. The metric is calculated as a rate, that is to say as a value between 0 and 1, and flipped so that better performance returns a higher rate. There are also 3 values associated with the normalized conditional entropy precision, recall, and F-measure, defined as (ex. 2[29]) where S_O measures over-segmentation S_U measures under-segmentation. $\mathrm{H(E|A)}$ refers to the conditional entropy of estimation to ground-truth (spurious information), $\mathrm{H(A|E)}$ refers to the same between ground-truth and estimation (missing information), and $\mathrm{N_e}$ and $\mathrm{N_a}$ define the size of the estimation and ground truth respectively. The full process of arriving at these entropy scores is outlined in [29].

Ex. 1

$$\mathrm{PWF_P} = \frac{|\mathrm{P_E} \cap \mathrm{P_A}|}{|\mathrm{P_A}|}, \qquad \mathrm{PWF_R} = \frac{|\mathrm{P_E} \cap \mathrm{P_A}|}{|\mathrm{P_E}|}, \qquad \mathrm{PWF_F} = \frac{2\mathrm{PWF_P}\mathrm{PWF_R}}{\mathrm{PWF_P} + \mathrm{PWF_R}}$$

Ex. 2

$$\mathrm{S_O} = 1 - \frac{\mathrm{H(E|A)}}{\log_2 \mathrm{N_e}}, \qquad \mathrm{S_u} = 1 - \frac{\mathrm{H(A|E)}}{\log_2 \mathrm{N_a}}$$

Structural segmentation is just one MIR task among many that are evaluated yearly at the MIREX, and in fact it is one of the newer tasks, added first in 2009[4]. Nonetheless, a number of collections have since been generated that allow for segmentation evaluations. The most recent MIREX event included 4 datasets of songs used for segmentation evaluation[5]: the original dataset collected for MIREX 2009, two

datasets collected for MIREX 2010, and a fourth dataset put together by the Structural Analysis of Large Amounts of Music Information (SALAMI) research team. The primary function of these datasets is to link commercially or freely available songs with what is known as ground-truth boundaries and labels (boundaries and labels determined by an expert listener) referred to as *annotations*. The annotations take three forms, the simplest two of which are boundary information alone and boundary information between simple, non-overlapping sections each with a single label like "intro," "outro," "chorus," etc.[30]. The former method of annotation is adopted by one of the MIREX '10 datasets[6] , the latter is used by the remaining MIREX '10 dataset[7] and the MIREX '09 dataset.[8] The SALAMI dataset differs from these in that it uses a unique annotation method that follows that proposed in [31] that allows for hierarchical sections (large-scale and small-scale) and accounts for possible similarities between different sections. For instance, while a "solo" and an "outro" may constitute separate musical sections, it is possible for them to share musical qualities that are similar. The method used in the SALAMI dataset allows for a broad characterization of sections, which could capture the musical functions of solo/outro, as well as narrower sections within these that can illustrate the musical similarities between them [30]. The SALAMI procedure modifies the original procedure of [31] by more strictly defining the hierarchical segmentations along three tracks: the musical function track ("outro," "chorus"), the musical similarity track, and a third track which defines the lead instrument at a given point [30]. In all four datasets, annotations are generated manually by musically-trained experts.

As mentioned, the primary function of these datasets it to link these annotations to commercially or freely available songs. The datasets used by MIREX for segmentation

are built from previously existing collections of music, which carries a number of advantages. These collections are often used for multiple MIR tasks, and so they often include a wide variety of potentially useful data. Collections may also be assembled with specific conditions in mind like genre breadth or specificity, or ease of access. For example, the MIREX ’10 datasets are constructed using songs from the Real World Computing (RWC) database, first proposed in [32] for the purpose of facilitating MIR evaluation across a variety of genres with publically accessible music. It includes songs from three broad genres (Classical, Jazz, and Popular) as well as a fourth component of entirely royalty-free music, all of which were performed and recorded for the purpose of inclusion in the RWC database 32 [32]. Songs are provided with corresponding MIDI files and full text of any lyrics used [32]. This source database prioritizes the accessibility of the included music and provides useful metadata, but because the included songs exist only for use in the database and are performed by a limited set of performers, it can only approximate the kind of variety in production and performance that a real-world collection of music might represent. The SALAMI dataset draws from multiple databases, including the RWC database, with a priority “to provide structural analysis for as wide a variety of music as possible, to match the diversity of music to be analyzed by the algorithms.”[30]. The largest component database, and the one used in the following study, is Codaich, chosen by SALAMI for its detailed curation of metadata [30]. Songs described in the Codaich database represent pre-existing commercial pieces contributed from three sources: the Marvin Duchow Music Library, the in-house database used by Douglas Eck of the Université de Montréal, and the personal music collections of the McGill Music Technology Area [33]. Metadata for Codaich, including the more than 50

subgenre tags, was first drawn from the Gracenote CD database[9] and then edited for clarity and consistency by the compilers of the database at McGill University [33]. The combination of robust metadata and variety of content makes the Codaich portion of the SALAMI dataset idea for testing the correlations between these metadata and the accuracy of a segmentation algorithm, although this is not the traditional way the segmentation is evaluated through MIREX.

Because databases like Codaich, which contain robust metadata on a variety of real-world songs, necessarily use commercially available songs in their collection, the database cannot be shared freely among researchers due to intellectual property concerns. As [27] says, "The constant stream of news stories about the Recording Industry Association of America (RIAA) bringing lawsuits against those accused of sharing music on peer-to-peer networks has had a profoundly chilling effect on MIR research and data sharing." Instead of having the songs in these databases be shared, MIREX has adopted an evaluation model wherein the datasets are held by one entity, MIREX itself, and multiple researchers each submit their algorithm to MIREX to be evaluated. This simplifies the matters of copyright by eliminating the need to share commercial music, but results in a particular model of evaluation. Namely, with multiple algorithms being evaluated against common collections simultaneously, the evaluations take on an adversarial nature. Multiple algorithms are run against the same collections and the results and results indicate their relative performance compared to each other.[10] This is helpful in determining the state of the art among participants, but time and resource constraints prevent MIREX from examining results in finer detail [26]. Although datasets like SALAMI provide detailed metadata for each track, results returned by MIREX are

flat; that is to say, the report evaluation results for each algorithm for each track but do not analyze possible variations in that data using the given metadata that accompanies the dataset. The following study will present one method of taking the evaluation metrics used in MIREX segmentation task evaluations and examining them in finer detail using the detailed metadata provided in the SALAMI dataset. By including metadata such as genre, class, tempo, duration, and time signature in the analysis of the evaluation, one is able to determine the specific relationship between these variables and the accuracy of the algorithm. For instance, one would be able to say not just that the algorithm is generally expected to accurately describe song structure, but to say that the algorithm is expected to describe song structure in one genre more accurately than another, or that the algorithm is expected to increase the accuracy of its description for songs in faster tempos. This is done by taking the evaluation scores across a full collection, similar to the results offered currently by MIREX, and analyzing the means and variances in these scores according to the given metadata. This study will compare the means and variances in evaluation scores between each genre and class and the evaluation scores and determine whether there are significant differences. Additionally, it will find whether significant correlations exist between tempo, duration, and time signature and the evaluation scores and determine the strength of those correlations. We will then use the information described previously about the spectral clustering algorithm to offer possible explanations for these differences and describe how they might affect the practical usage of the spectral clustering algorithm for automated description.

Notes

[1] naxosmusiclibrary.com
[2] musicbrainz.org
[3] According to http://www.internetlivestats.com/google-search-statistics/#trend. Just imagine if similar research could do something similar for music. Even something orders of magnitude less influential would be an unimaginable change in how we consume music.
[4] http://www.music-ir.org/mirex/wiki/2009:Structural_Segmentation
[5] http://www.music-ir.org/mirex/wiki/2015:MIREX2015_Results
[6] http://nema.lis.illinois.edu/nema_out/mirex2015/results/struct/mrx10_1/
[7] http://nema.lis.illinois.edu/nema_out/mirex2015/results/struct/mrx10_2/
[8] http://nema.lis.illinois.edu/nema_out/mirex2015/results/struct/mrx09/
[9] http://www.gracenote.com/
[10] A good example of MIREX results can be found at http://nema.lis.illinois.edu/nema_out/mirex2015/results/struct/salami/summary.html

III. Methodology

Overview

The broadly stated objective of this study is to determine how certain variables within a diverse collection of songs may affect the accuracy of the spectral clustering algorithm in segmentation tasks for that collection. In a general context, diversity in a collection of music can mean that the collection contains a breadth of songs from multiple genres, of varying durations, a wide range of years of release, multiple unique instrumentations, many keys, tempos, styles of (or absence of) vocalists, etc. Any sufficiently defined variable could theoretically be measured against the algorithm's performance, but given the strengths of the sources of data for this study described below, the specific variables examined here will be class (broad genre category), genre (narrower genre category), song duration, tempo and time signature. The accuracy of the spectral clustering algorithm for segmentation tasks will be evaluated according to the metrics of median deviation (trimmed) of the boundaries, hit rate (trimmed) of the boundaries with a 3 second window, pair-wise frame clustering, and normalized conditional entropy. These metrics will be analyzed in terms of the variables to show the extent to which those variables have an effect on the evaluation results. This information can be used in a number of ways. From the perspective of someone considering implementing the spectral clustering algorithm in describing the structure of songs in a particular collection, the analysis provided in this study will provide baseline expectations based on the

collection's known characteristics. For instance, someone with a collection that focuses on a single genre or genres within a single class would be able to predict the likely performance of the algorithm specifically in relation to that genre or class. On the other hand, someone with a collection that holds songs of varying durations could identify more easily which songs could be described effectively by the algorithm and which might require manual description. Being able to view multiple analyses like the one presented here that cover different description algorithms would allow those charged with picking among them to make a more informed decision based on their collection. Finally, in the case of the spectral clustering algorithm that operates with adjustable parameters, an analysis like the following can suggest possible conditions that warrant adjusting those parameters for more accurate segmentation estimations.

In the design of the experiment there are 4 primary components:

- First, a collection of song files is needed for which the structure of the collected songs in known. The collection must be large enough to represent a breadth of values among the variables to be examined in the collection. Furthermore, the structure of the collected songs must be determined with a reasonable level of expertise, preferably by hand by a subject matter expert, independent of the estimations provided by the structural segmentation algorithm.
- Second, a script must be utilized that is capable of creating these structure estimations for the given collection of digital audio files using the spectral clustering segmentation algorithm designed by McFee & Ellis [1]. The estimations created by this script must be in a machine-readable format.

- Third, another script must be utilized that is capable of referencing the estimations of the segmentation algorithm against the independent, ground-truth song structure for each audio file. The output of this script should be a set of numerical values corresponding to standardized evaluation metrics for structural segmentation tasks.
- Fourth, a statistical analysis will be performed on the data determining the extent to which the known variables in the given collection affect the values of the resultant evaluation metrics. Results will demonstrate which qualities are correlated with less effective (lower evaluation scores) or more effective (higher evaluation scores) performance of the segmentation algorithm.

Tools and resources

In order to realize this task, I am entirely reliant on the generous contributions of MIR researchers who have in recent years made vast quantities of both their own data and open-source software tools available online. Here I will provide a brief description of the various tools used and their value to the outline above.

The particular software tools for segmentation analysis and evaluation are taken from the Music Structure Analysis Framework (MSAF)[34], an open-source framework written in the Python programming language by Oriol Nieto and Juan Pablo Bello and first presented at the ISMIR 2015 conference. This software package was selected for its versatility and the extent of evaluation options included. MSAF defines functions in Python for five boundary algorithms and three labeling algorithms, including McFee and Ellis' spectral clustering algorithm. MSAF is dependent on *librosa* [35] for audio feature analysis and *mir_eval* [36] to compute evaluations. Statistical analysis of the evaluation results is done in JMP.

Structural annotations are sourced from the SALAMI annotation data, a project of the Digital Distributed Music Archives and Libraries lab (DDMAL) at McGill University in Montreal [30]. This dataset provides metadata and ground-truth structural annotations for more than 1400 songs from a wide variety of sources. The specific metadata provided varies based on the source database of the music. While SALAMI has annotations for songs from the Real World Computing (RWC) Music database, the Isophonics music database, the Internet Archive music database, and the Codaich database, only music from the Codaich database was selected for this study due to its more robust genre classifications.[11] Further metadata is provided by SALAMI in partnership with the Echo Nest[12] including duration and estimations of tempo and time signature subdivision.

Given that the songs in the Codaich database are all held under standard commercial copyright, the individual audio files had to be purchased through conventional means. Because SALAMI provides bibliographic data about the songs for which it created annotations in the XML format used by the iTunes library, the iTunes online store was selected as the means of purchase. Within the Codaich subsection of the SALAMI annotations, there are four broad genre classifications represented – popular, jazz, classical, and world – with 52 subgenres between them at a total of 835 pieces of music. While it would have been ideal to have all four genre classifications represented in this study, the collection was limited only to songs classified as *popular* or *jazz*. There were two reasons for this choice. First was a limitation of naming conventions in classical music – the construction of the iTunes library file provided limited metadata that was insufficient to ensure that any particular *classical* track that was purchased was the correct track as referenced in the SALAMI annotations. This is due peculiarities in the

classical tradition in which many different pieces by different composers may share the same title (e.g. “Sonatina”), and also the lack of rigorous naming conventions by commercial music services in which a piece may be known by multiple titles or the “artist” for a piece of classical music may be listed as alternately the composer, the performance ensemble, or the individual performers involved.[13] Second was a limitation of the Codaich database in regards to iTunes – many pieces of music classified as *world* music by Codaich appear on compilation CDs donated from researchers’ personal collections that may once have been available in a physical format, but are not available for purchase digitally through iTunes.[14]

Reducing the proposed collection to the two remaining classifications, *popular* and *jazz*, left the total number of songs available at 415 and the number of remaining subgenres at 33. Furthermore, the total size of the collection for this study was limited by funding. Funds for the purchase of music was provided by SILS up to the total of $200 through a Carnegie grant program. At the iTunes-standard cost of $0.99 to $1.29 per track, the size of the proposed collection was roughly estimated at about 165 pieces of music. This number allowed for an even representation of each remaining subgenre at 5 songs each. After these limiting factors, the remaining songs in the proposed collection were cross-referenced against the iTunes store to determine what was available for purchase. Tracks that seemed to be available but could not be confirmed as a direct match with the given metadata were passed over. Other tracks which could only be purchased as part of a full album (increasing their cost) were only purchased if they added to the representation of under-represented variables whether in genre or time signature. After these mitigating factors, the final collection totaled 143 pieces of music, with each genre

represented usually by 4 to 5 tracks. A full table of the songs used, including their title, artist, and the variables used in the subsequent study, can be found in appendix 1.

Procedure

These tracks were migrated to the Linux OS environment (Ubuntu 15.10) in which MSAF was set up to operate. Because iTunes stores purchased music in the M4A file format while MSAF requires either MP3, WAV, or AIFF, a small script using the FFMPEG command-line tool was written to convert all files in the collection to MP3 at a bit-rate of 192K. Due to the requirements of MSAF, each track was named according to its SALAMI track identification number. A second script was written in Python 2.7 to estimate structural segmentation for each track in the collection; this script is essentially only a wrapper for the spectral clustering algorithmic function defined in MSAF. Likewise, this script evaluates the results of these estimations against the ground-truth annotations and stores the scores for each evaluation metric previously outlined for each song in a CSV file that was imported as a data table into JMP. Metadata elements sourced from SALAMI and the Echo Nest representing the independent variables that can be found in appendix 1 were appended to this data.

For all results, a significant effect is assumed at a confidence of 95% or $p< 0.05$. Results against the nominal data of genre, class, and time signature are analyzed according to a one-way analysis of variance. The evaluation score results are assumed to fall along a normal distribution within each category for each variable. Between the two categories of class, *jazz* and *popular*, a two-tailed t-test is performed against the null hypothesis that both classes yield the same response in each evaluation metric to determine the statistical significance of any variation between the two categories. This

test is able to show that variance in results is unlikely to be random, but it cannot demonstrate that the independent variable of class is necessarily *causing* the variation. Between the multiple categories of genre, an *F*-test is employed to test the null hypothesis that there is no significant variance in the evaluation metric scores across genres, and a t-test between each pair of genres is used to determine possible significant differences within the collection. The F-test is likewise able to show the likelihood that variance among the entire collection due to genre is non-random; however, the F-test does not make any claim about specific genres within the collection in comparison to others. For this, the t-test among paired genres is used to demonstrate possible significant differences between them; however, these tests work with much smaller sets of data (two genres together are often comprised of only 7 to 10 songs). This limits their ability to comment generally on how accuracy may be correlated with specific genres, but still gives an idea of what variances might be affecting the results of the F-test. Results against the continuous data of song duration and tempo are analyzed according to their Pearson product-moment correlation coefficient with evaluation results. This measures the strength of the linear correlation between each pair of variables; however, it can be said that even a weak correlation between two variables may still be statistically significant. Again, a standard t-test is used to determine the significance of the Pearson correlation. Like with the previous tests, these results are not able to determine a causal relationship between variables. They can only show that there is a significant correlation between the data.

Research Questions

The study aims to provide evidence that answers these questions:

3) **Do either the narrow *genre* or broad *class* of a song have a significant effect on the accuracy of the spectral clustering algorithm for the structural segmentation of that song?**
    a. **Among all genres, is there a significant difference in the measurement of boundary hit rate, median deviation of the boundaries, pair-wise frame clustering, and normalized conditional entropy as determined by an F-test? And if so, are there significant differences between genres determined by a two-tailed t-test?**
    b. **Between the two classes, is there a significant difference in the measurement of boundary hit rate, median deviation of the boundaries, pair-wise frame clustering, and normalized conditional entropy as determined by a two-tailed t-test?**
4) **Is there a significant correlation between the *tempo*, *duration*, or *time signature* of a song and the accuracy of the spectral clustering algorithm for the structural segmentation of that song?**
    a. **Is there a significant correlation between the tempo of a song and the measurement of boundary hit rate, median deviation of the boundaries, pair-wise frame clustering, and normalized conditional entropy determined by the Pearson product-moment correlation?**
    b. **Is there a significant correlation between the duration of a song and the measurement of boundary hit rate, median deviation of the boundaries, pair-wise frame clustering, and normalized conditional entropy determined by the Pearson product-moment correlation?**

c. **Is there a significant correlation between the time signature of a song and the measurement of boundary hit rate, median deviation of the boundaries, pair-wise frame clustering, and normalized conditional entropy determined by the Pearson product-moment correlation?**

Notes

[11] Further information on the Codaich database can be found at http://jmir.sourceforge.net/index_Codaich.html.
[12] http://the.echonest.com/
[13] The perennial example of this is Beethoven's *Piano Sonata No. 14 in C-sharp Minor, Op. 27, No.2*, Mvt. 1, *Adagio sostenuto*, known colloquially as the *Moonlight Sonata*, performed by countless artists and ensembles under one or both names and appearing on countless compilation albums.
[14] See the Evaluation section in the review of the literature for more detail on the construction of the Codaich database.

IV. Results

Summary

This study found several significant correlations between the variables of genre, class, tempo, and song duration (none among time signature) and performance of the spectral clustering algorithm in the evaluation metrics of normalized conditional entropy (*S*), pair-wise frame clustering (*PWF*), trimmed hit-rate at 3 seconds (*HRt3s*), and trimmed median deviation (*MDt*) of the boundaries. Statistically significant findings can be summarized as the following:

an effect of genre on S_F;

an effect of class on $HRt3s_F$;

an effect of class on *MDt* from estimations to ground-truth (E to R);

a positive correlation between tempo and S_F;

a positive correlation between song duration and *MDt R to E*;

a negative correlation between song duration and $HRt3s_F$ and $HRt3s_R$, but a positive correlation between song duration and $HRt3s_P$.

a positive correlation between song duration and PWF_F and PWF_R, but a negative correlation between song duration and PWF_P;

a positive correlation between song duration and S_O, but a negative correlation between song duration and S_U.

A full account of the significant results is seen below. The complete table of evaluation metric results by song can be found in appendix 2. All following values are rounded to two decimal places except where necessary to report very small *p* values.

Genre

Genre and S_F	
p	0.04
R^2(adj.)	0.11
Mean S_F (collection)	0.56

Table 1

The genre of a song was found to have some effect on the S_F value for that song. The probability *p* that genre explained no difference in S_F was found to be 0.04. The expected effect on S_F is estimated by the adjusted R^2 was 0.11. The mean value of S_F divided by genre can be seen in table 8. A plot of the same, ordered by ascending S_F , can be seen in fig. 7 following the Discussion section. The mean value of S_F across all genres was 0.56. While no mean S_F in a single genre was significantly higher or lower than the global mean of the collection, individual genres differed paired against other genres. Notably, the S_F for songs in the "R&B - Funk" genre was significantly higher than those in the "Instrumental Pop," "R&B – Soul," "Jazz – Cool Jazz," "R&B – Contemporary R&B," "Blues – Urban Blues," "Blues- Country Blues," and "Modern Folk – Singer/Songwriter" genres. Because S_F represents a mean of over- and under-segmentation as well as label agreement, this means that the label placement and grouping for the genre "R&B – Funk" were significantly more accurate than those for the other listed genres. Additionally, the genre of "Modern Folk – Singer/Songwriter" returned significantly worse results than the 19 genres with the highest mean S_F values. A connecting letters report of S_F by genre can be seen in table 9, where genres that do not share a common letter had statistically significant results in a pair-wise comparison.

Class

Class and *HRt3s$_F$*	
p	<0.02
R^2(adj.)	0.03
Mean *HRt3s$_F$* (collection)	0.39
Mean *HRt3s$_F$* (pop)	0.42
Mean *HRt3s$_F$* (jazz)	0.36

Table 2

Class was found to have a probable effect on the placement of boundaries according to the metrics of *HRt3s$_F$* and *MDt E to R*. The algorithm placed boundaries near the ground-truth boundary points more often for *popular* songs, and the algorithm's boundaries were closer on average among *popular* songs than they were for *jazz* songs. The probability that class was not correlated with *HRt3s$_F$* or *MDt E to R* was found to be less than 0.02 and 0.0001 respectively according to a two-tailed t-test. The mean *HRt3s$_F$* for the entire collection was measured at 0.39. The mean *HRt3s$_F$* among the class *jazz* was measured at 0.36 while among the class *popular* it was measured at 0.42. The adjusted R^2 was calculated as 0.03. These results can be seen in table 10 and the fig. 8. The mean *MDt E to R* for the entire collection was measured at 5.69 seconds. For *jazz*, *MDt E to R* was measured at 6.84 seconds while *MDt E to R* for *popular* was 4.22 seconds. Adjusted R^2 for *MDt E to R* and class was calculated as 0.09. These results can be seen in table 11 and fig. 9.

Tempo

Tempo and *S$_F$*	
p	0.01
Pearson's *r*	0.25

Table 3

The tempo of a song was found to have a likely effect on its *S$_F$* score. Tempo was associated with a Pearson product-moment correlation value of 0.25, with a probability of no correlation found to be 0.01. This indicates that songs with a higher tempo were more likely to score better on the *S$_F$* evaluation. A scatter plot of the results of *S$_F$* by tempo can be seen at fig. 10.

Song Duration

Song Duration and *HRt3s$_F$*	
p	<0.01
Pearson's *r*	-0.22
Song Duration and *HRt3s$_P$*	
p	<0.01
Pearson's *r*	0.22
Song Duration and *HRt3s$_R$*	
p	<0.0001
Pearson's *r*	-0.47

Table 4

Song Duration and *MDt R to E*	
p	<0.0001
Pearson's *r*	0.78
Song Duration and *MDt R to E (- outliers)*	
p	<0.0001
Pearson's *r*	0.35

Table 5

Song duration was found to have many significant correlations. Song duration had a significant effect on boundary placement measured by *MDt R to E* and *HRt3s*. The probability of the null hypothesis for song duration and *MDt R to E* was found to be less than 0.0001. The Pearson correlation values between duration and *MDt R to E* was 0.78. A scatter plot of these results can be seen in fig. 11, which seems to reveal a possible outlier effect. With the 6 labeled outliers excluded, the value of *p* remains less than 0.0001; however, the Pearson correlation is reduced to 0.35. These modified results are seen in fig. 12. For duration and $HRt3s_F$, *p* was less than 0.01 and the Pearson correlation value was found to be -0.22. This is interesting as duration was also significantly correlated with the two values that make up $HRt3s_F$. Against $HRt3s_P$ and $HRt3s_R$, *p* was less than 0.01 and less than 0.0001 respectively. The correlation between duration and $HRt3s_P$ had a value of 0.22, while the correlation between it and $HRt3s_R$ has a value of -0.47. This means that longer duration was positively correlated with the hit rate precision, negatively correlated with the hit rate recall, and overall negatively correlated with their harmonic mean. These results can be seen in figs. 13 and 14. Song duration also seemed to have an effect on labeling. Duration was found to be correlated with PWF_F, PWF_R, and PWF_P at *p* of less than 0.01,

Song Duration and PWF_F	
p	<0.01
Pearson's *r*	0.22
Song Duration and PWF_P	
p	<0.0001
Pearson's *r*	-0.17
Song Duration and PWF_R	
p	<0.05
Pearson's *r*	0.44

Table 6

Song Duration and S_O	
p	<0.0001
Pearson's *r*	0.49
Song Duration and S_U	
p	<0.004
Pearson's *r*	-0.24

Table 7

less than 0.0001, and less than 0.05 respectively. Correlation between duration and PWF_F had a Pearson correlation value of 0.22, while the Pearson correlation between duration and PWF_R was measured at 0.44. Between duration and PWF_P, the Pearson correlation was found only to be -0.17. This suggests that longer duration was correlated with higher values of both PWF_F and PWF_R, but lower values of PWF_P. Figs. 15 and 16 show these results. Against S_U and S_O, *p* was respectively found to be less than 0.004 and less than 0.0001. Correlations differed in direction, however, with a Pearson correlation of -0.24 against S_u and 0.49 against S_O. These results are seen in fig. 17.

Discussion

The first thing one must mention in the interpretation of these results is that the sample size of this study, limited as it was by constraints on the number of songs that could be purchased, is smaller than the typical evaluation dataset. Because of this, patterns that have been identified in the dataset are interesting suggestions for what a more comprehensive analysis may or may not confirm. Likewise, the lack of significant correlations does not suggest that such correlations could not be present in a more comprehensive dataset. For example, this study did not reveal any significant correlation between the time signature, measured as the division of beats within a

bar, and any of the evaluation metrics. This could be in part due to the fact that the songs used fell into the class categories of *popular* and *jazz* and did not have equal representation between 3 and 4 divisions. That being said, there were a number of significant correlations that were identified. The following is a discussion of these effects and also their possible causes and ramifications that may be explored by further study.

With regards to the research questions, this study offers the following answers. Both *genre* and *class* have at least some effect on the accuracy of the spectral clustering algorithm. Genre seems to have a small but significant effect on the normalized conditional entropy measure S_F. This generally measures the agreement between the estimated segments and their labels compared to the ground-truth. In other words, genre seemed to have some effect on the ability of the algorithm to accurately identify which sections within each song were alike. Genre otherwise had no significant effect, including notably on the boundary identification evaluation metrics. Class was the opposite; it had no significant effect on *PWF* or *S*, the labeling evaluations, but did affect two of the boundary identification evaluations, the mean hit rate and the median deviation of boundaries in the estimation to the ground truth. For both, popular music fared better than jazz music in the evaluations, suggesting that spectral clustering is better at delineating sections in popular music. *Tempo* and *duration* both had at least some effect on the accuracy of the algorithm, although *time signature* seemed to have no significant effect at all. Tempo had a positive correlation with S_F, indicating the algorithms was better able to match similar sections with each other for songs at quicker tempos. Song duration had many significant effects, which will be discussed in depth below.

The correlative strength of song duration with so many of the evaluation metrics has ramifications for the application of the spectral clustering algorithm proposed by [1]. Song duration is often one of the most identifiable pieces of content-dependent information of a piece of music as it does not require any kind of sophisticated analysis to determine. Rather, duration is a value identifiable even in the most rudimentary systems. The implications of an easily identifiable variable on the evaluative outcome could signal that the value of *m* in the spectral clustering algorithm could be more precisely adjusted to a given song even before more complex values like tempo are estimated. How it might be adjusted is a more complicated question owing to the different kinds of measurements given by these evaluation metrics. For instance, duration was found to have a relatively strong correlation with higher values in the *MDt R to E* metric, indicating that at longer durations the time difference between the boundaries in the ground-truth and the nearest boundary in the estimation was likely to be higher and at times, much higher. There was no similar significant correlation in the corresponding metric of *MDt E to R*, suggesting that duration was not likely to be related to the time difference between boundaries in the estimation and the nearest boundary in the ground-truth. From this, we might infer that at longer durations, the boundaries placed by the algorithm were equally likely to be near a *true* boundary, but that true boundaries were often farther from estimated boundaries. This result is ambiguous in its implications. One possible interpretation is that the algorithm is not placing enough boundaries, but that those that it does place are equally likely to fall around the same distance from where they should according to the human listener. To say that the algorithm is not generating enough boundaries at long durations, we should expect that there would be more ground-truth boundaries that have no

estimated boundary that falls nearby. *HRt3s$_R$* measures the rate at which an estimated boundary falls within 3 seconds of a ground-truth boundary. Indeed, a negative correlation between duration and *HRt3s$_R$* demonstrates that at longer durations, ground-truth boundaries are more likely to be missed by the estimations. Likewise, to demonstrate that the boundaries that *are* being placed are not necessarily inaccurate, we would expect that the likelihood of an estimated boundary to be placed near a ground-truth boundary is not affected by duration. The *HRt3s$_P$* metric tests this by measuring the rate at which there is a ground-truth boundary within 3 seconds of each estimated boundary. What we find is that *HRt3s$_P$* actually has a significant but weak *positive* correlation with duration, indicating that the placed boundaries actually seem to fall close to ground-truth boundaries *more* often at longer durations. Without a corresponding correlation in *MDt E to R*, this effect merits further exploration. Regardless, we have further evidence that longer durations could possibly indicate that the algorithm will not generate as many estimated boundaries compared to the human listeners; in other words, it seems possible that longer durations result in under-segmentation.

We might expect that a failure to create enough boundaries might result in an increased likelihood that paired frames in the estimation are also paired in the ground-truth. Consider a situation where the algorithm estimates that there is only one all-encompassing section in a song, even while the ground-truth divides it into multiple sections. Even if the estimation is not accurate, we would expect that frames that belong to a common label in the estimation to be all frames. As a result, if we are examining the set of pairs that share a common label in the ground-truth and in the estimation in terms of the labels in the estimation, we expect that value to be maximized. This particular

measurement is what PWF_R measures, and indeed we find that longer duration is significantly and relatively strongly correlated with this metric. In the same vein, as duration correlates with a better pair-wise recall, it is also correlated with a worse pair-wise precision PWF_P. This seeks to measure the frames that share a common label in the ground-truth and in the estimation in terms of the labels of the ground-truth. In the case of under-segmentation, PWF_R should be higher but balanced in the harmonic mean by a lower PWF_P. In fact, because the correlation between duration and the recall rate is so much stronger than that of the precision rate, we actually see an overall significant positive correlation between duration and PWF_F. For further evidence of under-segmentation, we see that there is a significant positive correlation between duration and S_O and a corresponding negative correlation between duration and S_U. Due to the methods in which these metrics are derived, higher scores in each are the result of a *lack of disagreement errors* in terms of over- or under-segmentation respectively. What this means is that the positive correlation with S_O is interpretable as a correlation with fewer disagreements between ground-truth and estimation as a result of over-segmentation, and the negative correlation with S_U is a correlation with *more* disagreements as a result of under-segmentation. This bolsters the evidence that longer duration is correlated with a higher likelihood of under-segmentation.

Other results did not have as many strong correlations as duration, although that does not mean they might not be impactful on the evaluation of the spectral clustering algorithm. For example, it was observed that tempo had a significant positive correlation with the value of S_F. Given that higher values of the components of this metric are derived from a lack of disagreement errors, we might interpret this correlation as

evidence that higher tempo may result in fewer of such errors. While further research would have to demonstrate this more clearly, one possible explanation for this result is that the spectral clustering algorithm uses a beat-synchronous CQT and MFCC in the formulation of its similarity matrix. The practical ramification is that the feature vectors for each frame of the beat-synchronous representations are in fact the mean values of the multiple frames that fall within that beat. Additionally, the matrix used follows the model of [20] in that frames are concatenated with the information from previous frames in order to account for longer-term changes in features. One may expect that at slower tempos, features can more easily vary within each beat given that each beat accounts for a longer duration of time. Thus, a beat-synchronous representation may fail to account for necessary changes, and the multiple-beat concatenation may even multiply this effect. This is only a hypothesis, however, and further study may or may not bear this out.

Class was shown to have a significant effect on some of the evaluation metrics regarding boundary detection. As the researchers who designed the spectral clustering algorithm wrote in [21], “Features built to detect repeated chord progressions may work well for characterizing some genres (e.g., rock or pop), but fail for other styles (e.g., jazz or hip-hop) which may be structured around timbre rather than melody.” The spectral clustering algorithm is designed to compensate for this effect by utilizing both the CQT harmonic features “for detecting long-range repeating forms” as well as the MFCC timbre-related features “for detecting local consistency.”[1]. While the weighting of these features has been successful in that class had no significant effect on the more general metrics of *PWF* or *S*, boundary detection in terms of *MDt E to R*, $HRt3s_F$, and $HRt3s_R$, did seem to be correlated with lower values among the class *jazz* than the class *popular*.

Taken collectively, these evaluation metrics show that boundaries estimated for songs under the class *jazz* were significantly likely to be farther away from ground-truth boundaries than those in the class *popular*, and that estimated boundaries fell within 3 seconds of ground-truth boundaries at a lower rate among *jazz* songs than among *popular* songs. The latter effect on $HRt3s_R$ seems to have been strong enough to affect the mean value of $HRt3s_F$. This may indicate that while the harmonic features from the CQT are properly allowing for parity between the two classes when it comes to labeling, the timbral features from the MFCC may not be providing enough input to result in accurate boundary detection in the *jazz* class.

Although genre was not shown to have many significant correlations overall, it was shown to have some significant effect on the most commonly used general metric of S_F. Higher rates of S_F indicate that there are fewer errors resulting from either over- or under-segmentation as well as general agreement in the labeling of segments. An analysis of the mean S_F values between genres determined that none rose significantly above or dipped significantly below the mean S_F of the whole collection, a pair-wise comparison between genres gives us some indication as to the significant differences that arise between them. As a disclaimer, comparing genres pair-wise reduces the sample size of what is being compared from 143 to 10 or fewer. These values indicate only the possible values that may or may not be confirmed by a more comprehensive analysis. Even so, such an analysis may be quite useful to perform based on the findings in this study. Genre is something that is often known, at least in broad strokes, about a collection in terms of its scope. The Southern Folk-life collection at UNC Chapel Hill, for example, may wish to know that this algorithm performs significantly worse in values of S_F on the genres of

“Modern Folk - Singer/Songwriter” and “Blues – Country Blues” in comparison to 19 out of the 32 genres examined in this study given that the scope of their music collection includes many pieces of music that may well fall within those genres. While this information is less actionable to those implementing the spectral clustering algorithm for describing their collection, it is nonetheless relevant to those at the deciding stage for systems of description they may use. With the kinds of information provided by this study, those who are seeking to employ this algorithm for describing a collection would better understand the limitations of the algorithm given the specific characteristics of the collection.

Results tables

Table 8. Genre and S_F

Level	N	Mean S_F	Std Error	Lower 95%	Upper 95%
Alternative_Pop___Rock	5	0.562043	0.04871	0.46552	0.65856
Blues_-_Contemporary_Blues	5	0.54378	0.04871	0.44726	0.6403
Blues_-_Country_Blues	5	0.423507	0.04871	0.32699	0.52003
Blues_-_Urban_Blues	5	0.428002	0.04871	0.33148	0.52452
Country	3	0.522994	0.06289	0.39839	0.6476
Dance_Pop	3	0.514647	0.06289	0.39004	0.63925
Electronica	5	0.613374	0.04871	0.51685	0.70989
Hip_Hop___Rap	5	0.565755	0.04871	0.46923	0.66227
Humour	2	0.559028	0.07702	0.40642	0.71164
Instrumental_Pop	5	0.529122	0.04871	0.4326	0.62564
Jazz_-_Acid_Jazz	5	0.566546	0.04871	0.47003	0.66307
Jazz_-_Avant-Garde_Jazz	3	0.599126	0.06289	0.47452	0.72373
Jazz_-_Bebop	4	0.616886	0.05446	0.50897	0.7248
Jazz_-_Cool_Jazz	5	0.502802	0.04871	0.40628	0.59932
Jazz_-_Hard_Bop	5	0.610432	0.04871	0.51391	0.70695
Jazz_-_Latin_Jazz	5	0.618476	0.04871	0.52196	0.715
Jazz_-_Post-Bop	5	0.562355	0.04871	0.46583	0.65887
Jazz_-_Soul_Jazz	5	0.597663	0.04871	0.50114	0.69418
Jazz_-_Swing	5	0.562923	0.04871	0.4664	0.65944
Modern_Folk_-_Alternative_Folk	5	0.593867	0.04871	0.49735	0.69039
Modern_Folk_-Singer___Songwriter	5	0.411234	0.04871	0.31471	0.50775
R_B_-_Contemporary_R_B	5	0.48891	0.04871	0.39239	0.58543
R_B_-_Funk	5	0.671183	0.04871	0.57466	0.7677
R_B_-_Gospel	5	0.605283	0.04871	0.50876	0.7018
R_B_-_Rock___Roll	3	0.544208	0.06289	0.4196	0.66881
R_B_-_Soul	5	0.515823	0.04871	0.4193	0.61234
Reggae	4	0.600778	0.05446	0.49287	0.70869
Rock_-Alternative_Metal___Punk	5	0.63489	0.04871	0.53837	0.73141

Rock_-_Classic_Rock	6	0.625543	0.04447	0.53743	0.71365
Rock_-_Metal	5	0.563916	0.04871	0.4674	0.66044
Rock_-_Roots_Rock	5	0.5844	0.04871	0.48788	0.68092

Table 9. Genre and S_F Connecting Letters Report

Genres that do not share a letter in common are significantly different.

Genre						Mean S_F
R_B_-_Funk	A					0.67118279
Rock_-_Alternative_Metal___Punk	A	B				0.63489022
Rock_-_Classic_Rock	A	B				0.62554329
Jazz_-_Latin_Jazz	A	B	C			0.61847567
Jazz_-_Bebop	A	B	C			0.61688584
Electronica	A	B	C			0.61337448
Jazz_-_Hard_Bop	A	B	C			0.61043240
R_B_-_Gospel	A	B	C			0.60528299
Reggae	A	B	C			0.60077823
Jazz_-_Avant-Garde_Jazz	A	B	C			0.59912626
Jazz_-_Soul_Jazz	A	B	C			0.59766332
Modern_Folk_-_Alternative_Folk	A	B	C			0.59386694
Rock_-_Roots_Rock	A	B	C			0.58440027
Jazz_-_Acid_Jazz	A	B	C			0.56654557
Hip_Hop___Rap	A	B	C			0.56575475
Rock_-_Metal	A	B	C	D		0.56391648
Jazz_-_Swing	A	B	C	D		0.56292296
Jazz_-_Post-Bop	A	B	C	D		0.56235483
Alternative_Pop___Rock	A	B	C	D		0.56204331
Humour	A	B	C	D	E	0.55902833
R_B_-_Rock___Roll	A	B	C	D	E	0.54420774
Blues_-_Contemporary_Blues	A	B	C	D	E	0.54378000
Instrumental_Pop		B	C	D	E	0.52912171

Genre						Mean S_F
Country	A	B	C	D	E	0.52299386
R_B_-_Soul		B	C	D	E	0.51582274
Dance_Pop	A	B	C	D	E	0.51464747
Jazz_-_Cool_Jazz		B	C	D	E	0.50280246
R_B_-_Contemporary_R_B			C	D	E	0.48891003
Blues_-_Urban_Blues				D	E	0.42800191
Blues_-_Country_Blues					E	0.42350655
Modern_Folk_-_Singer___Songwriter					E	0.41123357

Table 10. Class and HRt3s_F

Class	N	Mean	Std Error	Lower 95%	Upper 95%
jazz	80	0.355574	0.01818	0.31964	0.39151
popular	63	0.422887	0.02048	0.38239	0.46338

Table 11. Class and MDt E to R

Class	N	Mean	Std Error	Lower 95%	Upper 95%
jazz	80	6.83993	0.44521	5.9598	7.7201
popular	63	4.22081	0.5017	3.229	5.2126

Results figures

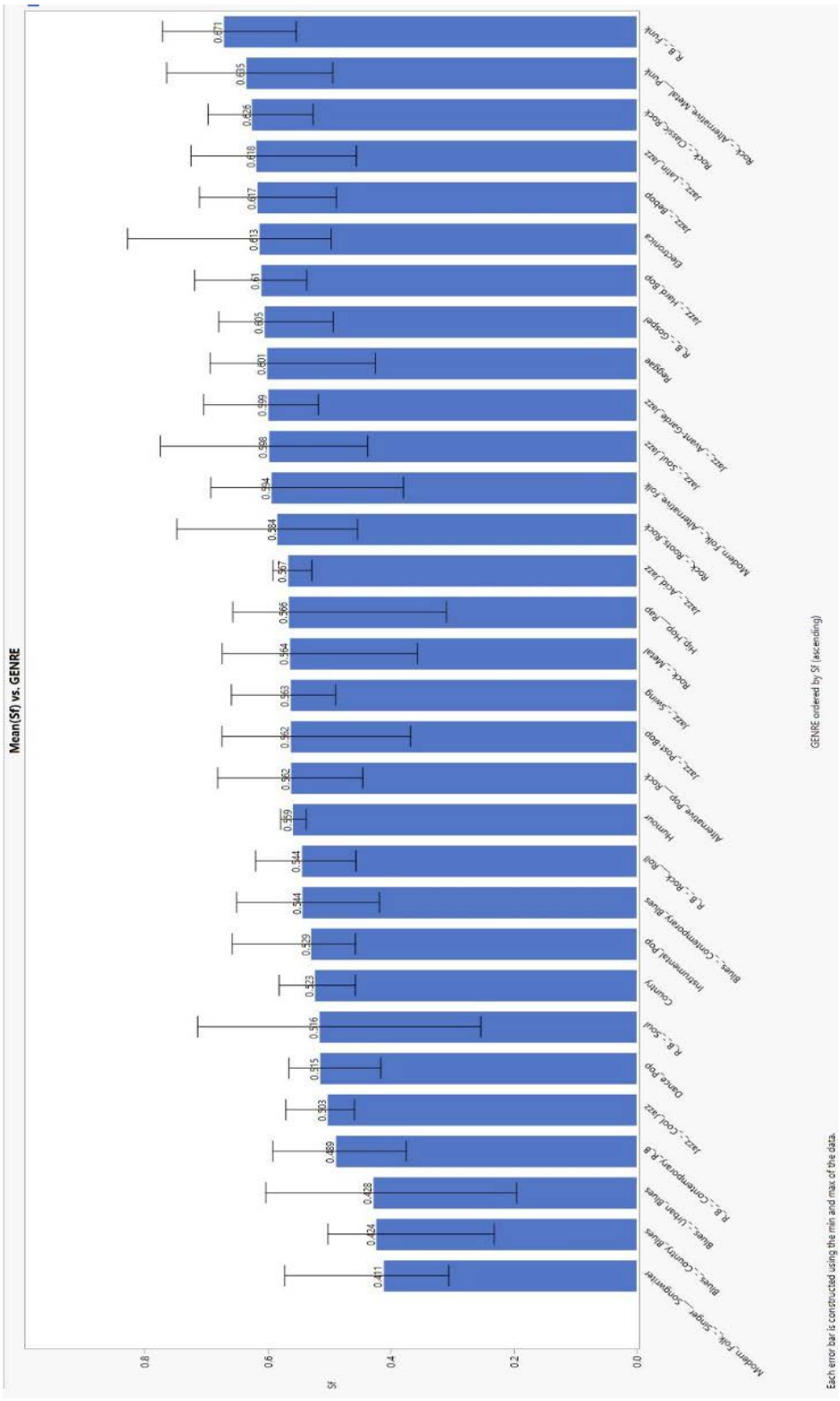

Figure 7. Mean S_F by Genre, ordered by ascending mean S_F. Error bars show range of S_F values in each genre

Figure 9. HRt3s$_F$ by Class

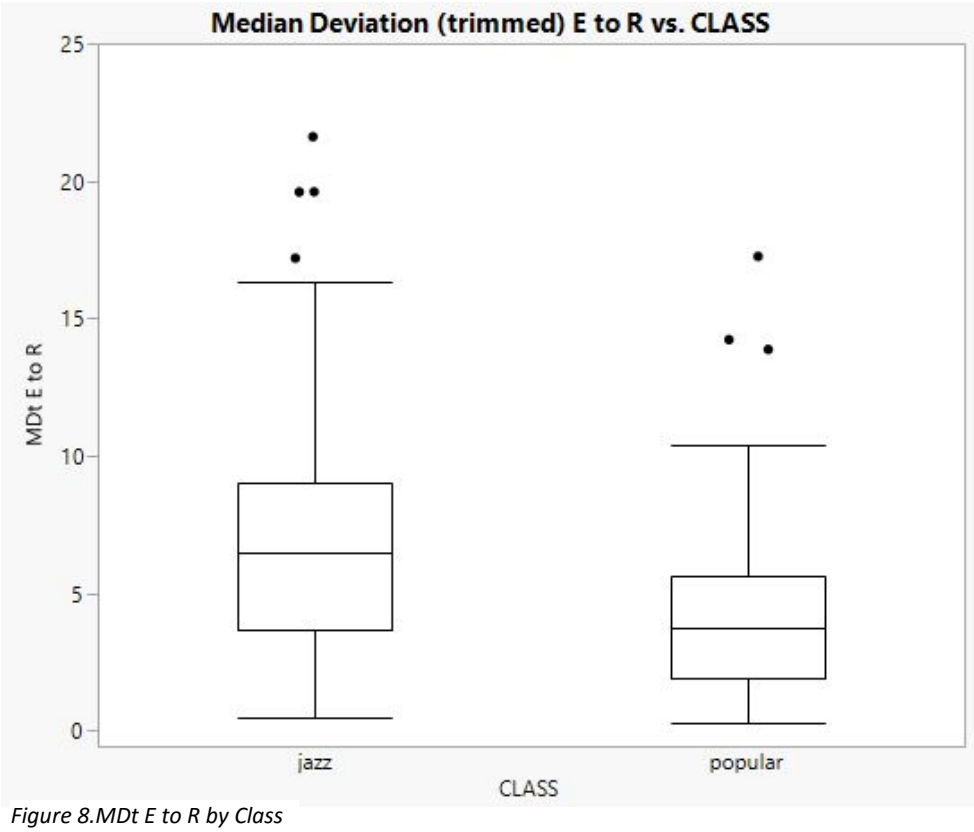

Figure 8.MDt E to R by Class

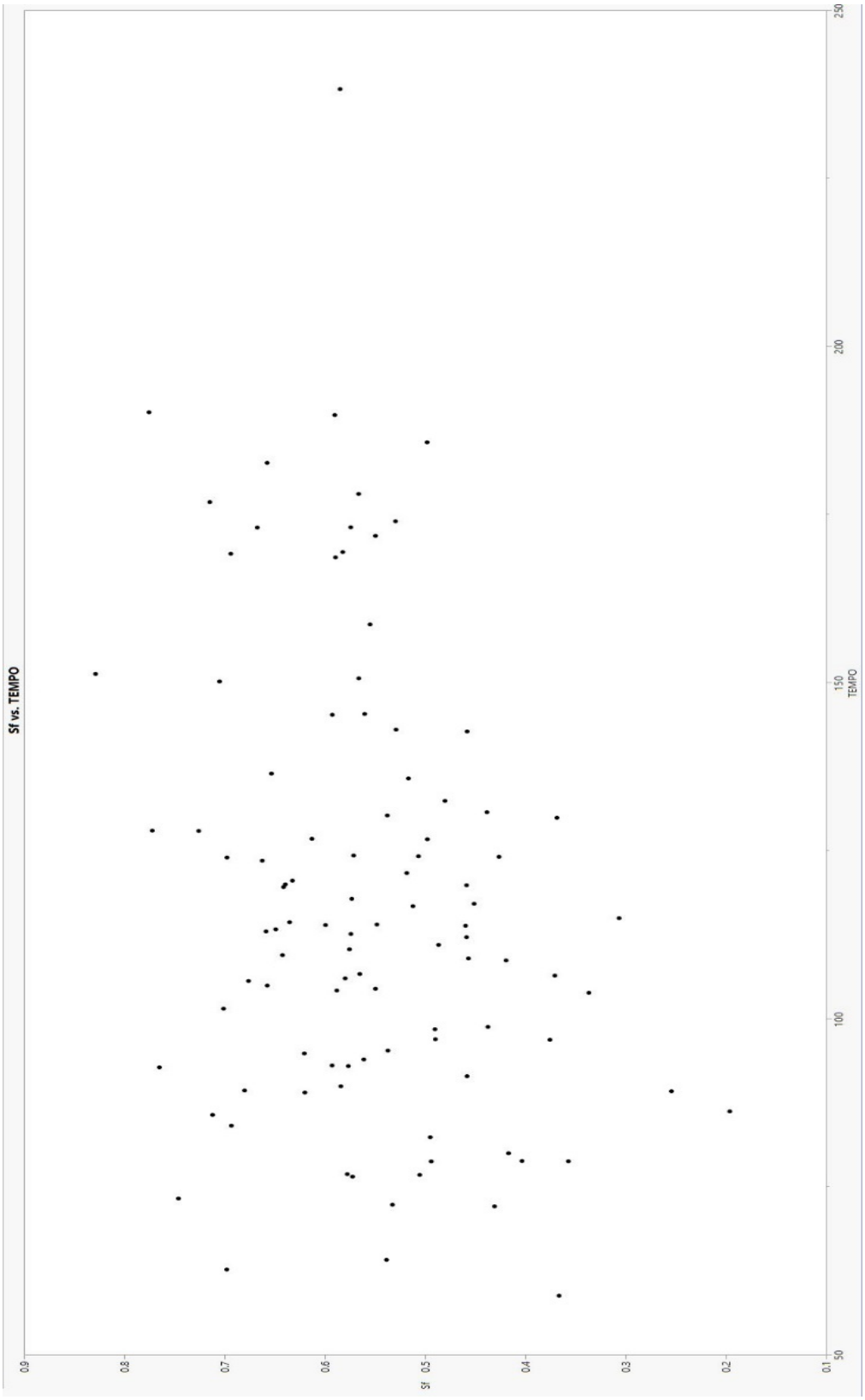

Figure 10. S_F by tempo.

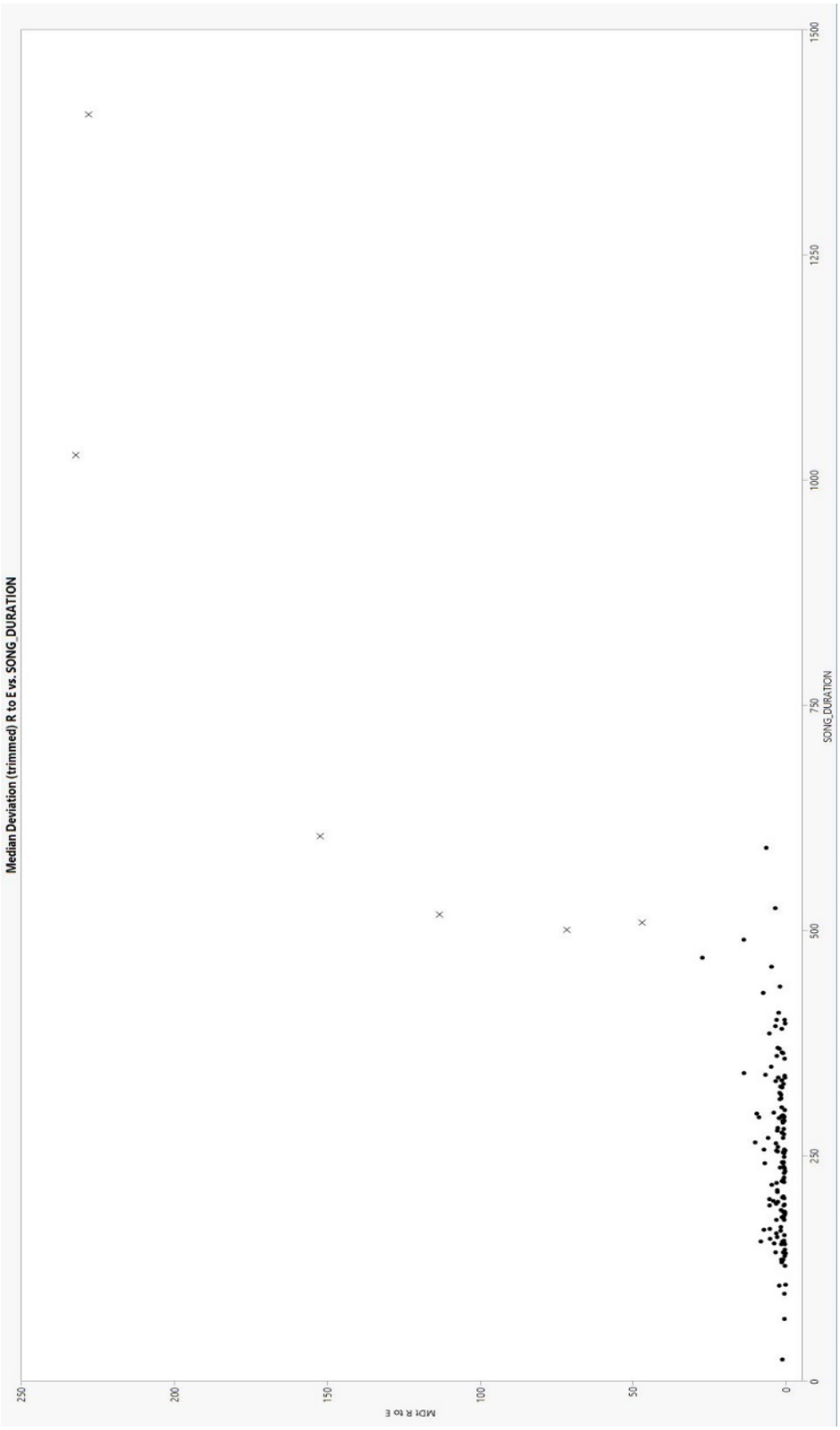

Figure 11. Median Deviation (trimmed) R to E by Song Duration. The Outliers are marked X. The following plot shows these results without these outliers.

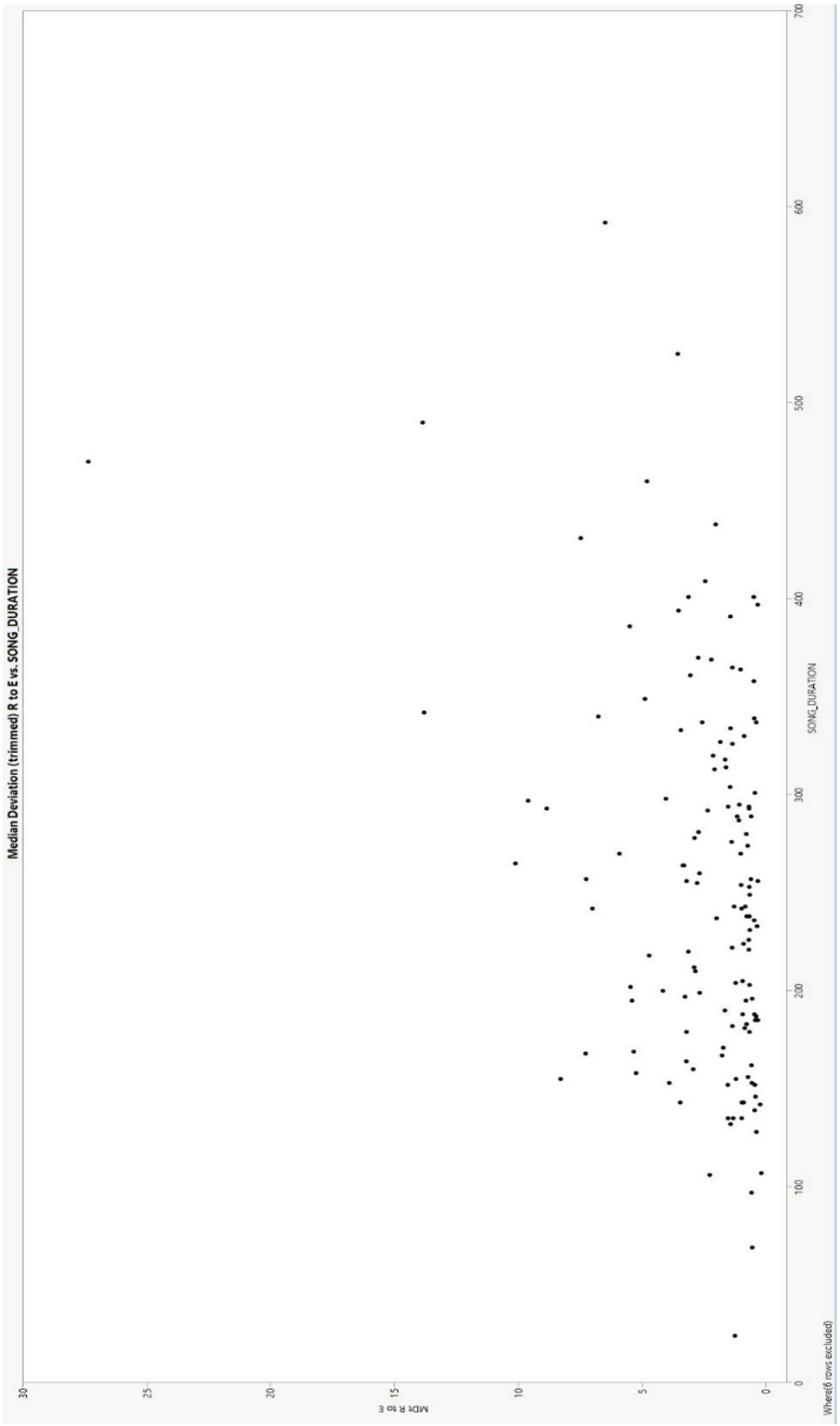

Figure 12. Median Deviation (trimmed) R to E by Song Duration without outliers.

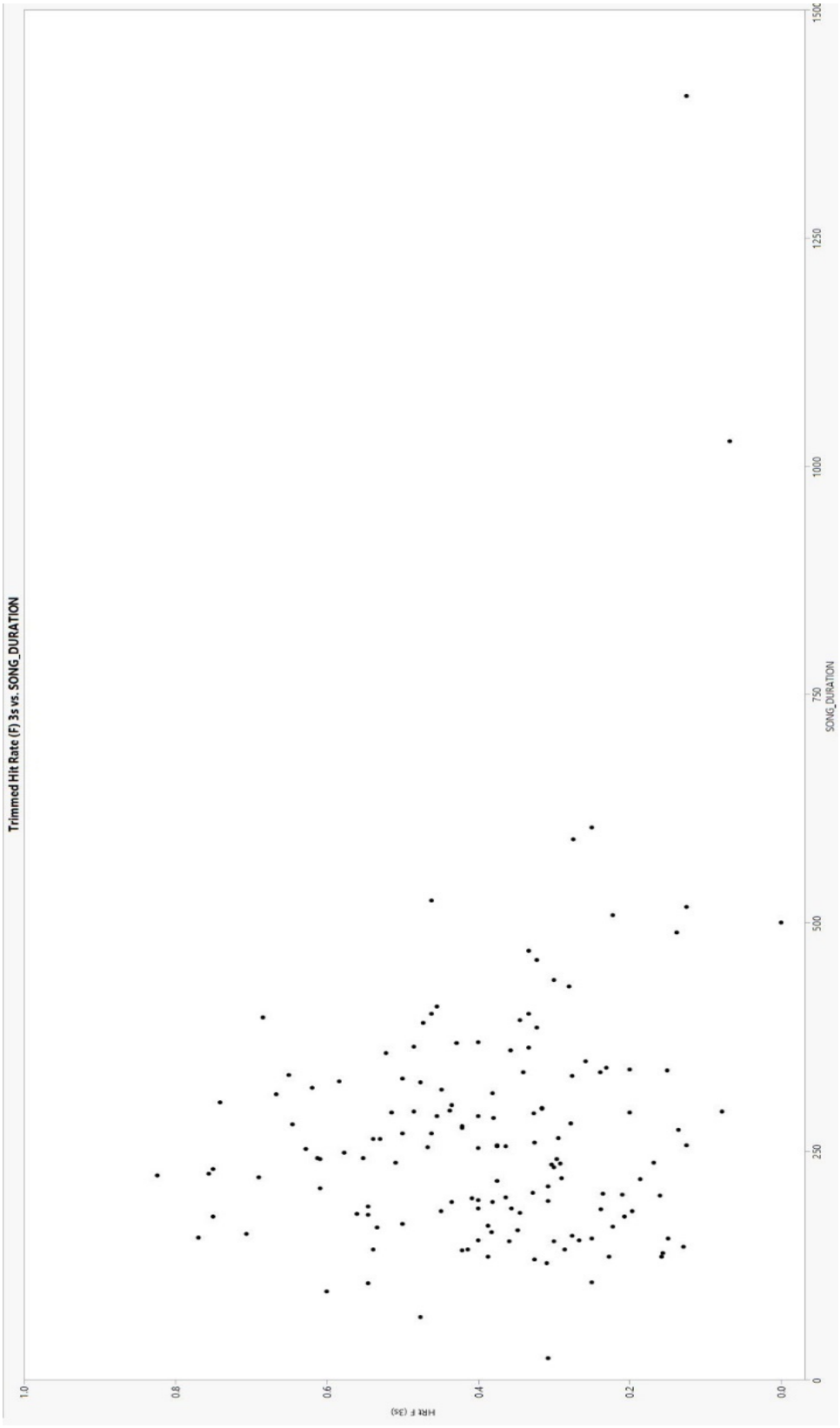

Figure 13. Trimmed Hit Rate (F) at 3 seconds by Song Duration

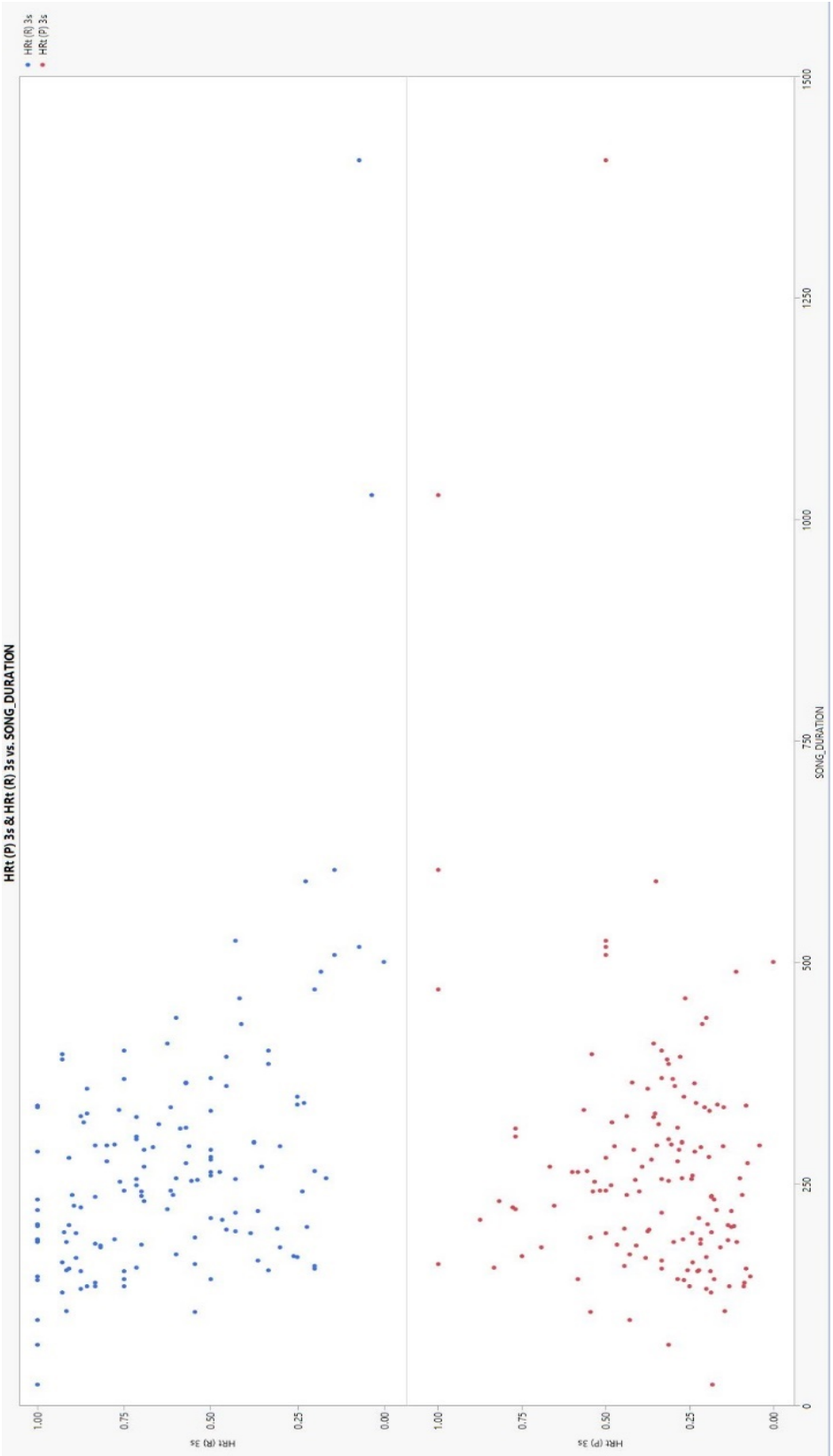

Figure 14. Top- Trimmed Hit Rate (R) at 3s by Song Duration. Bottom – Trimmed Hit Rate (P) at 3s by Song Duration.

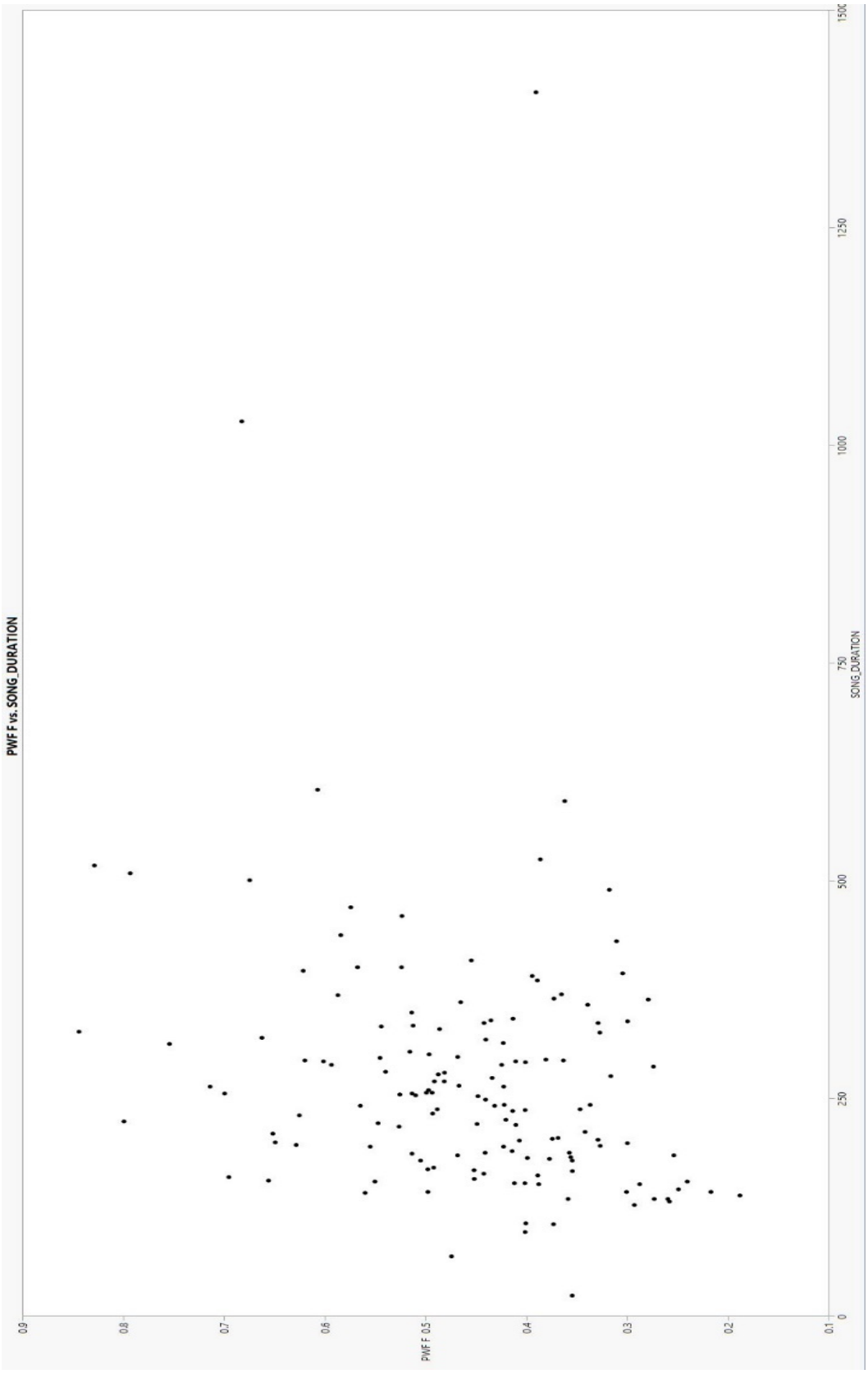

Figure 15. PWF$_F$ by Song Duration.

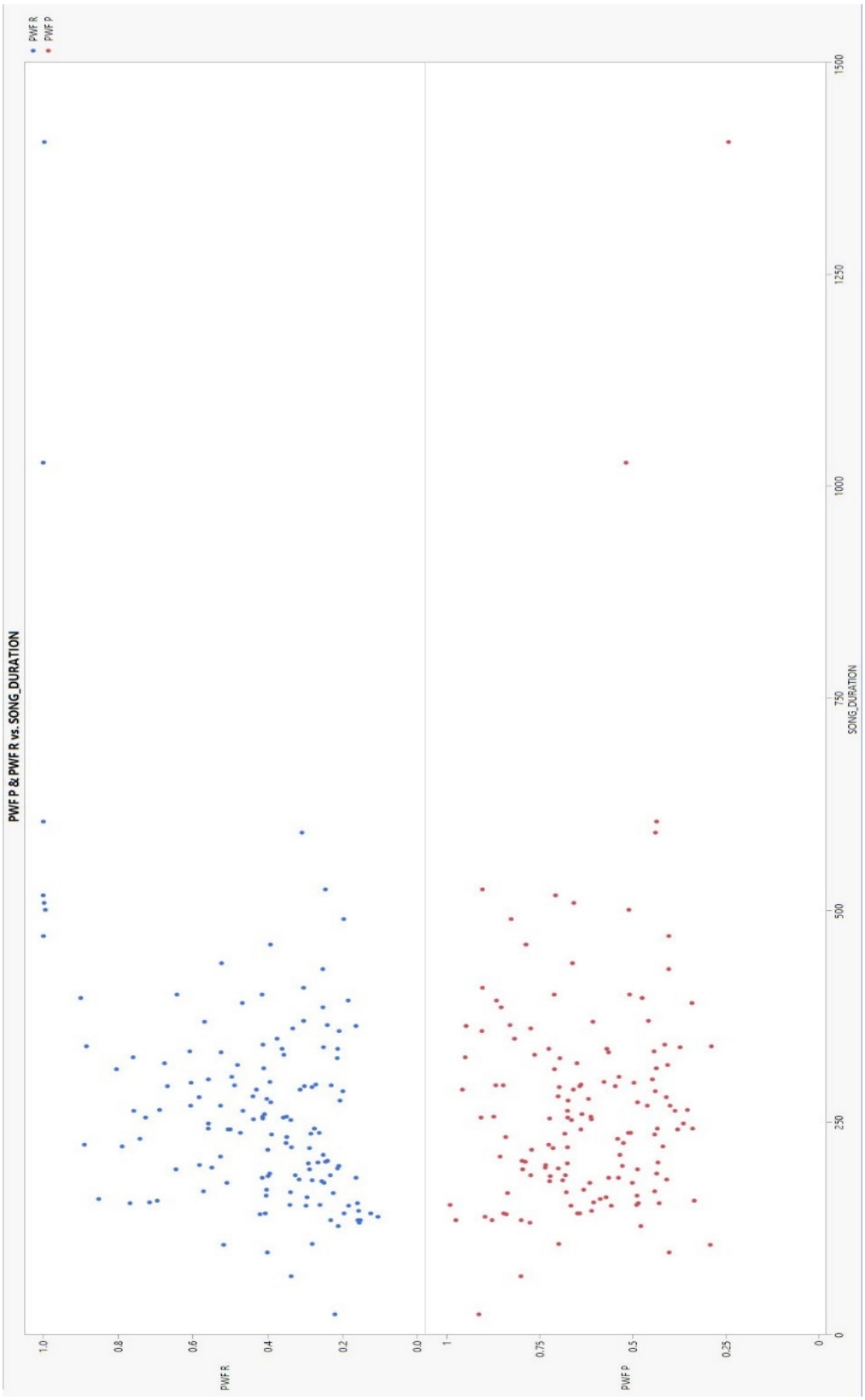

Figure 16. Top - PWF_R by Song duration. Bottom – PWF_P by Song Duration.

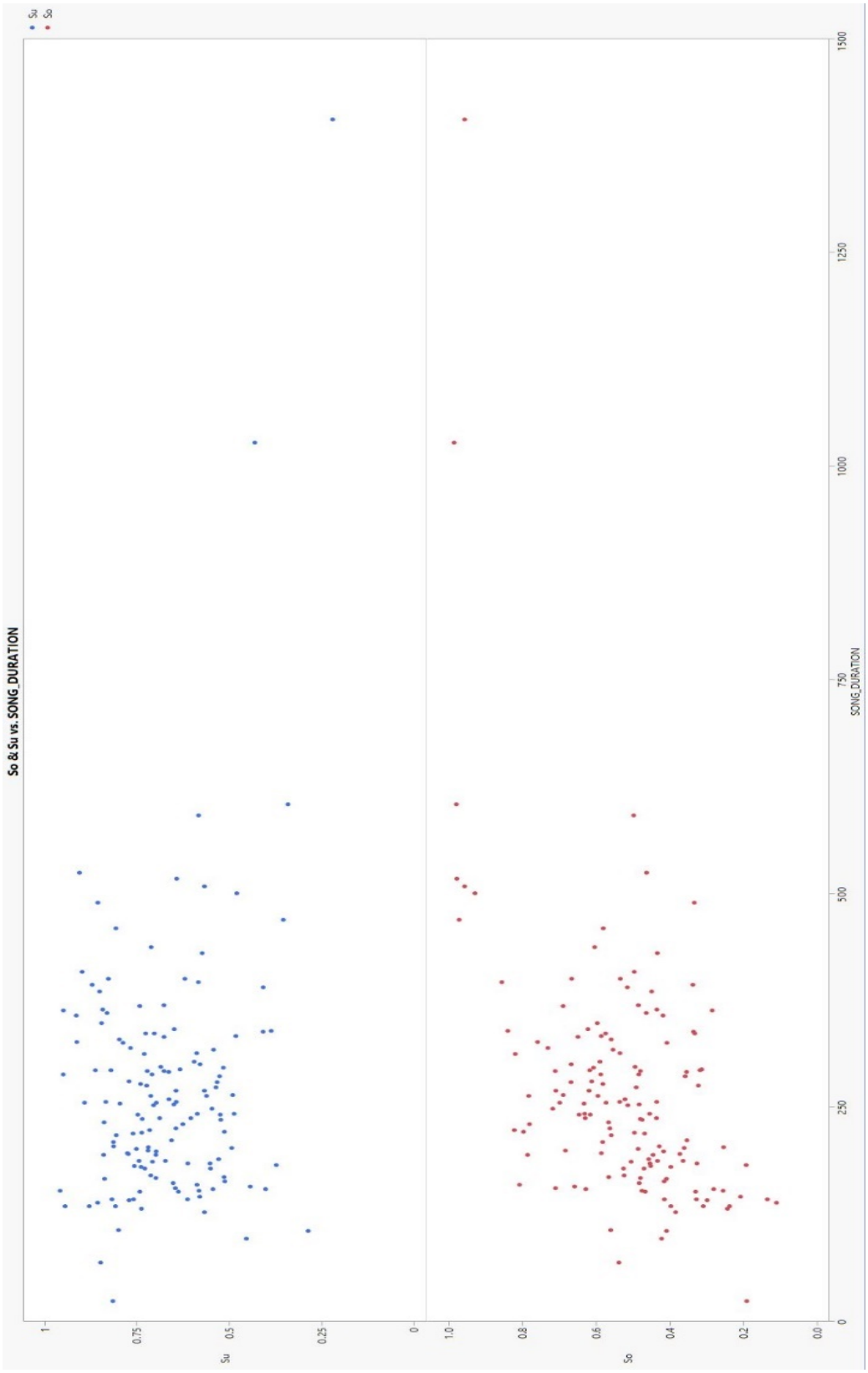

Figure 17. Top – S_U by Song Duration. Bottom – S_O by Song Duration.

V. Conclusion

This study has presented an evaluation of the spectral clustering algorithm for music segmentation in terms of genre, class, tempo, song duration, and time signature. This presentation differs from the standard evaluation of segmentation algorithms that compare multiple algorithms against a collection. This evaluation has instead focused on one algorithm in the context of multiple variables within the collection. This was done to determine the effect these variables may or may not have had on various categories of performance of the algorithm including boundary identification and labeling accuracy. It has revealed that the duration of a song is correlated with many evaluation metrics in both categories. Tempo, class, and genre were also shown to have a significant effect on evaluation scores. This study has thus demonstrated how the algorithm may be evaluated according to known variables in a collection to predict its likely performance for a given collection where those variables are known. The possible causes and implications of these effects on evaluation scores were explored based on the construction of the spectral clustering algorithm and its potential use. Further research based on larger and representative datasets will need to be conducted to confirm the results of this study and may demonstrate how the algorithm may be adjusted in specific ways to account for worse performance in certain contexts according to the hypotheses presented here.

References

[1] B. McFee and D. P. W. Ellis, “Analyzing Song Structure with Spectral Clustering,” in *ISMIR*, 2014, pp. 405–410.

[2] T. Bertin-Mahieux et al., “Autotagger: A model for predicting social tags from acoustic features on large music databases,” *Journal of New Music Research*, vol. 37, no. 2, pp. 115–135, 2008.

[3] D. Turnbull et al., “Five approaches to collecting tags for music,” in *ISMIR*, 2008, vol. 8, pp. 225–230.

[4] J. Pablo Bello and K. Underwood, “Improving access to digital music through content-based analysis,” *OCLC Systems & Services: International digital library perspectives*, vol. 28, no. 1, pp. 17–31, 2012.

[5] J.-J. Aucouterier et al., “The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music,” *The Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 881–891, 2007.

[6] K. Ellis et al., “Semantic Annotation and Retrieval of Music using a Bag of Systems Representation,” in *ISMIR*, 2011, pp. 723–728.

[7] M. Casey et al., “Analysis of minimum distances in high-dimensional musical spaces,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 1015–1028, 2008.

[8] M. Riley et al., “A text retrieval approach to content-based audio retrieval,” in *ISMIR*, 2008, pp. 295–300.

[9] D. Turnbull et al., “Semantic annotation and retrieval of music and sound effects,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 467–476, 2008.

[10] J.-C. Wang et al., “Learning the Similarity of Audio Music in Bag-of-frames Representation from Tagged Music Data,” in *ISMIR*, 2011, pp. 85–90.

[11] P. Cano and M. Koppenberger, “Automatic sound annotation,” in *Proc. IEEE Workshop Mach. Learn. Signal Process.*, 2004, pp. 391–400.

[12] J.-J. Aucouturier and F. Pachet, “Improving timbre similarity: How high is the sky,” *Journal of negative results in speech and audio sciences*, vol. 1, no. 1, pp. 1–13, 2004.

[13] M. Bay et al., “Evaluation of Multiple-F0 Estimation and Tracking Systems.,” in *ISMIR*, 2009, pp. 315–320.

[14] R. Typke, *Music retrieval based on melodic similarity*. 2007.

[15] J. Paulus et al., “State of the Art Report: Audio-Based Music Structure Analysis,” in *ISMIR*, 2010, pp. 625–636.

[16] J.-J. Aucouturier and M. Sandler. Segmentation of musical signals using hidden Markov models. In *Proc. of 110th Audio Engineering Society Convention*, Amsterdam, The Netherlands, May 2001

[17] J. Foote, “Visualizing music and audio using self-similarity,” in *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, 1999, pp. 77–80.
[18] G. Tzanetakis and P. Cook, “Multifeature audio segmentation for browsing and annotation,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1999, pp. 103–106.
[19] J. Foote, “Automatic audio segmentation using a measure of audio novelty,” in *IEEE International Conference on Multimedia and Expo, 2000. ICME 2000*, 2000, vol. 1, pp. 452–455.
[20] J. Serra et al., “Unsupervised detection of music boundaries by time series structure features,” in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
[21] B. McFee and D. P. W. Ellis, “Learning to segment songs with ordinal linear discriminant analysis,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 5197–5201.
[22] K. Jensen, “Multiple scale music segmentation using rhythm, timbre, and harmony,” *Journal on Advances in Signal Processing*, vol. 2007, no. 1, pp. 1–11, 2006.
[23] J. T. Foote and M. L. Cooper, “Media segmentation using self-similarity decomposition,” *Electronic Imaging 2003*, pp. 167–175, 2003.
[24] O. Nieto and J. P. Bello, “Music segment similarity using 2d-fourier magnitude coefficients,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014*, 2014, pp. 664–668.
[25] M. Levy and M. Sandler, “Structural segmentation of musical audio by constrained clustering,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 318–326, 2008.
[26] J. S. Downie et al., “The music information retrieval evaluation exchange: Some observations and insights,” *Advances in music information retrieval*, pp. 93–115, 2010.
[27] J. S. Downie, “The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research,” *Acoustical Science and Technology*, vol. 29, no. 4, pp. 247–255, 2008.
[28] D. Turnbull et al., “A Supervised Approach for Detecting Boundaries in Music Using Difference Features and Boosting,” in *ISMIR*, 2007, pp. 51–54.
[29] H. M. Lukashevich, “Towards Quantitative Measures of Evaluating Song Segmentation,” in *ISMIR*, 2008, pp. 375–380.
[30] J. B. L. Smith et al. “Design and creation of a large-scale database of structural annotations.” In *Proceedings of the International Society for Music Information Retrieval Conference*, Miami, FL, 2011, pp. 555–60.
[31] Peeters, G., and E. Deruty. “Is music structure annotation multi-dimensional? A proposal for robust local music annotation.” In *Proc. LSAS*, 2009, pp. 75–90.
[32] M. Goto et al., “RWC Music Database: Popular, Classical and Jazz Music Databases.,” in *ISMIR*, 2002, vol. 2, pp. 287–288.
[33] C. McKay et al., “A Large Publicly Accessible Prototype Audio Database for Music Research.,” in *ISMIR*, 2006, pp. 160–163.

[34] O. Nieto and J. P. Bello, “MSAF: Music Structure Analytics Framework,” in *ISMIR*, Málaga, Spain, 2015.
[35] B. McFee et al., “librosa: Audio and music signal analysis in python,” in *Proceedings of the 14th Python in Science Conference*, 2015.
[36] C. Raffel et al., “mir_eval: A transparent implementation of common MIR metrics,” in *ISMIR*, 2014.
[37] M. Davy, “An introduction to statistical signal processing and spectrum estimation,” Springer US, 2006, pp. 21–64.
[38] J. C. Brown, “Calculation of a constant Q spectral transform,” *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.
[39] S. Imai, “Cepstral analysis synthesis on the mel frequency scale,” in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP ’83*, vol. 8, pp. 93–96.
[40] S. S. Stevens et al., “A Scale for the Measurement of the Psychological Magnitude Pitch,” *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.
[41] B. Logan, “Mel Frequency Cepstral Coefficients for Music Modeling,” in *ISMIR*, 2000.
[42] R. Bain, "The Harmonic Series," in *The Harmonic Series (Overtone Series)*, 2003. http://in.music.sc.edu/fs/bain/atmi02/hs/index-noaudio.html.
[43] M. Kennedy, Ed., “Pitch,” *The Oxford Dictionary of Music*.
[44] S. H. Nawab et al., “Identification of musical chords using constant-q spectra,” in *IEEE International Conference Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP’01)*, 2001, vol. 5, pp. 3373–3376.
[45] T. Cho et al., “Exploring common variations in state of the art chord recognition systems,” in *Proceedings of the Sound and Music Computing Conference (SMC)*, 2010, pp. 1–8.
[46] T. Fujishima, “Realtime chord recognition of musical sound: A system using common lisp music,” in *ICMC Proceedings*, 1999, pp. 464–467.
[47] J. P. Bello, “Audio-Based Cover Song Retrieval Using Approximate Chord Sequences: Testing Shifts, Gaps, Swaps and Beats,” in *ISMIR*, 2007, vol. 7, pp. 239–244
[48] J. P. Bello and J. Pickens, “A Robust Mid-Level Representation for Harmonic Content in Music Signals,” in *ISMIR*, 2005, vol. 5, pp. 304–311.
[49] X. Yu et al., “An audio retrieval method based on chromagram and distance metrics,” in *International Conference on Audio Language and Image Processing (ICALIP), 2010*, 2010, pp. 425–428.
[50] A. Sheh and D. P. W. Ellis, “Chord segmentation and recognition using EM-trained hidden Markov models,” in *ISMIR*, 2003, pp. 185–191.
[51] H. Papadopoulos and G. Peeters, “Large-scale study of chord estimation algorithms based on chroma representation and HMM,” in *International Workshop on Content-Based Multimedia Indexing. CBMI’07*, 2007, pp. 53–60.
[52] J. P. Bello, “Measuring structural similarity in music,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2013–2025, 2010

APPENDIX 1. Song metadata by Song ID

SONG ID	SONG TITLE	ARTIST	CLASS	GENRE	SONG DURATION	TIME SIGNATURE	TEMPO
3	Golden_Age	Beck	popular	Alternative Pop/Rock	276		
4	I_close_my_eyes	Shivaree	popular	Alternative Pop/Rock	236	4	126.749
10	How_Beautiful_You_Are	The_Cure	popular	Alternative Pop/Rock	314	4	145.388
11	Coldsweat___Remix__	Sugarcubes	popular	Alternative Pop/Rock	222		
15	Blow_Out	Radiohead	popular	Alternative Pop/Rock	281		
18	Mojo_Boogie	Johnny_Winter	jazz	Blues - Contemporary Blues	287	4	124.141
22	Dangerous_Mood___With_Joe_Cocker__	B_B__King	jazz	Blues - Contemporary Blues	295	3	108.726
24	Blood_On_That_Rock	S_Word	jazz	Blues - Contemporary Blues	202	4	189.858
27	So_Close___So_Far_Away	The_Derek_Trucks_Band	jazz	Blues - Contemporary Blues	278		
28	Exercise_in_C_Major_for_Harmonica	John_Mayall	jazz	Blues - Contemporary Blues	501	4	120.583
30	Honey_Babe	Lightnin___Hopkins	jazz	Blues - Country Blues	155	4	96.988
31	Ramblin___On_My_Mind____Alternate_Take	Robert_Johnson	jazz	Blues - Country Blues	143		
35	The_Last_Mile	Brownie_McGhee	jazz	Blues - Country Blues	292		

SONG ID	SONG TITLE	ARTIST	CLASS	GENRE	SONG DURATION	TIME SIGNATURE	TEMPO
39	Hey_Hey	Eric_Clapton	jazz	Blues - Country Blues	196		
40	Just_Like_A_Bird_Without_A_Feather	Compilations	jazz	Blues - Country Blues	142	4	72.139
43	Howlin___For_My_Darlin__	Howlin___Wolf	jazz	Blues - Urban Blues	153		
44	Straight_From_the_Heart	Mississippi_Heat	jazz	Blues - Urban Blues	337	4	117.142
46	Evil	Muddy_Waters	jazz	Blues - Urban Blues	139	4	86.262
52	Looking_the_World_Over	Big_Mama_Thornton	jazz	Blues - Urban Blues	132	4	58.837
55	I_Cried_My_Eyes_Out	Ronnie_Earl	jazz	Blues - Urban Blues	171		
306	It__s_Just_About_Time_1	Johnny_Cash	popular	Country	128	4	91.497
307	Only_One_And_Only	Gillian_Welch	popular	Country	334		
320	Calling_My_Children_Home	Emmylou_Harris	popular	Country	195	1	169.46
322	Party	Nelly_Furtado	popular	Dance Pop	242	4	178.121
324	One_Kiss_From_You	Britney_Spears	popular	Dance Pop	205	4	93.988
334	Crazy_Little_Thing_Called_Love	Rihanna	popular	Dance Pop	203	4	80.031
338	Feed_Me	Tricky	popular	Electronica	243	4	171.888
339	Another_Day	Jaga_Jazzist	popular	Electronica	210		
342	Glass_Museum	Tortoise	popular	Electronica	327	4	151.346
350	Annie__s_Parlor	Kid_Koala	popular	Electronica	243	3	116.788
352	Neighbors	Gnarls_Barkley	popular	Electronica	185	4	185.804
358	Fu_Gee_La	The_Fugees	popular	Hip Hop/Rap	260	4	90.007
359	Missy__s_Finale	Missy_Elliott	popular	Hip Hop/Rap	24		
364	The_Dusty_Foot_Philosopher	K__naan	popular	Hip Hop/Rap	238	4	182.741

SONG ID	SONG TITLE	ARTIST	CLASS	GENRE	SONG DURATION	TIME SIGNATURE	TEMPO
366	I_Gotcha_Back	GZA	popular	Hip Hop/Rap	301	4	89.072
368	Le_Ou_Marye	Wyclef_Jean	popular	Hip Hop/Rap	330	4	104.972
370	Boy_Band	The_Arrogant_Worms	popular	Humour	220	4	106.049
382	A_Night_on_Dildo	The_Arrogant_Worms	popular	Humour	169	4	64.18
386	Glow_Worm_Cha_Cha_Cha	Compilations	popular	Instrumental Pop	143	4	142.788
392	Go_Slow	Compilations	popular	Instrumental Pop	135	3	72.368
395	James_Bond_Theme	Compilations	popular	Instrumental Pop	107		
396	Minor_Swing	David_Grisman_Quintet	popular	Instrumental Pop	179	4	130.288
400	Big_Town	Compilations	popular	Instrumental Pop	164	4	112.171
402	Lively_Up_Yourself	Charlie_Hunter	jazz	Jazz - Acid Jazz	340	4	143.043
404	Minaret	Erik_Truffaz	jazz	Jazz - Acid Jazz	358	4	112.655
408	Come_As_You_Are	Charlie_Hunter	jazz	Jazz - Acid Jazz	370	3	106.7
410	Betty	Erik_Truffaz	jazz	Jazz - Acid Jazz	257	4	76.554
414	Rebel_Music	Charlie_Hunter	jazz	Jazz - Acid Jazz	280	4	145.257
416	Moods_In_Mambo	Charles_Mingus	jazz	Jazz - Avant-Garde Jazz	255	4	150.213
422	Asmarina___My_Asmara	Ethio_Jazz	jazz	Jazz - Avant-Garde Jazz	298	3	173.156
424	A_Love_Supreme___Part_One__	John_Coltrane	jazz	Jazz - Avant-Garde Jazz	470	4	121.72
428	So_Sorry_Please	Bud_Powell	jazz	Jazz - Bebop	197	4	173.116
431	Monk__s_Mood	Thelonious_Monk	jazz	Jazz - Bebop	188		
432	Lover_Come_Back_To_Me	Coleman_Hawkins	jazz	Jazz - Bebop	1028	4	113.975

SONG ID	SONG TITLE	ARTIST	CLASS	GENRE	SONG DURATION	TIME SIGNATURE	TEMPO
440	Wee___Allen__s_Alley__	Dizzy_Gillespie___Stan_Getz_ ___Sonny_Stitt	jazz	Jazz - Bebop	509	4	85.732
442	Night_and_Day_2	Stan_Getz____Bill_Evans	jazz	Jazz - Cool Jazz	394	4	111.029
444	My_Funny_Valentine	Stan_Getz____J__J__Johnson	jazz	Jazz - Cool Jazz	490	4	132.458
446	Lover_Man	Billie_Holiday	jazz	Jazz - Cool Jazz	179	4	124.325
448	I__ll_Never_Be_The_Same	Coleman_Hawkins	jazz	Jazz - Cool Jazz	212	4	113.873
450	Boplicity	Miles_Davis	jazz	Jazz - Cool Jazz	181	4	135.789
467	The_Kicker	Horace_Silver	jazz	Jazz - Hard Bop	326		
471	Born_To_Be_Blue	Grant_Green	jazz	Jazz - Hard Bop	294		
474	Soy_Califa	Compilations	jazz	Jazz - Hard Bop	386	4	104.24
475	Klachnikov	Marsh_Dondurma	jazz	Jazz - Hard Bop	253		
478	A_Tribute_To_Someone	Herbie_Hancock	jazz	Jazz - Hard Bop	525	4	126.812
480	My_Funny_Valentine	Chucho_Valde_s	jazz	Jazz - Latin Jazz	337	4	119.64
482	Los_Teenagers_Bailan_Changui	Marc_Ribot____Los_Cubanos_ Postizos	jazz	Jazz - Latin Jazz	289	4	127.979
483	Cool_Breeze	Dizzy_Gillespie	jazz	Jazz - Latin Jazz	168		
487	Manha_De_Carnival___Morning_of_the_C a	Stan_Getz	jazz	Jazz - Latin Jazz	349		
492	Eu_E_Voce	Stan_Getz____Astrud_Gilberto	jazz	Jazz - Latin Jazz	152	4	109.008
494	Afro_Blue	The_Derek_Trucks_Band	jazz	Jazz - Post-Bop	342	3	114.392
495	Love_And_Broken_Hearts	Wynton_Marsalis	jazz	Jazz - Post-Bop	460		

SONG ID	SONG TITLE	ARTIST	CLASS	GENRE	SONG DURATION	TIME SIGNATURE	TEMPO
496	Al_Green	Charlie_Hunter	jazz	Jazz - Post-Bop	339	3	129.937
498	In_My_Solitude	Oliver_Jones____Clark_Terry	jazz	Jazz - Post-Bop	289	3	110.368
503	Someday_We__ll_All_Be_Free	Charlie_Hunter	jazz	Jazz - Post-Bop	297		
506	Groovin__	Jack_McDuff	jazz	Jazz - Soul Jazz	318	4	114.054
508	First_Street	Soulive	jazz	Jazz - Soul Jazz	401	4	113.347
514	Politely	Art_Blakey	jazz	Jazz - Soul Jazz	364	4	130.783
516	Low_Down____Dirty	George_Benson	jazz	Jazz - Soul Jazz	518	3	190.263
518	Little_Birdie	Wynton_Marsalis	jazz	Jazz - Soul Jazz	264	4	76.929
523	These_Foolish_Things	Yehudi_Menuhin____Stephane_Grappelli	jazz	Jazz - Swing	199		
524	God_Bless_The_Child	Billie_Holiday	jazz	Jazz - Swing	190	1	98.484
526	Honeysuckle_Rose	Johnny_Hodges	jazz	Jazz - Swing	182	4	150.676
528	Little_Man___You__ve_Had_A_Busy_Day__	Count_Basie____Sarah_Vaughan	jazz	Jazz - Swing	293	3	93.001
531	The_Mooche	Louis_Armstrong____Duke_Ellington	jazz	Jazz - Swing	218		
532	Providence	Ani_DiFranco	popular	Modern Folk - Alternative Folk	438	3	136.519
535	When_the_Day_Is_Short	Martha_Wainwright	popular	Modern Folk - Alternative Folk	226		

SONG ID	SONG TITLE	ARTIST	CLASS	GENRE	SONG DURATION	TIME SIGNATURE	TEMPO
536	Nevada_City___California	Utah_Philips____Ani_DiFranco	popular	Modern Folk - Alternative Folk	401	4	109.499
539	You_Were_Here	Sarah_Harmer	popular	Modern Folk - Alternative Folk	293		
543	The_Footsteps_Die_Out_Forever	Kaki_King	popular	Modern Folk - Alternative Folk	135		
550	Gospel_Train___Orchestral__	Tom_Waits	popular	Modern Folk - Singer/Songwriter	153	4	78.882
552	COWBOY_GROOVE	JEAN_LECLERC	popular	Modern Folk - Singer/Songwriter	146	4	115.009
554	Country_Pie	Bob_Dylan	popular	Modern Folk - Singer/Songwriter	97	4	98.828
556	Singing_To_The_Birds	Lisa_Germano	popular	Modern Folk - Singer/Songwriter	265	4	117.885
562	Ruby_II	Amy_Millan	popular	Modern Folk - Singer/Songwriter	106	3	103.909
567	Flow	Sade	jazz	R&B - Contemporary R&B	274		
568	Green_Eyes	Erykah_Badu	jazz	R&B - Contemporary R&B	605	4	76.807

SONG ID	SONG TITLE	ARTIST	CLASS	GENRE	SONG DURATION	TIME SIGNATURE	TEMPO
570	Bad_Habit	Joss_Stone	jazz	R&B - Contemporary R&B	221	4	93.093
572	Interlude__6_Legged_Griot_Trio___Wear	Me__Shell_Ndege_ocello	jazz	R&B - Contemporary R&B	294	4	119.904
576	Attention___featuring_Raphael_Saadiq__	Kelis	jazz	R&B - Contemporary R&B	204	4	96.908
578	Where_Do_We_Go_from_Here	Jamiroquai	jazz	R&B - Funk	313	4	128.035
579	Baby___You__re_Right___feat	The_Derek_Trucks_Band	jazz	R&B - Funk	254		
583	Mr. Thomas	Donald_Byrd	jazz	R&B - Funk	304		
584	Over_The_Rainbow	Maceo_Parker	jazz	R&B - Funk	256	4	62.743
587	I_Need_More_Time	The_Meters	jazz	R&B - Funk	195		
590	Didn__t_It_Rain	Mahalia_Jackson	jazz	R&B - Gospel	160	4	89.365
591	Glory_Train	Montreal_Jubilation_Gospel_Choir	jazz	R&B - Gospel	237		
594	Since_The_Last_Time	Lyle_Lovett	jazz	R&B - Gospel	431	4	78.826
595	Lo_And_Behold	James_Taylor	jazz	R&B - Gospel	156		
599	Church	Lyle_Lovett	jazz	R&B - Gospel	361		
606	Runaround_Sue	Dion	jazz	R&B - Rock & Roll	162	4	158.698
610	Jeepster	Compilations	jazz	R&B - Rock & Roll	249	4	94.884
615	Lonesome_Town	Compilations	jazz	R&B - Rock & Roll	135		
616	Spooky	Compilations	jazz	R&B - Soul	155	4	106.453

SONG ID	SONG TITLE	ARTIST	CLASS	GENRE	SONG DURATION	TIME SIGNATURE	TEMPO
618	Let__s_Stay_Together	Al_Green	jazz	R&B - Soul	200	4	101.541
622	Rock_And_Roll_Again	Donald_Byrd	jazz	R&B - Soul	369	3	176.907
623	Love_Is_Plentiful	The_Staple_Singers	jazz	R&B - Soul	152		
626	Don__t_Cry_For_Louie	Vaya_Con_Dios	jazz	R&B - Soul	183	4	89.259
631	Ride_Natty_Ride	Bob_Marley	popular	Reggae	231		
634	Refuge	Matisyahu	popular	Reggae	242	4	84.138
635	Johnny_Too_Bad	Compilations	popular	Reggae	185		
636	Alarm___Remix__	Jessy_Moss	popular	Reggae	187	4	168.68
646	Dieu_se_pique	Les_Vulgaires_Machins	popular	Rock - Alternative Metal/Punk	158	4	174.051
650	New_Millenium_Homes	Rage_Against_The_Machine	popular	Rock - Alternative Metal/Punk	224	4	92.806
652	Bailey__s_Walk	The_Pixies	popular	Rock - Alternative Metal/Punk	143	4	82.42
654	Mouth_Of_Ghosts	The_Dillinger_Escape_Plan	popular	Rock - Alternative Metal/Punk	409	4	120.034
658	My_Immortal	Evanescence	popular	Rock - Alternative Metal/Punk	264	4	73.287
662	Free_Four	Pink_Floyd	popular	Rock - Classic Rock	256	4	124.032
663	Shakin___All_Over	Flamin___Groovies	popular	Rock - Classic Rock	365		
664	The_Spy	The_Doors	popular	Rock - Classic Rock	257	3	238.334

SONG ID	SONG TITLE	ARTIST	CLASS	GENRE	SONG DURATION	TIME SIGNATURE	TEMPO
667	Take_It_Back	Cream	popular	Rock - Classic Rock	188		
668	Anyday	Derek_and_the_Dominos	popular	Rock - Classic Rock	397	4	169.225
676	The_Loner	Neil_Young	popular	Rock - Classic Rock	233	4	105.669
678	Estranged	Guns_N___Roses	popular	Rock - Metal	592	4	95.303
680	Now_I_Am_Become_Death_the_Destroyer	Nadja	popular	Rock - Metal	1406	3	78.844
683	I__d_Die_For_You	Bon_Jovi	popular	Rock - Metal	270		
687	More_Human_Than_Human	White_Zombie	popular	Rock - Metal	270		
690	Go_Go_Not_Cry_Cry	Compilations	popular	Rock - Metal	69	4	113.029
694	Factory_Girl	The_Rolling_Stones	popular	Rock - Roots Rock	167	4	104.498
695	Safeway_Cart	Neil_Young	popular	Rock - Roots Rock	391		
696	Imitation_Of_Life	R_E_M_	popular	Rock - Roots Rock	238	4	124.211
703	One_Of_Us	Joan_Osborne	popular	Rock - Roots Rock	320		
708	I_Can__t_Make_You_Love_Me	Bonnie_Raitt	popular	Rock - Roots Rock	333	4	123.564

APPENDIX 2. Evaluation results by Song ID

SONG ID	MDt E2R	MDt R2E	HRt3s$_F$	HRt3s$_P$	HRt3s$_R$	PWF$_F$	PWF$_P$	PWF$_R$	S$_F$	S$_O$	S$_U$
3	2.52678	1.37554	0.421052632	0.285714286	0.8	0.316264	0.673835	0.206621	0.44608	0.322363	0.723899
4	3.75628	0.46571	0.303030303	0.185185185	0.833333	0.413582	0.441266	0.389166	0.497699	0.474928	0.522764
10	14.22869	1.60281	0.380952381	0.285714286	0.571429	0.423015	0.436738	0.410127	0.560241	0.534969	0.588019
11	1.20653	1.35424	0.689655172	0.769230769	0.625	0.547142	0.418933	0.788429	0.624029	0.796449	0.512977
15	7.18281	2.71136	0.277777778	0.192307692	0.5	0.5397	0.700805	0.438821	0.682167	0.611361	0.771522
18	4.209805	1.08194	0.379746835	0.234375	1	0.273923	0.439361	0.198994	0.426179	0.358098	0.526224
22	9.62855	1.06569	0.4375	0.304347826	0.777778	0.380714	0.638994	0.271126	0.419183	0.313215	0.633515
24	8.27722	5.45669	0.16	0.125	0.222222	0.406903	0.675682	0.291105	0.590116	0.485717	0.751682
27	7.00608	2.87515	0.421052632	0.363636364	0.5	0.487394	0.619057	0.401914	0.651211	0.581175	0.740441
28	9.2009	71.66297	0	0	0	0.67454	0.510642	0.993378	0.63221	0.928134	0.479369
30	3.36293	8.28408	0.25	0.333333333	0.2	0.550275	0.429043	0.767001	0.489684	0.627604	0.40146
31	6.180475	0.962165	0.285714286	0.176470588	0.75	0.216901	0.846967	0.124376	0.232048	0.135198	0.818103
35	6.80977	2.34278	0.326530612	0.216216216	0.666667	0.401003	0.697596	0.281373	0.462261	0.354527	0.664054
39	5.07923	0.5461	0.307692308	0.184615385	0.923077	0.326755	0.700452	0.213077	0.502891	0.372565	0.773448
40	10.93558	0.22279	0.421052632	0.266666667	1	0.560054	0.839805	0.420109	0.43065	0.298686	0.771517
43	3.97955	0.55948	0.4	0.255813953	0.916667	0.401561	0.489771	0.340276	0.523217	0.475341	0.581816

SONG ID	MDt E2R	MDt R2E	$HRt3s_F$	$HRt3s_P$	$HRt3s_R$	PWF_F	PWF_P	PWF_R	S_F	S_O	S_U
44	8.059895	2.56369	0.23880597	0.148148148	0.615385	0.328881	0.725682	0.212621	0.450969	0.331915	0.703199
46	8.777145	0.44739	0.15625	0.086206897	0.833333	0.188132	0.897172	0.105084	0.195849	0.110559	0.856889
52	7.59655	1.42102	0.325581395	0.2	0.875	0.258101	0.776529	0.154772	0.366199	0.243514	0.738021
55	6.353955	1.716745	0.5	0.428571429	0.6	0.49198	0.631942	0.402774	0.603775	0.523654	0.712843
306	2.38163	0.37451	0.30952381	0.185714286	0.928571	0.292949	0.478324	0.211126	0.458015	0.384071	0.567219
307	2.7176	1.42327	0.65	0.565217391	0.764706	0.512338	0.442948	0.607507	0.528838	0.585901	0.481904
320	5.60472	0.79528	0.380952381	0.242424242	0.888889	0.422713	0.796136	0.287747	0.582129	0.445257	0.840499
322	4.29859	6.99483	0.296296296	0.4	0.235294	0.431516	0.379675	0.499753	0.566212	0.615677	0.524103
324	5.12216	0.931845	0.327868852	0.196078431	1	0.368569	0.797591	0.239658	0.561057	0.428199	0.813448
334	3.921565	0.65043	0.20979021	0.1171875	1	0.328957	0.432419	0.265445	0.416674	0.360816	0.492995
338	0.814675	0.82667	0.551724138	0.5	0.615385	0.422029	0.339286	0.558145	0.549463	0.630686	0.486773
339	0.54521	2.83846	0.608695652	0.875	0.466667	0.651558	0.856506	0.525754	0.678612	0.58188	0.813918
342	2.722595	1.83438	0.583333333	0.4375	0.875	0.843952	0.950343	0.758984	0.828739	0.758455	0.91338
350	1.78666	1.2771	0.612244898	0.517241379	0.75	0.336686	0.434192	0.274942	0.512137	0.454456	0.586589
352	3.64857	0.318425	0.448979592	0.297297297	0.916667	0.468357	0.538718	0.414253	0.497922	0.453631	0.551797
358	4.51442	2.67206	0.325581395	0.24137931	0.5	0.497143	0.636603	0.407806	0.583965	0.521405	0.663583
359	3.92417	1.24227	0.307692308	0.181818182	1	0.354485	0.913623	0.219903	0.309952	0.191336	0.815529
364	0.81341	0.65467	0.509090909	0.4375	0.608696	0.488404	0.505653	0.472293	0.657516	0.629271	0.688415

SONG ID	MDt E2R	MDt R2E	$HRt3s_F$	$HRt3s_P$	$HRt3s_R$	PWF_F	PWF_P	PWF_R	S_F	S_O	S_U
366	13.87381	0.44016	0.434782609	0.3125	0.714286	0.496572	0.447386	0.557909	0.619923	0.667199	0.578903
368	6.146915	0.870705	0.5	0.352941176	0.857143	0.4862	0.763905	0.356574	0.657418	0.558926	0.798048
370	6.226725	3.12152	0.186046512	0.125	0.363636	0.410415	0.714296	0.287924	0.579648	0.468126	0.760923
382	2.343765	5.32644	0.387096774	0.75	0.26087	0.497782	0.441151	0.571094	0.538409	0.56559	0.51372
386	4.34214	0.89365	0.413793103	0.285714286	0.75	0.30064	0.648611	0.195667	0.45797	0.327827	0.75947
392	5.31288	0.97013	0.387096774	0.25	0.857143	0.358648	0.79973	0.231157	0.532442	0.39691	0.808526
395	3.207275	0.18161	0.25	0.144736842	0.916667	0.400563	0.699061	0.280703	0.658987	0.560162	0.800151
396	0.71979	0.65433	0.75	0.692307692	0.818182	0.504964	0.501326	0.508656	0.537685	0.525359	0.550603
400	3.204335	3.20433	0.347826087	0.333333333	0.363636	0.442236	0.48861	0.403902	0.458525	0.415278	0.511825
402	6.604285	6.76095	0.2	0.166666667	0.25	0.435182	0.288668	0.883709	0.528844	0.839348	0.386036
404	19.59819	0.47864	0.52173913	0.375	0.857143	0.339213	0.905606	0.208691	0.573945	0.418109	0.914966
408	6.59383	2.72468	0.4	0.333333333	0.5	0.365153	0.457928	0.303637	0.565069	0.484986	0.67683
410	7.894785	7.244625	0.125	0.1	0.166667	0.499238	0.873634	0.349472	0.572307	0.43545	0.834621
414	2.95196	0.78095	0.64516129	0.5	0.909091	0.481246	0.40983	0.582804	0.592564	0.667722	0.532613
416	4.27547	2.77075	0.466666667	0.411764706	0.538462	0.525585	0.723444	0.41271	0.70505	0.632471	0.796445
422	9.79882	4.02985	0.315789474	0.272727273	0.375	0.468239	0.577522	0.393734	0.574132	0.49387	0.685546
424	1.33825	27.35227	0.333333333	1	0.2	0.574297	0.403062	0.998492	0.518196	0.971072	0.353388
428	7.221405	3.258	0.4	0.375	0.428571	0.628343	0.735132	0.548643	0.667422	0.585549	0.775911

SONG ID	MDt E2R	MDt R2E	$HRt3s_F$	$HRt3s_P$	$HRt3s_R$	PWF_F	PWF_P	PWF_R	S_F	S_O	S_U
431	6.775055	0.92884	0.4	0.269230769	0.777778	0.357348	0.77447	0.232257	0.488834	0.363904	0.744385
432	2.030935	232.1121	0.06779661	1	0.035088	0.682455	0.518198	0.99917	0.599269	0.985012	0.430629
440	12.19177	47.0614	0.222222222	0.5	0.142857	0.793027	0.658377	0.996914	0.712019	0.956334	0.567134
442	6.74757	3.51782	0.344827586	0.277777778	0.454545	0.304451	0.866249	0.184679	0.486531	0.337449	0.871591
444	21.60895	13.84894	0.137931034	0.111111111	0.181818	0.317767	0.82683	0.196677	0.480031	0.333478	0.856389
446	9.07766	3.197265	0.206896552	0.157894737	0.3	0.354477	0.615695	0.248884	0.571312	0.469752	0.728898
448	8.521975	2.89041	0.307692308	0.222222222	0.5	0.341889	0.534569	0.251307	0.459598	0.353632	0.656241
450	2.56168	0.8519	0.545454545	0.409090909	0.818182	0.377162	0.72326	0.255094	0.51654	0.397011	0.73905
467	3.70221	1.34322	0.476190476	0.357142857	0.714286	0.327168	0.695914	0.213853	0.537371	0.407711	0.787957
471	9.63501	1.516745	0.484848485	0.347826087	0.8	0.619832	0.848266	0.488327	0.719686	0.617188	0.863007
474	5.19612	5.49152	0.322580645	0.3125	0.333333	0.389114	0.853831	0.251972	0.587971	0.449063	0.851302
475	2.889205	0.66054	0.62745098	0.533333333	0.761905	0.44808	0.664706	0.337945	0.594306	0.513637	0.705036
478	2.777265	3.55084	0.461538462	0.5	0.428571	0.386149	0.903997	0.245511	0.612827	0.462967	0.906142
480	16.30973	0.380705	0.340425532	0.205128205	1	0.442022	0.569714	0.36109	0.641209	0.573849	0.726486
482	3.718265	0.58744	0.454545455	0.416666667	0.5	0.59338	0.957998	0.429797	0.725878	0.587324	0.949987
483	5.95741	7.26859	0.222222222	0.2	0.25	0.45191	0.678554	0.338761	0.568786	0.479538	0.69885
487	4.5424	4.876065	0.258064516	0.266666667	0.25	0.513875	0.81785	0.374633	0.699785	0.596555	0.846218
492	11.22669	1.532695	0.3	0.1875	0.75	0.287569	0.666219	0.183357	0.45672	0.329794	0.742471

SONG ID	MDt E2R	MDt R2E	HRt3s$_F$	HRt3s$_P$	HRt3s$_R$	PWF$_F$	PWF$_P$	PWF$_R$	S$_F$	S$_O$	S$_U$
494	7.13324	13.79265	0.230769231	0.230769231	0.230769	0.413282	0.414253	0.412316	0.635099	0.621695	0.649093
495	8.0341	4.79492	0.322580645	0.263157895	0.416667	0.523462	0.78705	0.392134	0.675258	0.580283	0.807405
496	10.61049	0.46431	0.150537634	0.081395349	1	0.299676	0.37285	0.250511	0.368393	0.335786	0.408015
498	6.020825	1.1525	0.4	0.28125	0.692308	0.424465	0.659598	0.312916	0.575409	0.484016	0.709348
503	11.60996	9.58982	0.315789474	0.272727273	0.375	0.545297	0.496857	0.604203	0.557616	0.606812	0.515799
506	3.903525	1.63844	0.448275862	0.342105263	0.65	0.440332	0.406567	0.480214	0.548003	0.553794	0.542332
508	10.28172	0.48082	0.461538462	0.333333333	0.75	0.523918	0.711325	0.414669	0.649046	0.533828	0.827689
514	14.57639	1.01789	0.333333333	0.235294118	0.571429	0.27918	0.947972	0.163694	0.438255	0.284857	0.949648
516	2.42819	113.2681	0.125	0.5	0.071429	0.828575	0.70771	0.999226	0.775416	0.977366	0.642631
518	2.40498	3.34368	0.529411765	0.6	0.473684	0.422574	0.386864	0.465548	0.577596	0.594914	0.561258
523	2.28092	2.667115	0.408163265	0.37037037	0.454545	0.299884	0.528067	0.2094	0.521813	0.416622	0.698066
524	1.64564	1.64564	0.545454545	0.545454545	0.545455	0.41413	0.43589	0.394439	0.490166	0.456828	0.528755
526	3.93079	1.344425	0.56	0.466666667	0.7	0.399052	0.688468	0.280948	0.566022	0.451887	0.757294
528	10.87513	8.84622	0.2	0.15	0.3	0.410586	0.64255	0.301678	0.57647	0.47965	0.722261
531	4.70755	4.70755	0.375	0.333333333	0.428571	0.526243	0.771823	0.399219	0.660143	0.55879	0.806409
532	10.3866	2.01754	0.3	0.2	0.6	0.584238	0.662225	0.522683	0.653268	0.603628	0.711805
535	1.321405	0.68181	0.755555556	0.653846154	0.894737	0.420351	0.52512	0.350434	0.600568	0.562186	0.644575
536	6.76571	3.1169	0.333333333	0.333333333	0.333333	0.567517	0.508498	0.642036	0.64224	0.66609	0.620039

SONG ID	MDt E2R	MDt R2E	HRt3s$_F$	HRt3s$_P$	HRt3s$_R$	PWF$_F$	PWF$_P$	PWF$_R$	S$_F$	S$_O$	S$_U$
539	0.41071	0.669735	0.514285714	0.473684211	0.5625	0.601359	0.547193	0.667427	0.693169	0.710606	0.676568
543	17.25642	1.523085	0.157894737	0.088235294	0.75	0.273128	0.975614	0.158791	0.380089	0.23788	0.945063
550	4.19463	3.895145	0.266666667	0.222222222	0.333333	0.41196	0.990957	0.26003	0.403209	0.255297	0.958586
552	5.73141	0.41002	0.129032258	0.068965517	1	0.249077	0.610558	0.156451	0.306298	0.208144	0.579635
554	2.52617	0.5756	0.6	0.428571429	1	0.401333	0.402361	0.400311	0.437237	0.422288	0.453283
556	2.00129	10.10564	0.294117647	0.555555556	0.2	0.466821	0.353152	0.688392	0.573018	0.688761	0.490578
562	2.26059	2.26059	0.545454545	0.545454545	0.545455	0.373128	0.291796	0.51732	0.336406	0.408644	0.285871
567	9.22271	0.72993	0.13559322	0.076923077	0.571429	0.433989	0.487049	0.391355	0.512509	0.490959	0.536038
568	1.092245	152.2516	0.25	1	0.142857	0.607125	0.436129	0.998691	0.505348	0.97856	0.340628
570	5.81426	0.68118	0.289855072	0.169491525	1	0.448901	0.67534	0.336181	0.59277	0.495674	0.737172
572	19.60505	0.675725	0.078125	0.040983607	0.833333	0.363353	0.868352	0.229743	0.458485	0.318097	0.820683
576	4.532665	1.21146	0.235294118	0.135135135	0.909091	0.374292	0.788839	0.245354	0.375439	0.253983	0.719516
578	1.1376	2.06657	0.666666667	0.769230769	0.588235	0.754119	0.710565	0.80336	0.772131	0.818981	0.730351
579	4.772415	0.99422	0.4	0.3125	0.555556	0.509882	0.611139	0.43741	0.554173	0.483015	0.649921
583	1.41378	1.4385	0.740740741	0.769230769	0.714286	0.515559	0.537303	0.495506	0.592208	0.589288	0.595159
584	8.08565	3.19556	0.375	0.333333333	0.428571	0.699415	0.674904	0.725772	0.697866	0.69824	0.697492
587	2.345205	5.40183	0.434782609	0.5	0.384615	0.554914	0.487192	0.644503	0.739535	0.785198	0.698892
590	2.244025	2.92778	0.705882353	1	0.545455	0.695147	0.587606	0.850868	0.680001	0.807478	0.587286

SONG ID	MDt E2R	MDt R2E	$HRt3s_F$	$HRt3s_P$	$HRt3s_R$	PWF_F	PWF_P	PWF_R	S_F	S_O	S_U
591	6.227875	1.984715	0.291666667	0.184210526	0.7	0.40129	0.682773	0.284146	0.580891	0.479922	0.735664
594	17.1915	7.47122	0.28	0.212121212	0.411765	0.310616	0.403538	0.252478	0.493855	0.43377	0.573262
595	0.45907	0.71496	0.769230769	0.833333333	0.714286	0.655817	0.605506	0.715246	0.676372	0.709751	0.645992
599	7.22094	3.04621	0.357142857	0.294117647	0.454545	0.46527	0.774464	0.332518	0.595295	0.463668	0.831282
606	2.738605	0.575485	0.382352941	0.240740741	0.928571	0.388713	0.571618	0.294484	0.554842	0.482774	0.652202
610	1.86959	0.64438	0.576923077	0.483870968	0.714286	0.440519	0.363955	0.557877	0.620417	0.717051	0.546736
615	7.812675	1.31277	0.227272727	0.131578947	0.833333	0.259563	0.878283	0.152284	0.457364	0.309037	0.87949
616	4.61986	1.19843	0.149253731	0.081300813	0.909091	0.240443	0.485179	0.159824	0.370434	0.280838	0.543982
618	3.99651	4.1541	0.363636364	0.444444444	0.307692	0.64916	0.734145	0.58181	0.701104	0.683067	0.72012
622	10.98304	2.194805	0.428571429	0.3	0.75	0.587013	0.607176	0.568147	0.714787	0.689049	0.742522
623	7.95632	0.43778	0.358974359	0.225806452	0.875	0.387797	0.557941	0.297174	0.538686	0.46635	0.637583
626	8.20463	0.772755	0.344827586	0.217391304	0.833333	0.356127	0.408854	0.315447	0.254102	0.192832	0.37244
631	0.31857	0.6409	0.75	0.818181818	0.692308	0.625254	0.540832	0.740907	0.69463	0.780654	0.625684
634	1.14081	0.976265	0.608695652	0.538461538	0.7	0.564703	0.639441	0.505607	0.693285	0.646054	0.747967
635	4.482565	0.4228	0.196721311	0.109090909	1	0.253789	0.564741	0.16367	0.425843	0.326424	0.612346
636	5.32008	0.38534	0.238095238	0.135135135	1	0.513529	0.721665	0.398576	0.589354	0.504743	0.708046
646	0.79175	5.23592	0.275862069	0.444444444	0.2	0.45173	0.334713	0.694543	0.529425	0.658074	0.44285
650	0.89701	0.895015	0.823529412	0.777777778	0.875	0.799308	0.726102	0.88893	0.76497	0.821609	0.715637

SONG ID	MDt E2R	MDt R2E	HRt3s$_F$	HRt3s$_P$	HRt3s$_R$	PWF$_F$	PWF$_P$	PWF$_R$	S$_F$	S$_O$	S$_U$
652	1.96949	3.452545	0.538461538	0.583333333	0.5	0.497671	0.642892	0.405968	0.49471	0.414625	0.613138
654	8.383035	2.44075	0.454545455	0.357142857	0.625	0.454697	0.903921	0.303745	0.639344	0.496166	0.898672
658	1.94611	3.30013	0.538461538	0.583333333	0.5	0.713867	0.675296	0.75711	0.746002	0.782443	0.712804
662	6.69281	0.322005	0.363636364	0.243902439	0.714286	0.513471	0.907964	0.357949	0.697642	0.572642	0.89245
663	5.63236	1.34525	0.484848485	0.421052632	0.571429	0.372668	0.829988	0.240276	0.573809	0.434913	0.843052
664	1.742915	0.598805	0.375	0.272727273	0.6	0.493578	0.612743	0.413216	0.584712	0.535583	0.643764
667	4.975865	0.45601	0.356164384	0.216666667	1	0.440988	0.681215	0.326019	0.527178	0.433229	0.673155
668	0.885835	0.32618	0.684210526	0.541666667	0.928571	0.621395	0.474815	0.898892	0.693839	0.855385	0.583618
676	4.819085	0.347345	0.3	0.176470588	1	0.493078	0.841848	0.34864	0.676081	0.565964	0.839399
678	4.52076	6.48578	0.274509804	0.35	0.225806	0.361953	0.438764	0.308029	0.537082	0.497805	0.583089
680	7.088415	227.9324	0.125	0.5	0.071429	0.390521	0.242865	0.996156	0.356995	0.95627	0.219463
683	3.75002	1.01006	0.5	0.391304348	0.692308	0.48148	0.399709	0.605313	0.674972	0.708732	0.644281
687	0.62694	5.90703	0.461538462	0.666666667	0.352941	0.491251	0.461009	0.525739	0.591933	0.618857	0.567254
690	3.42344	0.54161	0.476190476	0.3125	1	0.474302	0.800637	0.336959	0.6586	0.538062	0.848737
694	4.47483	1.76046	0.533333333	0.380952381	0.888889	0.354318	0.836487	0.224761	0.549462	0.408677	0.83822
695	7.3888	1.42763	0.472727273	0.317073171	0.928571	0.394309	0.341078	0.467229	0.455077	0.514893	0.407712
696	6.48412	0.768425	0.168224299	0.092783505	0.9	0.346705	0.512014	0.262088	0.506484	0.435933	0.604281
703	2.42889	2.12481	0.619047619	0.481481481	0.866667	0.662431	0.650097	0.675243	0.748587	0.730352	0.767755

SONG ID	MDt E2R	MDt R2E	$HRt3s_F$	$HRt3s_P$	$HRt3s_R$	PWF_F	PWF_P	PWF_R	S_F	S_O	S_U
708	5.7768	3.426	0.275862069	0.19047619	0.5	0.543931	0.565225	0.524183	0.662392	0.648808	0.676556

APPENDIX 3. Abbreviations and acronyms

2D-FMC – Two-Dimensional Fourier Magnitude Coefficients.

BoF – Bag of Features.

CQT – Constant-Q Transform.

DFT – Discrete Fourier Transform.

F0 – Fundamental frequency.

FT – Fourier Transform.

GMM – Gaussian Mixture Model.

HMM – Hidden Markov Model.

HRt3s – Hit Rate (trimmed) at 3 seconds.

- ***HRt3s$_R$*** – the recall rate of *HRt3s.*
- ***HRt3s$_P$*** – the Precision rate of *HRt3s.*
- ***HRt3s$_F$*** – the harmonic mean of *HRt3s$_R$* and *HRt3s$_P$*.

IR – Information Retrieval.

m – the parameter of the maximum eigenvector in the spectral clustering algorithm.

MFCC – Mel-Frequency Cepstral Coefficients.

MIR – Music Information Retrieval.

MIREX – the Music Information Retrieval Evaluation eXchange.

MDt E to R – Median Deviation (trimmed) from the estimated to the ground-truth boundaries.

MDt R to E – Median Deviation (trimmed) from the ground-truth to the estimated boundaries.

MSAF – Music Structure Analysis Framework.

PWF – Pair-Wise Frame clustering.

- $\boldsymbol{PWF_R}$ – the Recall rate of *PWF*
- $\boldsymbol{PWF_P}$ – the Precision rate of *PWF*
- $\boldsymbol{PWF_F}$ – the harmonic mean of PWF_R and PWF_P.

S – Normalized conditional entropy.

- $\boldsymbol{S_O}$ – the Oversegmentation score as measured by normalized conditional entropy.
- $\boldsymbol{S_U}$ – the Undersegmentation score as measured by normalized conditional entropy.
- $\boldsymbol{S_F}$ – the harmonic mean of S_O and S_U.

SALAMI – Structural Analysis of Large Amounts of Music Information.

SSM – Self-Similarity Matrix.

STFT – Short-Time Fourier Transform.

SVD – Singular Value Decomposition.

TFR – Time and Frequency Representation.

APPENDIX 4. Time and Frequency Representations (TFRs)

The Spectrogram

We understand generally that text can be broken down into data that represents it in some way while reducing its complexity. An index is just that; it represents the words used in a document or collection without preserving the full complexity of their order. Likewise, it is possible to reduce the complexity of a piece of music by representing only certain important *features*. When a music theorist attempts an analysis of a piece of music, they do not often do it solely by listening to it. They use the aid of a particularly famous kind of feature representation: a score. The score is not the piece of music itself; it only represents which notes are played by which instruments on which beats. Digital representations of music are fundamentally similar. What we want to create is what [37] calls a time and frequency representation (TFR). A score is just one type of TFR, where bars and tempo represent time and notes and instrumentation represent frequency. While the TFRs used in digital content-analysis take a different form, they operate under these same simple parameters.

The most basic TFR used in digital music content-analysis is called the spectrogram, defined by [37] as "the Fourier transform of successive signal frames," where the signal is the pertinent piece of music. The Fourier transform is the function which discerns the amplitudes of the constituent harmonics of a signal, a function of time. In other words, given a sound, the Fourier transform represents the relative amplitudes of the frequencies that make up that sound. The mathematics predate this application by more than a century, first proposed by French mathematician Joseph Fourier in 1822 in his *Théorie*

analytique de la chaleur and forming the basis for the Fourier series and Fourier analysis. The most commonly applied variation on the Fourier transform used in the field of digital music content analysis is known as the Discrete Fourier Transform (DFT), which can be defined as (Ex. 3 [37]) where *k* denotes the discrete frequency and *x(n)* denotes signal as a function of discrete time *n*. The result is a complex value representing the amplitudes of the signal at a given range of time in the frequency domain, which is visualized in fig. 18.

Ex. 3

$$\mathrm{DFT}_x(k) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi kn}$$

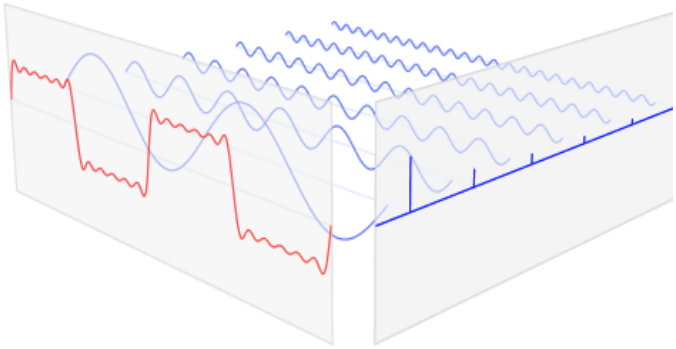

Figure 18. In red, left – the signal over a given range of time. In blue, center– the Fourier series of constituent harmonics in the signal. In blue, right – the amplitudes of the constituent harmonics mapped in the frequency domain. By Lucas V. Barbosa (Own work) [Public domain], via Wikimedia Commons

The Fourier transform (FT) computes only one set of amplitude values for a signal. It cannot capture the nuances of how frequencies and amplitudes may shift in a signal over time; therefore, a Fourier transform of a complete piece of music is not useful as far as discerning internal features is concerned. This process is computed, as [37] indicates, for "successive frames of the signal." These frames represent short, overlapping windows of time in the signal which commonly range from a few dozen milliseconds up to a full second. The window of time in the frame should be short enough that no functional change in frequency amplitudes is expected. A frame of a signal can be represented as (Ex. 4[37]), where $s_n^w[m]$ is the frame localized around discrete time point *n* and computed with the windowing function $w[n - m]$ multiplied by the framed signal

x[*m*]. Windowing functions may take many forms, but the most common include Gaussian (seen in figure 19), Hamming, and Hanning functions. This windowing function is multiplied across the range of the signal, returning successive frames of a given duration. For each frame, a DFT is computed which gives the localized amplitudes of frequencies in that frame. The DFT for each frame *x(m)* are represented in a matrix known as the Short-Time Fourier Transform (STFT), expressed as the sum of a number DFTs with a duration in discrete samples *N* as (Ex. 5[37]). The spectrogram, the most common TFR, is expressed as (Ex. 6[37]) or as the absolute square of the STFT. It is commonly plotted as a heat map representing the amplitude of frequency ranges in the signal over time.

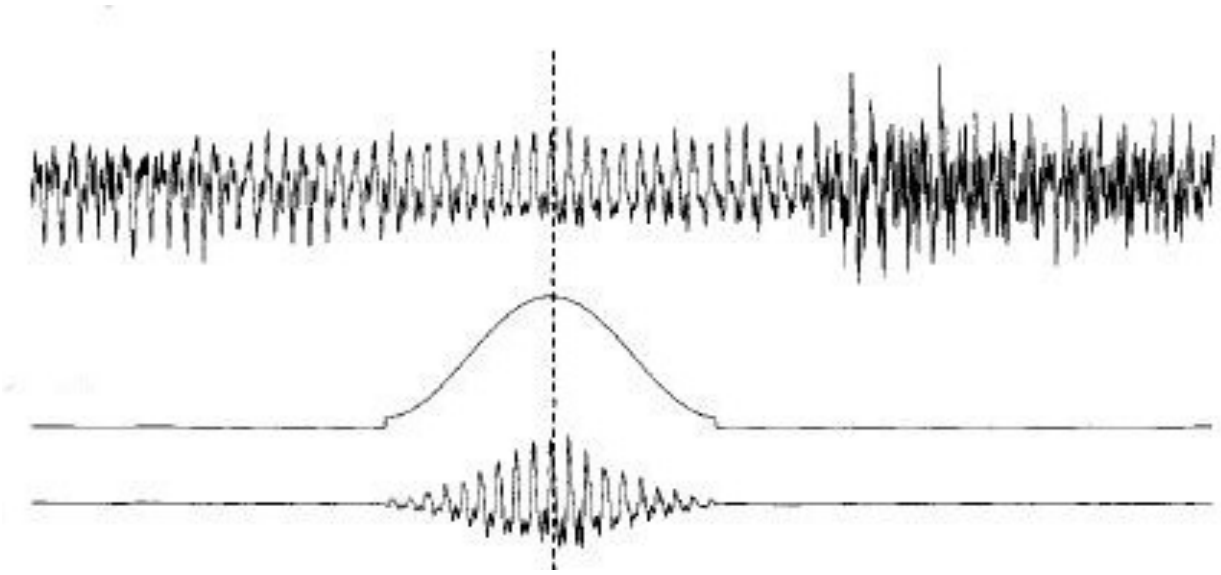

Figure 19. Top – Signal x(n). Center – Window function w(n-m). Bottom – Frame s(m). From [37]

$$s_n^w[m] = x[m]w[n-m]$$

Ex. 4

Ex. 5

$$\mathrm{STFT}_x^w[n,k] = \sum_{m=0}^{N-1} x[m]w[n-m]e^{\frac{-j2\pi km}{N}}$$

Ex. 6

$$\mathrm{SP}_x^w[n,k] = |\mathrm{STFT}_x^w[n,k]|^2$$

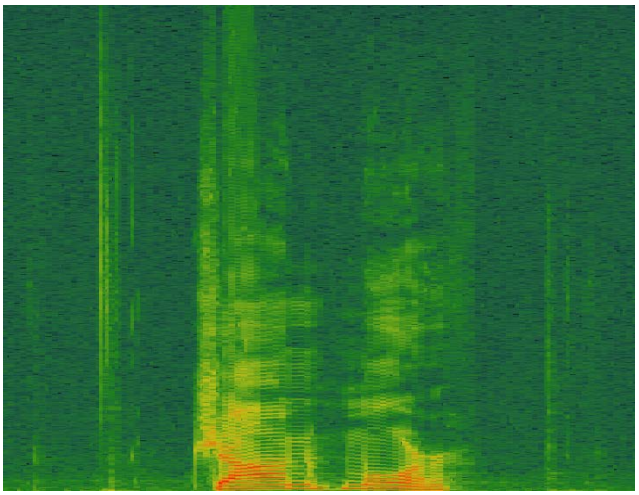

Figure 20. A sample spectrogram of the author pronouncing his own name (frequencies shown from 0 to 22K Hz).

The spectrogram is one of the most well-known TFRs for signal analysis generally, but it is not the most ideal for analysis in music. [38] explains that the linear frequency representation and constant resolution does not lend itself efficiently to the mapping of musical frequencies, which operate mainly within the range of comfortable human hearing as opposed to the full spectrum of sound. There are two popular approaches in MIR to compensating for the weaknesses of the standard spectrogram: the mel frequency cepstral coefficients and the Constant Q transform. The first is a filtering method that modifies the STFT using cepstral filtering according the mel scale, while the latter is an alternative computation to the STFT itself.

Mel Frequency Cepstral Coefficients

The *cepstrum* (an anagram of spectrum) of a signal is broadly defined as an inverse Fourier transform of the logarithm of the Fourier transform of a signal. As the Fourier

transform operates on a signal to produce a frequency-amplitude spectrum, the inverse Fourier transform operates on a frequency-amplitude spectrum to produce a signal cepstrum. Likewise, similar to the windows of discrete time used to simulate a continuous and dynamic Fourier transform over a signal (the STFT), to compute an Inverse Discrete Fourier Transform (IDFT) one must use windows of discrete frequency. The points on the frequency domain at which we define the center of these windows are known as the *cepstral coefficients.* [37] identifies two of the most common ways to establish these coefficients: linear prediction and mel frequency. Only mel frequency coefficients, however, are commonly used for signal analysis in music.

Mel frequency cepstral coefficients were first proposed by [39] and are defined according to the mel frequency scale. Whereas a standard musical scale describes notes according to real frequency, the mel scale is a concept in psychoacoustics that describes the perceptual distance between pitches to a human observer as a function of their frequency [40]. As frequency increases, the perceptual distance between them decreases exponentially. One can convert real frequency into mel frequency according to the law (Ex. 7[40]). The same law is used to define the mel frequency cepstral coefficients in such a way that the center points of each frequency window are equidistant on the mel scale, but exponential on the real frequency scale. The resulting resolution of each window is constant according to perceptual frequency but decreases according to real frequency. The number of coefficients, denoted K_{mel}, is a parameter that may be set at any integer, although $K_{mel} = 40$ is typical [37] [41]. The window surrounding each coefficient has a triangular shape such that frequencies at the center peak of the window filter are weighted most heavily (see fig. 21). The mel frequency cepstrum is computed for each

DFT of a frame of the original signal. The values in the cepstrum are represented as a vector of the dimensions K_{mel} for the given frame, in which the values of $k_{mel}(1)$ through $k_{mel}(K)$ are the sum of the amplitudes within each corresponding cepstral coefficient. The logarithm of the cepstrum is mapped back into the time domain with a Discrete Cosine Transform described by (Ex. 8[39]), producing an energy representation similar to the spectrogram but reduced in complexity from *real* frequency resolution to *perceptual* frequency resolution. This representation is referred to as an MFCC, as the cepstral coefficients are its unique contribution.

Ex. 7

$$m = 2595 \log_{10}\left[\frac{f}{700} + 1\right]$$

Ex. 8

$$DCT_x(i) = \sum_{n=1}^{T} x(n) \cos\left[\frac{\pi}{T} i \left(n - \frac{1}{2}\right)\right]$$

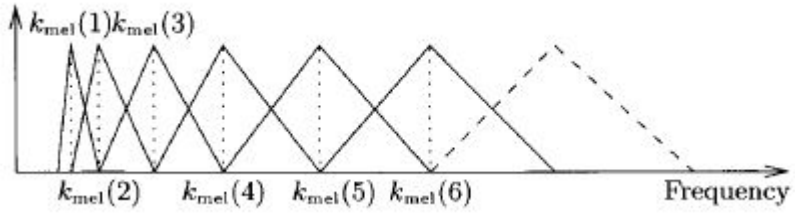

Figure 21. From [37]

The Constant-Q Transform

The Constant-Q transform is similar to the MFCC in that they both seek to generate a TFR that prioritizes frequency representations according to our perceptual understand of the frequencies. It differs, however, in that it is not a way to filter and modify an STFT; it

is instead an alternative computation to the STFT. It rests on the same basic principle as the MFCC that a logarithmic conceptualization of frequency is more musically meaningful than the linear conception of the STFT and thus the standard spectrogram. In the first proposal where the CQT is adapted for music, [38] explains that an additional benefit of a transform against logarithmic frequency is that it expresses the harmonic series of a frequency as an easily recognized and linearly consistent pattern. Given that the relative distance between frequencies in the overtone series above a fundamental frequency, or pitch, is constant, the constant Q transform seeks to preserve these patterns [42]. Because timbre is a function of the various amplitudes of the frequencies in the overtone series above the fundamental, the CQT is also well-suited to applications that want to identify the source of a frequency, like instrument identification tasks as well as more general timbre related functions.

The most essential aspect of the CQT, however, is behind its name. In order to ensure a frequency resolution sufficient for musical analysis, [38] proposed that the resolution be directly related to the frequency such that the ratio between them maintains

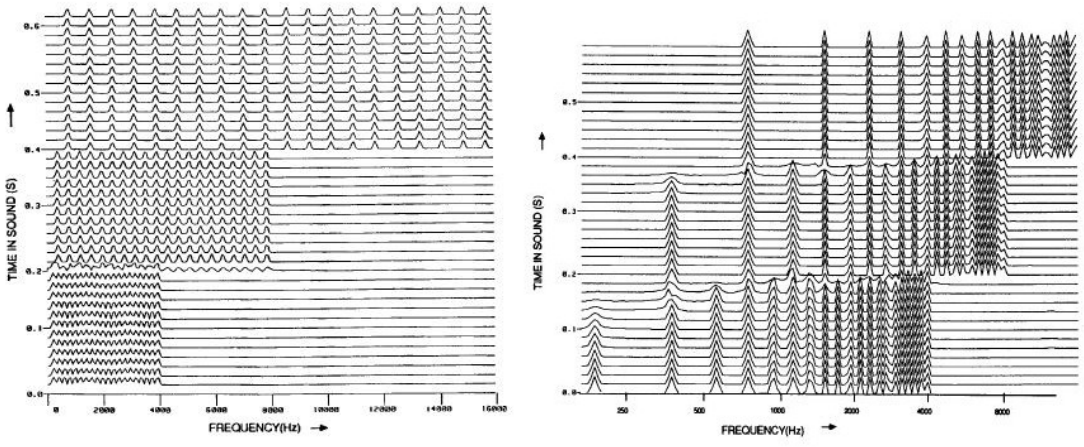

Figure 22.. Left - A DFT of 3 complex sounds with the fundamentals 196 Hz, 392 Hz, and 784 Hz, each having 20 harmonics of equal amplitude. Right – A CQT of the same sounds. One can see that the ratio relationships between the harmonics have been preserved linearly when expressed in the logarithmic frequency domain. From [38]

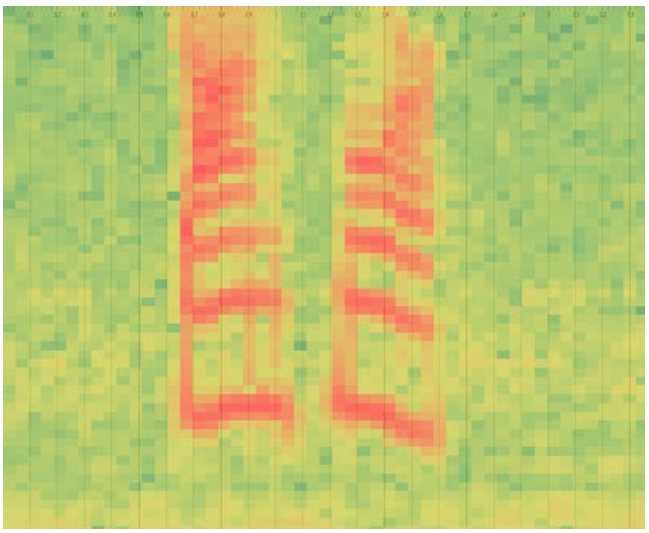

Figure 23. A sample CQT of the author pronouncing his own name. Notice the harmonics moving in parallel and the resolution decreasing as frequency increases.

a constant quality or *Q* defined as (Ex. 9[38]) where *f* is frequency and δf is resolution. Her value of Q was set to provide a resolution that could distinguish a quarter-step in the traditional western chromatic scale. Assuming equal-temperament, the change in frequency from one note to its quarter-step neighbor is always a change of 3%; therefore, the relationship must always resolve to at least $f/.029f$ or $Q = 34$. As explained previously, the frequency resolution of a frame varies as a function of the duration in time of that frame. The constant Q transform operates, then, by varying the duration of the frames inversely with frequency in order to maintain $Q = 34$. This requires that the duration in the frame in samples, denoted N[*k*], for a given frequency bin *k* contain at minimum *Q* periods of a given frequency in order to distinguish it from its nearest quarter-step neighbor. The constant Q proposed by [38] uses a Hamming window function of variable duration to determine the shape of each frame, defined in terms of the frequency spectral component *k* and the signal fragment *x*[*n*] as (Ex. 10[38]) with the given parameters. The CQT of the frame is computed using this variable window function, and can be expressed similarly to the previously defined DFT as (Ex. 11[38]). To distinguish between harmonics above 1568 Hz, Q is modified to resolve to 68. A sequence of CQTs for analyzed frames can then be mapped into the time domain to produce an energy representation akin to the spectrogram. The resultant representation

displays frequency amplitudes in the logarithm of the frequency domain and preserves patterns of harmonic distance regardless of fundamental frequency, as can be seen in fig 22.

Ex. 9 $$Q = \frac{f}{\delta f}$$

Ex. 10 $$w[k,n] = \alpha + (1-\alpha)\cos\left(\frac{2\pi n}{N[k]}\right), \alpha = \frac{25}{46}, 0 \leq n \leq N[k]-1$$

Ex. 11 $$CQT_x[k,n] = \frac{1}{N[k]}\sum_{m=0}^{N[k]-1} w[k,n]x[n]^{\frac{-j2\pi Qn}{N[k]}}$$

APPENDIX 5. Chord Sequence Estimation

One can understand a chord as a functional harmonic relationship between multiple pitches with some intervals of separation, so to identify a chord requires that one know two things: the combination of pitches being played and the functional relationship implied by that combination. For this, a system must be capable of discerning pitches out of a signal. A pitch is defined in terms of the amplitudes of the frequencies being sounded in a signal. Identifying a pitch depends on a process known as fundamental frequency, of F0, estimation. When a listener perceives a pitch, it is the F0 that defines the "note" that the pitch is sounding [43]. Logically then, one can estimate pitches and the chords they create based on the TFRs previously outlined, and indeed [44] has demonstrated a method of doing so using the CQT. As shown, a strength of the CQT is that it preserves the harmonic relationships between frequencies as an easily recognizable pattern. The harmonic relationships between frequencies that belong to the

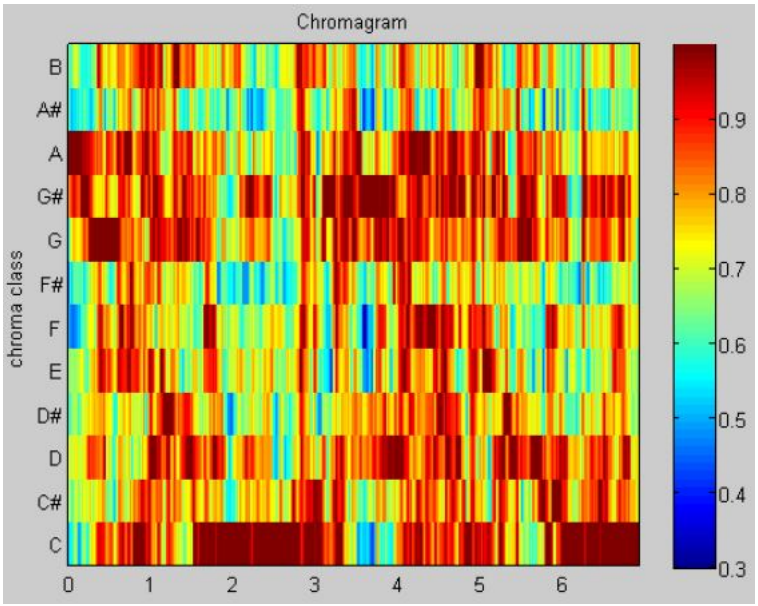

Figure 24. The chroma features of an audio sample. The frequency features are filtered by pitch class, and greater value in the heat map represents sum amplitude.

same pitch are defined according to the overtone series [42], and as pitch changes the ratio between these frequencies does not change. This means that a fundamental frequency can be defined as a frequency over which the overtone series of frequencies can be identified. In situations where only one pitch is sounding, this is a simple procedure; however, when applied to chord identification where multiple pitches are sounding at once, interference between frequencies can cause strange behavior in the frequency-amplitude spectrum. [44] demonstrates that a component that should appear on the frequency-amplitude spectrum of the CQT at the frequency f_1 will not appear if another component exists on the spectrum at the frequency f_2 according to the relationship (Ex. 12[44]). In practical terms, out of the 57 chords in the Western tradition that can be defined as a combination of up to 4 pitches, 14 of these chords consist of a combination of pitches that includes two frequencies with this relationship. This makes these chords impossible to identify with complete accuracy even given a completely accurate TFR.

Ex. 12

$$\frac{Q}{Q+1} f_2 \leq f_1 \leq \frac{Q}{Q+1} f_2$$

While [44] explores how one might compensate for this effect, most contemporary research is built upon an alternative representation that is designed to provide an estimation of likely pitches over a signal in combination with a pattern recognition model that can take these estimations as input and return the likely chords as output. [45] identifies three components common to chord identification systems under this paradigm: chroma feature extraction, filtering, and pattern matching. Chroma feature extraction refers to the generation of a type of this specialized TFR known as a pitch class

profile or chromagram. This TFR was first proposed for in [46] and is comparable in some ways to the MFCC. In both cases, the frequency-amplitude values determined by each frame of a TFR of a signal are passed through specialized filter bins, reducing the complexity of the representation to signify frequency features organized by some concept of relevance. In the case of the MFCC, these filter bins sit along the frequency spectrum with their distances determined by mel scale. It thus organizes frequency-amplitude by perceptual frequency distance. The pitch class profile organizes frequencies with another kind of filter, that of pitch. Windows in the frequency domain divide frequencies by the pitches that a fundamental in that range would signify. For instance, the range of frequencies around 440 Hz are filtered into the pitch class "A." These filters are octave-agnostic, meaning that all frequencies at would resolve to an A of any octave are filtered into the same pitch class of "A." The sum of the amplitudes of the frequencies that fall within each pitch class filter in each frame of the STFT determine the value for that pitch class. Given that there are 12 pitch classes in the Western classical tradition (one for each note on the chromatic scale of an octave), the values for each frame of the STFT are thus reduced from a full frequency representation to a vector of 12 dimensions, for which the values represent the sum of amplitudes in that respective pitch class. In other words, the pitch class profile approximates the relative strength of each pitch at every frame of the signal, regardless of octave. Alternative PCP's could be devised using vectors or greater or fewer dimensions corresponding to the organization of pitches used in other musical traditions. This TFR is considered foundational in the field of chord-sequence based retrieval [47]—[50]. Like with other TFRs, the duration of the frame has significant effects on the resolution of the represented frequency-amplitudes. This is what makes the

filtering step necessary before pattern matching can begin[45]. On one hand, the frame must be of a short enough to fit within the expected rate of chord change in order to capture that change. On the other hand, the shorter the frame, the more susceptible the frequency-amplitude representation is to noise. Commonly, researchers will pass a chromagram with frames of short duration through a low-pass filter that minimizes frequency-amplitude values that do not persist over a significant number of frames [46][48][51].

The next step is to take these pitch estimations and formulate some function that can use them to identify chord structures. The method originally proposed in [46] took the form of a simple nearest-neighbor calculation in the vector space defined by the pitch classes; however, [51] notes that this method has only been found to work well in cases of synthetic sound and not in real, often more chaotic polyphonic recordings. The first method to find success with live sound was proposed in [50] using hidden Markov models (HMMs) trained with an Expectation-Maximization (EM) algorithm. HMMs are a machine-learning algorithm in which the state of a set of data out of some finite set of states is predicted based on observations in that data. The use of HMMs is common in the signal analysis for speech signals, in which the state determined is a phoneme being pronounced. In [50], a vocabulary of chords forms the finite set of states. The probability of each state is defined as a single Gaussian distribution in N-dimensional space where N matches the dimensions of the pitch class vector. The probability is then adjusted (trained) based on the performance according to an EM algorithm defined as (ex. 13[50]) where E is the estimation of the chord in terms of the probability P given the observed features X and unknown chord labels Q according the probability parameters Θ. The

estimation is determined as a function of the probability of both the current parameters and previous parameters such that the value of log$P(X,Q|\Theta)$ is maximized as the sum of estimated labels increases. The original Gaussian distribution model is set initially at random parameters and is then tuned by this E-M training process. As [50] notes, the original model parameters could be estimated directly only if the delineation between states (the boundaries between chords) was known beforehand. [48] attempted to approximate these boundaries by introducing high-level rhythmic information into the pitch class representation with some success, outperforming the original work done by [50]. Once a sequence of chord states is determined, sequence alignment can allow for quantization of similarity between chord sequences. This method has achieved marked success in cover-song and contrafact identification, for which instrumentation, timbre, duration, tempo, key, and potentially other qualities like melody are expected to vary, but harmonic progression is likely to remain largely the same [47].

Ex. 13

$$\mathrm{E}[\log \mathrm{P}(\mathrm{X}, \mathrm{Q}|\Theta)] = \sum_{\mathrm{Q}} \mathrm{P}(\mathrm{Q}|\mathrm{x}, \Theta_{\mathrm{old}}) \log(\mathrm{P}(\mathrm{X}|\mathrm{Q}, \Theta)\mathrm{P}(\mathrm{Q}|\Theta))$$