# GROUPING RECORDED MUSIC BY STRUCTURAL SIMILARITY

**Juan Pablo Bello**
Music and Audio Research Lab (MARL), New York University
jpbello@nyu.edu

## ABSTRACT

This paper introduces a method for the organization of recorded music according to structural similarity. It uses the Normalized Compression Distance (NCD) to measure the pairwise similarity between songs, represented using beat-synchronous self-similarity matrices. The approach is evaluated on its ability to cluster a collection into groups of performances of the same musical work. Tests are aimed at finding the combination of system parameters that improve clustering, and at highlighting the benefits and shortcomings of the proposed method. Results show that structural similarities can be well characterized by this approach, given consistency in beat tracking and overall song structure.

## 1. INTRODUCTION

Characterizing the temporal structure of music has been one of the main goals of the MIR community, with example applications including thumbnailing, long-term segmentation and synchronization between multiple recordings [1, 2]. Despite this focus, however, there has been little in terms of using structure as the main driver of audio-based retrieval and organization engines.

This paper proposes and evaluates a methodology for the characterization of structural similarity between musical recordings. The approach models similarity in terms of the information distance between music signals represented using self-similarity matrices. These matrices are well-known for their ability to characterize recurring patterns in structured data, and are thus widely used in MIR for the analysis of musical form. However, in retrieval applications they are mostly used as intermediate representations from which a final representation (e.g. beat spectrum, segment labels) is derived. In this paper we argue that self-similarity matrices can be used directly in the computational modeling of texture-, tempo- and key-invariant relationships between songs in a collection. Our approach is mainly inspired by the work in [3], which uses the same principle to compare the structure of protein sequences.



### 1.1 Background

The use of structure for audio-based MIR was first proposed in [4]. This approach is based on the idea that long-term structure can be characterized by patterns of dynamic variation in the signal. In this approach, song similarity is measured as the cost of DP-based pairwise alignment between sequences of local energy or magnitude spectral coefficients. Experimental results, albeit preliminary, show the potential of this idea for retrieval.

A similar concept is explored in [5], and more extensively in [6], where variations of spectral content are quantized into a symbolic sequence, obtained via vector quantization or HMMs. In these works, pairwise song similarity is measured using the edit distance or, more efficiently, locality sensitive hashing [6].

The mentioned sequences are not only able to represent the texture and harmony of musical pieces, but also structural patterns, from motifs and phrases to global form. Musical sequences sharing style, origin or functionality will be likely to show structural similarity, despite differences in actual sequence content. Hence, a change of key does not preclude listeners from identifying a 12-bar blues, and the relationship between different variations and renditions of a work remain close, despite changes of instrumentation, ornamentation, tempo, dynamics and recording conditions. Unfortunately, all representations discussed above are sensitive to one or more of these variables. As a result, their success at characterizing music similarity depends on their ability to marginalize those changes. Examples include the use of modified distance metrics and suboptimal feature transposition methods [2, 5].

Structure comparison has been extensively studied in other fields, such as bioinformatics. For protein sequences, for example, structures are usually characterized using *contact maps*, which are, simply put, binary self-similarity matrices where a 1 characterizes a contact (i.e. similarity higher than a certain threshold) and a 0 the lack of it. The problem of comparing protein topologies using contact maps is known as *maximum contact map overlap*, with many proposed solutions in the literature. In this paper we concentrate on the one proposed in [3], which uses an approximation of the information distance between two contact maps known as the *normalized compression distance* (NCD), to be discussed in more detail in section 2.2.

In music, the NCD has been used on raw MIDI data for clustering and classification based on genre, style and melody [7, 8]. More recently, it has been used on audio data for sound and music classification [9] and, with lim-

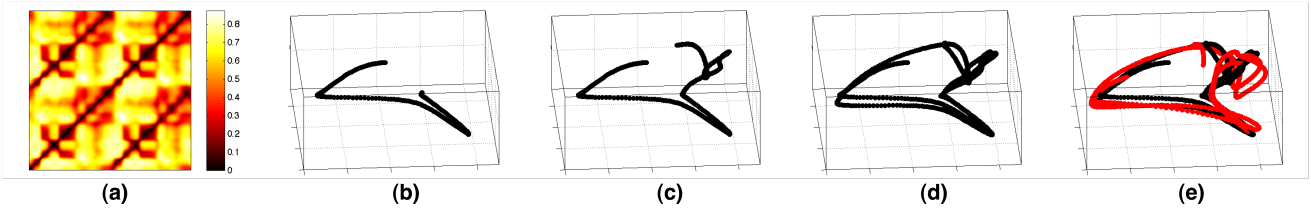


**Figure 1**. (a) Self similarity matrix of the first 248 bars of a performance of Beethoven $5^{th}$ Symphony; MDS projection of a quarter (b), half (c) and full matrix (d) to 3 dimensions; (e) comparison of two different performances.

ited success, in cover-song identification [10]. To the best of our knowledge this paper proposes the first use of NCD to characterize structural similarity between music audio recordings.

### 1.2 Example

Figure 1(a) shows a self-similarity matrix of the first 248 bars of the first movement of Beethoven's $5^{th}$ symphony. The recording is of a 2006 performance by the Russian National Orchestra conducted by Mikhail Pletnev. Figures 1(b-d) are the result of taking the distances in the matrix and projecting them into a 3-dimensional space using classical multidimensional scaling (MDS). The figures show the trajectory of the piece at a quarter, half and full segment length, respectively. Figures 1(b) and (c) depict the famous opening section of this symphony as a loop, while figure 1(d) shows the recapitulation as simply another, approximate instance of the same loop. This example clearly shows how self-similarity matrices are able to characterize primary (the trajectory itself) and, at least, secondary (local motifs such as the loop) structure in music. Figure 1(e) shows the full segment trajectory described above (in black), and a new trajectory, corresponding to a 1963 recording by the Berlin Philharmonic conducted by Herbert von Karajan (in red). The goal of our approach is to quantify the (dis)similarity of these representations, and to use the results to group related music together.

## 2. APPROACH

The proposed approach consists of three main parts: (a) representation, where a self-similarity matrix is generated from the analysis of the audio signal; (b) similarity, where the pairwise distance between the representations is computed using the NCD; and (c) clustering, where the matrix of NCDs is used for the grouping of songs. The details are explained in the following.

### 2.1 Representation

In our implementation we use a beat-synchronous feature set $F$, composed of either MFCC or chroma features. The first 20 MFCCs are calculated using a 36-band filterbank, frame size of 23.22ms and 50% overlap. The chroma features are computed via the constant-Q transform using a minimum frequency of 73.42 Hz, 36 bins per octave and a 3-octave span, on a signal downsampled to $f_s = 5512.5$ Hz. The resulting features are tuned and their dimensionality reduced to 12 with a weighted sum across each 3-bin pitch class neighborhood. For beat tracking we use the algorithm in [11], and average the extracted features between consecutive beats. Beat tracking is used to reduce the size of the self-similarity matrix and to minimize the effect of tempo-variations on the representation.

The feature set is smoothed using zero-phase forward-backward filtering with a second order Butterworth filter. Filter cutoff is at $1/128^{th}$ of the feature rate. Finally, the features are standardized (separately for each song).

The computation of self-similarity matrices has been discussed extensively elsewhere in the literature and will not be discussed in any detail here. Suffices to say that for our tests we use both the euclidean and cosine distances. Once computed, matrices are normalized (per song) to the [-1,1] range, their upper triangular part extracted, and the values uniformly quantized and encoded into $B$ bits. In our experiments $B$ assumes the values 2, 3 and 4. It is worth noting that we have favored the notion of "fuzzy" rather than binary self-similarity, as it is not clear what an adequate definition of *contact* may be in the context of this work. For the same reason we have favored the use of uniform quantization over other possible partitions of the similarity range.

### 2.2 Similarity

We measure similarity using the normalized compression distance (NCD), which will be briefly introduced here (For a comprehensive discussion the reader is referred to [7]).

It can be shown that the information distance between two objects $o_1$ and $o_2$, up to a logarithmic additive term, is equivalent to:

$$ID(o_1, o_2) = max\{K(o_1|o_2), K(o_2|o_1)\} \qquad (1)$$

where $K(.)$ denotes the Kolmogorov complexity. The conditional complexity $K(o_1|o_2)$ measures the resources needed by a universal machine to specify $o_1$ given $o_2$.

The information distance in Eq. 1 suffers from not considering the size of the input objects, and from the non-computability of $K(.)$. To solve the first problem, a normalized information distance can be defined as:

$$NID(o_1, o_2) = \frac{max\{K(o_1|o_2), K(o_2|o_1)\}}{max\{K(o_1), K(o_2)\}} \qquad (2)$$

To solve the second problem, we can approximate $K(.)$ using $C(.)$, the size in bytes of an object when compressed

using a standard compression algorithm. Using this principle, it can be shown that equation 2 can be approximated by the normalized compression distance:

$$NCD(o_1, o_2) = \frac{C(o_1 o_2) - min\{C(o_1), C(o_2)\}}{max\{C(o_1), C(o_2)\}} \quad (3)$$

where $C(o_1 o_2)$ is obtained by compressing the concatenation of objects $o_1$ and $o_2$ [7]. For our implementation the objects are the encoded self-similarity matrices for each song. We use the NCD implementation in the *CompLearn* toolkit [1] with the bzip2 and PPMd compression algorithms.

### 2.3 Clustering

We use an algorithm from Matlab's statistics toolbox that builds a hierarchical cluster tree using the complete linkage method [12]. The clusters are defined by finding the smallest height in the tree at which a cut across all branches will leave $MaxClust$ or less clusters. The output of the process is a vector containing the cluster number per item in the test set.

## 3. EXPERIMENTAL SET-UP

### 3.1 Test Data

We use two datasets in our experiments. The first set, which we call P56, consists of 56 recordings of piano music, including excerpts of 8 works by 3 composers (Beethoven, Chopin and Mozart), played by 25 famous pianists between 1946 and 1998. It was collected as part of the computational study of expressive music performance discussed in [13]. Each work has, at least, 3 associated renditions and at most 13, with audio file lengths in the range of 1 to 8 minutes.

The second set (S67, collected by the authors) includes 67 recordings of symphonic music, including one movement for each of 11 works by 7 composers (Beethoven, Berlioz, Brahms, Mahler, Mendelssohn, Mozart and Tchaikovsky). The set includes instances from 56 different recording sessions scattered between 1948 and 2008, featuring 34 conductors. Each work has 6 associated renditions, with the sole exception of the 3rd movement of Brahm's Symphony No. 1 in C minor, for which 7 performances are available. The duration of the recorded movements range from 3 to 10 minutes.

Classical music is used as, apart from the odd repetition of a motif or section, the structure of renditions can be expected to be the same. The two sets are composed of recordings using similar instrumentation (piano, orchestra), to emphasize the difference with timbre-base similarity approaches. Both sets, however, present significant variations in recording condition and interpretation (notably in dynamics and tempo). All files are 128 kb/s MP3s with sampling frequency of 44.1kHz.

[1] http://www.complearn.org

### 3.2 Methodology

Clustering methods are highly sensitive to both the number and relative size of partitions in a dataset. To account for variations of those factors and avoid overfitting, every test is performed $I$ times, each using a random sample of size $N < M$, where $M$ is the number of items in the dataset. For every test, we report the mean accuracy of clusters across the $I$ subsets, measured as follows.

Given a partition of the dataset into $R$ groups, $Q = \{q_1, ..., q_R\}$, produced by the clustering algorithm, and a target partition, $T = \{t_1, ..., t_P\}$, we can validate $Q$ using the Hubert-Arabie Adjusted Rand (AR) index as:

$$AR = \frac{\binom{N}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{N}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]} \quad (4)$$

where $\binom{N}{2}$ is the total number of object pairs in our dataset. AR measures the correspondence between $Q$ and $T$, as a function of the number of the following types of pairs: $(a)$ pairs with objects in the same group both in $Q$ and $T$; $(b)$ objects in the same group in $Q$ but not in $T$; $(c)$ objects in the same group in $T$ but not in $Q$; and $(d)$ objects in different groups in both $Q$ and $T$. The AR index accounts for chance assignments and does not require arbitrary assignment of cluster labels not $P = R$, as might be the case when using classification accuracy to validate clustering. Readers unfamiliar with the AR index might find the following guidelines useful: AR = 1 means perfect clustering, while values above 0.9, 0.8 and 0.65 reflect, respectively, excellent, good and moderate cluster recovery. Random partitions of the dataset result on AR $\rightarrow$ 0 (can also assume small negative values). For a detailed discussion of the properties and benefits of the AR index see [14].

## 4. RESULTS AND DISCUSSIONS

The main goal of our experiments is to test the capacity of the proposed approach in characterizing structural similarity. As similarity is an elusive concept which is not easily quantified, we test an approximate scenario: the task of clustering a music collection into groups of renditions of the same work. Thus, for example, a partition $Q$ of S67, generated using the approach in section 2 with parameters $\theta$, is validated using AR and a target partition $T$ of 11 groups, where each group contains the 6 or 7 renditions of one of the works in the collection.

Specifically, our experiments seek to: (1) find the parameterization $\theta$ that maximizes AR, (2) assess the impact of the used clustering methodology, and (3) highlight the strengths and shortcomings of our approach.

### 4.1 Parameterization

In our experiments $\theta = \{F, d, B, C, MaxClust\}$, where $F$ is the feature set (MFCC or chroma), $d$ the distance metric used to compute the self-similarity matrix (euclidean or cosine), $B$ the number of bits used to quantize the matrix (2, 3 or 4), $C$ the compression method used for the computation of the NCD (bzip2 or PPMd), and $MaxClust$ the

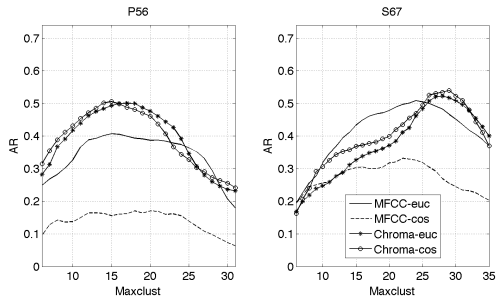


**Figure 2**. Comparison of mean AR results for all $F, d$ combinations on sets P56 (left) and S67 (right).

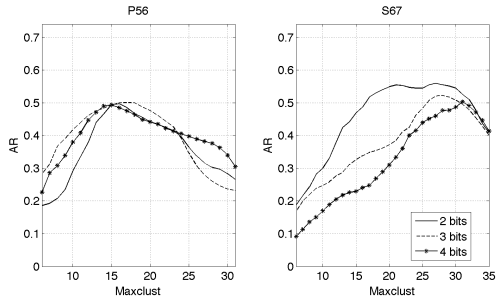


**Figure 3**. Comparison of mean AR results for $B = \{2, 3, 4\}$ on sets P56 (left) and S67 (right).

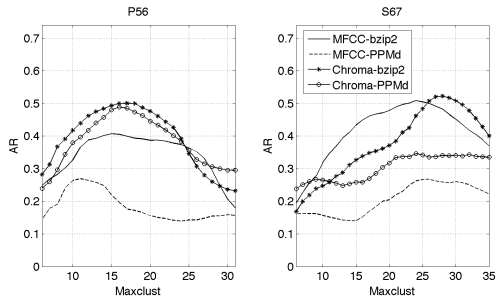


**Figure 4**. Comparison of mean AR results for $C = \{bzip2, PPMd\}$ on sets P56 (left) and S67 (right).

maximum number of clusters to be retrieved from the tree (between 6 and 35).

All possible combinations of $\theta$ are tested $I = 50$ times [2], using random samples of size $N = 0.75 \times M$ (42 for P56, 50 for S67). In all tests, both collections are tested independently.

Figure 2 shows results for all $F, d$ combinations for $C = bzip2$ and $B = 3$. As with most figures in this section, it separately shows AR values for P56 (left) and S67 (right), across the range of $MaxClust$ values. For both datasets, chroma features outperform MFCCs, clearly for P56 and slightly for S67. This is consistent with the notion of harmonic content as a reliable indicator of structure in music, as has been repeatedly found in the segmentation literature [1, 2]. The better performance of MFCCs in S67 compared to P56 is to be expected, as within-song timbre differences and dynamic changes are more pronounced in orchestral than in piano music. For chroma features, the use of euclidean or cosine distances in the computation of the self-similarity matrix makes little difference. For MFCCs, however, the euclidean distance results in significantly better performance, indicating that dynamics are as important as timbre changes in defining the structure of a piece.

Figure 3 illustrates the importance of the number of bits $B$ used in the encoding and quantization of the self-similarity matrix, for $F = chroma$, $d = euclidean$ and $C = bzip2$. Apart from $B = 2$ giving the best results for S67, no clear trend is visible in these plots (at least not common to both sets). This hints at process independence from the choice of $B$. The good performance of $B = 2$, however, opens the door for a binary definition of contacts in music, although more extensive testing is necessary to define an appropriate threshold.

Finally, figure 4 compares two compression methods for the computation of NCD. In these plots, $F = chroma$, $d = euclidean$ and $B = 3$. In all cases $bzip2$ outperforms $PPMd$, which is unfortunate as the latter is much faster than the former. This result seems to contradict findings in the literature where the PPM family of compression methods usually works best for the NCD computation [7].

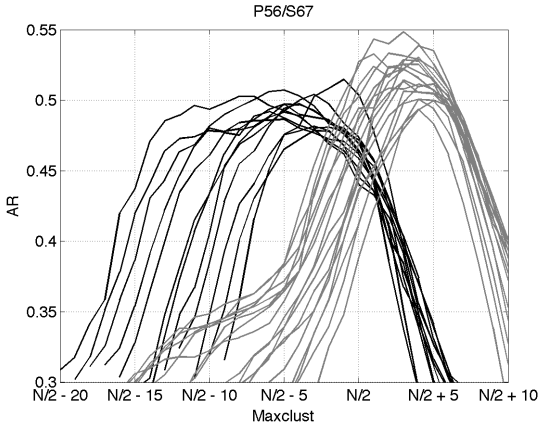


**Figure 5**. Variation of mean AR according to random sample size $N$ (P56 in black, S67 in gray).

### 4.2 Clustering methods

On a separate experiment, we tested our system against variations of the random sample size $N$ for both collections. $N$ values ranged from 30 to 52 for P56, and 64 for S67. We used $F = chroma$, $d = euclidean$, $B = 3$ and $C = bzip2$. Figure 5 shows results for P56 (in black) and S64 (in gray, skewed towards the right), across a range of $MaxClust$ values ranging from $N/2 - 20$ to $N/2 + 10$. Each curve corresponds to a value of $N$. Variations of peak AR across $N$ appear to be uniformly distributed in the depicted range for each test set. Their location within this range does not follow any obvious trend. For example, for P56, the minimum peak corresponds to the $N = 32$ curve, while the maximum peak is for $N = 30$ (closely followed

[2] We tested $I = \{10, 20, 50, 100, 200, 500, 1000\}$ and found variations of mean AR to be minimal for $I \geq 50$.

by $N = 48$). All other peaks are randomly located in between.

Notably, the location of peaks appears to be a function of $N$, with most peaks in $(N/2 - 5) \pm 3$ for P56 and in $(N/2 + 3) \pm 2$ for S67. The difference between the sets, however, also indicates that the size of the collection $M$, the number of groups within that collection and the size of those groups have a hand in the results. While $N$ and $M$ are always known, it is unreasonable to expect the number and size of groups to be known, making the choice of value for the critical $MaxClust$ parameter a complex one. Our inability to define $MaxClust$ with prior information is a major shortcoming of the proposed approach.

As an alternative we have tested a different clustering algorithm, which operates by merging clusters whose separation, measured in their connecting node, is less than a pre-specified *Cutoff* value, ranging between 0 and 1. Notably, this method does not require any prior information about cluster numbers. Additionally, we test building the hierarchical cluster tree using single, average and weighted linkage in addition to the complete linkage method used in the rest of this paper [12]. Figure 6 shows the results of these tests using $F = chroma$, $d = euclidean$, $B = 2$ and $C = bzip2$. The AR = 0.63 result for weighted linkage and Cutoff = 0.85 in S67 is the highest obtained in our experiments, a significant increase on our previous best (visible in the "complete" curve of the same graph). It clearly shows that gains can be made by improving our clustering stage. However, this result is not indicative of a general trend, as illustrated by the low results obtained for the same method in the P56 dataset. An in-depth exploration of the space of clustering methods and their parameterizations will be the focus of future work.

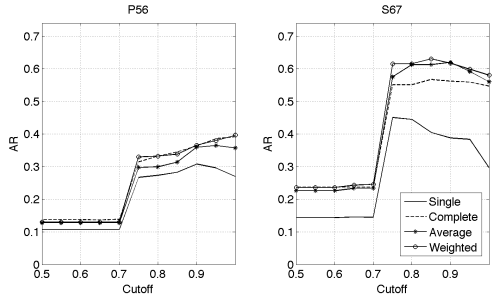


**Figure 6**. Test of cutoff clustering with 4 linkage methods.

### 4.3 An example tree

Figure 7 is generated using yet another linkage algorithm on the full S67 dataset, the quartet method described in [7], using $F = chroma$, $d = euclidean$, $B = 3$ and $C = bzip2$. Clustering on this tree using $MaxClust = 36$ results on $AR = 0.55$, which makes this graph representative of system performance using the best parameterization.

The tree branches out into 10 clusters, each corresponding to a work in the collection. Four of those clusters group all renditions of a given work. Figure 7(a) shows a detail of the tree exemplifying one such cluster, corresponding to the 7 renditions of the third movement of Brahm's Symphony No. 1 in C minor. Two clusters group 5 out of 6 performances, for example those for the third movement of Mozart's Symphony No. 4 in G minor k550 depicted in Figure 7(c). One cluster, for the second movement of Mahler's Symphony No. 1 in D major "Titan", groups 4 out of 6 performances as shown in Figure 7(b). The three remaining clusters group only 3 or 2 performances out of 6. Only one work results in no clusters of any kind. In total, 47 out of 67 recordings are correctly assigned to a group. Ungrouped recordings are located in the stem of the tree, which has been gray-shaded in the graph.

Figures 7(b) and (c) also help illustrate the effect of beat tracking accuracy on the proposed approach. The number of detected beats in the missing performance of Mozart's k550, visible in the stem of the tree in Fig. 7(b), is approximately twice as many as those detected in all other performances of the same piece. Octave errors act as filters on the feature set, which can result on a significant loss of detail in the corresponding self-similarity matrix and, as the tree shows, a poor characterization of structural similarity between the recordings. This is an important drawback of our approach as octave errors are common in beat tracking. Another example of the same problem are the two missing recordings in Mahler's Symphony 1 cluster in Fig. 7(b), which are located in the lower end of the stem of the tree. An informal analysis of the results shows that a good portion of overall clustering errors are associated to inconsistencies in beat tracking. It is worth noting that "inconsistency" is the right word in this case, as what is really important is not that beats are correctly tracked, but that their relation to the actual tempo of the piece is the same for all performances.

An additional observation relates to the six performances of the fourth movement of Berlioz's "Symphonie Fantastique". The score includes a repetition of the first 77 bars of this movement before entering its second half, roughly describing an AAB structure. Half of the performances in our dataset, however, ignore that repetition resulting on a shorter AB structure. Correspondingly, the cluster in the tree related to this piece groups only the latter, while the other three performances appear close together in the lower end of the tree. While in theory the common part of the structure should be enough to identify the similarity between all six recordings, in practice this is clearly not the case. This sensitivity to common structural changes, e.g. repetitions, raises questions about the potential use of NCD-based similarity in the modeling of the relationships that exist amongst variations, covers, remixes and other derivatives of a given work. Further research is now being conducted to fully explore this issue.

## 5. CONCLUSIONS AND FUTURE WORK

This paper presents a novel approach for the organization of recorded music according to structural similarity. It uses the Normalized Compression Distance (NCD) on self-similarity matrices extracted from audio signals, using stan-

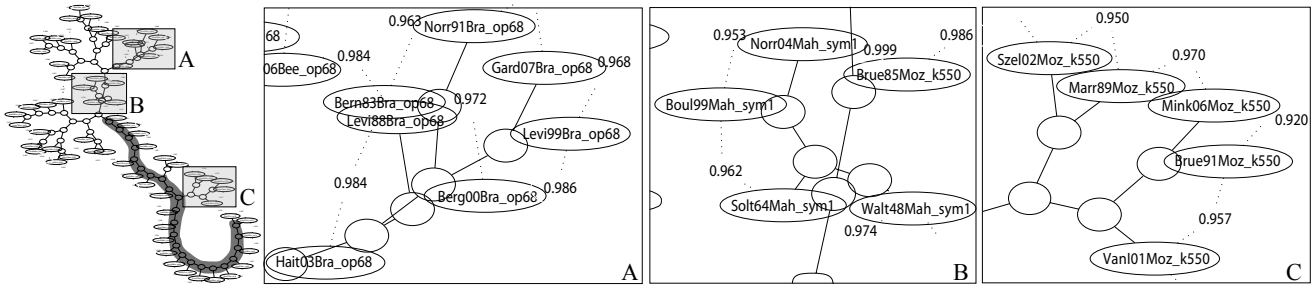


**Figure 7**. Uprooted binary tree of S67 using the quartet method. Details show a perfect cluster (A) and two partial clusters (B and C)

dard features and distance metrics. The approach is evaluated on its ability to facilitate the clustering of different performances of the same piece together. Experimental results on piano and orchestral music datasets show that the approach is able to successfully group the majority of performances in a collection, resulting on average AR values in the 0.5-0.6 range. Our tests show that best results are obtained for self-similarity matrices computed using chroma features and the euclidean distance, and encoded using 2-3 bits. They also show that the NCD works best when using the $bzip2$ compression algorithm. Preliminary results also indicate that further gains can be made by improving the clustering stage.

On the downside, the approach has shown sensitivity to octave errors in beat tracking and, predictably, to structural changes, which limit the potential application of the current implementation to the retrieval and organization of other types of musical variations. To address these issues, future work will concentrate on two main areas. First, the improvement of the self-similarity representation, along the lines of work in [2], to include transposition invariance, path following and the merging of matrices computed at 1/2, 1 and 2 times the tracked tempo. Second, we will explore alternatives to the use of NCD for the maximum contact map overlap problem. We plan to explore solutions based on the branch and cut approach (e.g. [15]) and adapt them to the specificities of music data.

## 6. ACKNOWLEDGEMENTS

The author would like to thank Gerhard Widmer and Werner Goebl for the P56 dataset, and Dan Ellis and the CompLearn team for free distribution of their code libraries. This material is based upon work supported by the NSF (grant IIS-0844654) and by the IMLS (grant LG-06-08-0073-08).

## 7. REFERENCES

[1] M. A. Bartsch and G. H. Wakefield. To catch a chorus: Using chroma-based representations for audio thumbnailing. In *WASPAA-01, NY, USA*, pages 15–18, 2001.

[2] M. Müller. *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[3] N. Krasnogor and D. A. Pelta. Measuring the similarity of protein structures by means of the universal similarity metric. *Bioinformatics*, 20(7):1015–1021, 2004.

[4] J. Foote. Arthur: Retrieving orchestral music by long-term structure. In *ISMIR*, 2000.

[5] J.-J. Aucouturier and M. Sandler. Using long-term structure to retrieve music: Representation and matching. In *ISMIR 2001, Bloomington, Indiana, USA*, 2001.

[6] M. Casey and M. Slaney. Song intersection by approximate nearest neighbour retrieval. In *ISMIR-06, Victoria, Canada*, 2006.

[7] R. Cilibrasi and P. M. B. Vitányi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.

[8] Ming Li and Ronan Sleep. Genre classification via an LZ78 string kernel. In *ISMIR-05, London, UK,*, 2005.

[9] M. Helén and T. Virtanen. A similarity measure for audio query by example based on perceptual coding and compression. In *DAFx-07, Bordeaux, France*, 2007.

[10] T. Ahonen and K. Lemström. Identifying cover songs using normalized compression distance. In *MML'08, Helsinki, Finland*, 2008.

[11] D. Ellis. Beat Tracking by Dynamic Programming. *Journal of New Music Research*, 36(1):51–60, March 2007.

[12] R. Xu and D. Wunsch. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.

[13] G. Widmer, S. Dixon, W. Goebl, E. Pampalk, and A. Tobudic. In search of the Horowitz factor. *AI Mag.*, 24(3):111–130, 2003.

[14] D. Steinley. Properties of the Hubert-Arabie adjusted Rand index. *Psychological methods*, 9(3):386–396, September 2004.

[15] W. Xie. and N. V. Sahinidis. A branch-and-reduce algorithm for the contact map overlap problem. *Research in Computational Biology (RECOMB 2006), Lecture Notes in Bioinformatics*, 3909:516–529, 2006.