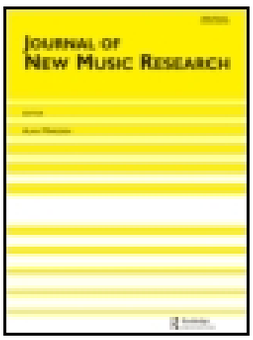

Journal of New Music Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/nnmr20

An investigation of music analysis by the application of grammar-based compressors

David Humphreys, Kirill Sidorov, Andrew Jones & David Marshall

To cite this article: David Humphreys, Kirill Sidorov, Andrew Jones & David Marshall (2021): An investigation of music analysis by the application of grammar-based compressors, Journal of New Music Research, DOI: 10.1080/09298215.2021.1978505

To link to this article: https://doi.org/10.1080/09298215.2021.1978505

Published online: 24 Sep 2021.

JOURNAL OF NEW MUSIC RESEARCH
https://doi.org/10.1080/09298215.2021.1978505

Routledge
Taylor & Francis Group

An investigation of music analysis by the application of grammar-based compressors

David Humphreys, Kirill Sidorov, Andrew Jones and David Marshall

School of Computer Science and Informatics, Cardiff University, Cardiff, UK

ABSTRACT

Many studies have presented computational models of musical structure, as an important aspect of musicological analysis. However, the use of grammar-based compressors to automatically recover such information is a relatively new and promising technique. We investigate their performance extensively using a collection of nearly 8000 scores, on tasks including error detection, classification, and segmentation, and compare this with a range of more traditional compressors. Further, we detail a novel method for locating transcription errors based on grammar compression. Despite its lack of domain knowledge, we conclude that grammar-based compression offers competitive performance when solving a variety of musicological tasks.

ARTICLE HISTORY
Received 22 January 2020
Accepted 1 September 2021

KEYWORDS
Grammar-based compression; music analysis; classification

1. Introduction

The analysis of structure within a musical piece is an important approach to music analysis, and traditionally achieved through the context of music theory. Schoenberg et al. (1967) asserted that organisation in the form of logic and coherence is what separates random noise and musical form, comparing this structure to the application of grammar to a language. Schenker and Salzer (1969) was also convinced of the existence of organisation, and developed a pitch-based hierarchical model of musical form which he used to analyse a great many works. Both authors identified hierarchical elements based on pitch and rhythm, but did not offer a scientific formalism of their observations. Many other analytical approaches exist, including those which focus on performance (Lerch et al., 2019), compositional form, and even physical gesture (Gritten & King, 2006), where a listener's physical responses may be used to map expressive dynamics to the musical surface. We have selected structural analysis, and the role of grammars in performing such analysis, as the basis of this study.

Repetitive structures present within a musical piece may be leveraged to compress it. The Minimum Description Length principle (Rissanen, 1978) suggests that the best description of a given data series, such as that formed by the sequences of notes present in a musical score, is represented by the smallest model capable of reconstructing that series with the minimum error. Since repeating patterns within note sequences may be replaced by references to a single copy of each pattern, such patterns may be used to compress a sequence; this provides one method of generating a compact model, and thus, for our purposes, measuring description length.

Following this observation, we hypothesise that compression may be used to solve a number of musicological tasks, despite an absence of knowledge of the musical domain: to detect and correct errors, such as might occur during transcription; to classify pieces by melodic characteristic; to segment pieces in a manner similar to an expert human analyst; and to offer a structural approach to the manual editing of music. This paper extends the work of Sidorov et al. (2014), and validates the hypothesis experimentally, through the use of several compressors: Lempel-Ziv Welch (Ziv & Lempel, 1978), Burrows-Wheeler with run-length encoding (Burrows & Wheeler, 1994), and GZIP (Deutsch, 1996). We compare the performance of these general-purpose compression algorithms against two algorithms designed specifically for grammar-based compression: Zig-Zag (ZZ) (Carrascosa et al., 2010, 2011, 2012) and Iterative Repeat Replacement with Most Compressive score function (Carrascosa et al., 2010, 2011, 2012). Our experiments are conducted on a collection of 7928 musical scores gathered from sources which include the *Acadia*

CONTACT David Humphreys humphreysdj@cardiff.ac.uk

Early Music Archive (Callon, 1998–2009), the *Choral Public Domain Library* (CPDL organisation, 2018), and *Musopen* (Musopen organisation, 2018).

In Section 2 we discuss background material relevant to this study, and outline the materials and methods necessary for our experiments in Section 3, including a description of the music used and how it is represented. We then evaluate the performance of the above compressors for four different applications: detection of errors in music (Section 4), classification of musical pieces by tune family (Section 5), and two applications which exploit the structure which grammars provide (Section 6), namely automatic segmentation of pieces by J. S. Bach in the style of an expert musicologist, and the manipulation of a grammar to allow assisted editing of a musical score. Finally, we provide a summary of our findings.

2. Related work

The material here is separated into three specific areas. In Sub-section 2.1, we highlight string-based compressors, as these are the main focus of our study. In particular, we discuss those which generate grammars, and work which has demonstrated their success when applied to structure discovery given musical or non-musical input. In Subsection 2.2, we highlight studies which show compressors can be useful for computing pairwise information distance between inputs such as text and music, and therefore suggest that grammar-based compression is relevant to the task of music classification. In Sub-section 2.3, we highlight algorithms which were designed to discover patterns or structure in music, and discuss some of the challenges when selecting a suitable representation of musical data.

2.1. Grammars and compressors

A context-free grammar (Chomsky, 1956) is a member of the Chomsky hierarchy of formal grammars, and is useful for modelling the hierarchical representation of natural language phrases. A straight-line grammar is a form of context-free grammar which generates exactly one output string; the sizes of such models may be minimised by optimisation, providing a compressed representation of the input data. Structure present in the output of a grammar-based compressor may represent structure which is meaningful in the context of the composition of its input (Rissanen, 1978); for instance, the three sections of a Sonata may be reflected by the top level of a grammar's hierarchy being formed from three distinct rules, each aligned with its respective section of the original musical input. Grammar-based compressors have been successfully applied to both linguistic and biological problems, such as the discovery of structure within text and DNA (Gallé, 2011). An earlier study presented the Sequitur algorithm (Nevill-Manning & Witten, 1997), showing it was able to compress large DNA sequences more effectively than competing algorithms, and segment text in a meaningful manner, despite having no prior knowledge of either subject. Its ability to correctly select motifs from two Bach chorales was also demonstrated.

Carrascosa et al. (2010, 2011, 2012) showed that a variety of existing grammar-based compressors performed identical steps during the construction process. These compressors differed only in score function, selecting one of three specific functions: Maximal Length (ML), where the repeating term with the greatest length was chosen; Most Frequent (MF), where the repeating term with the highest number of occurrences in the input was chosen; and Most Compressive (MC), where both term length l and frequency f were combined as lf to allow selection of the term offering the greatest reduction in encoding length when all its instances were replaced within the input. The study presented an algorithm unifying these approaches, which they termed *Iterative Repeat Replacement* (IRR) schemes, and a new algorithm able to optimise constituent occurrence, using their *Zig-Zag* optimiser (ZZ). Occurrence optimisation addressed an issue present in all IRR algorithms, and consistently smaller models were produced by ZZ relative to IRR-Mx across inputs of varying size. Benz and Kötzing's GA-MMAS algorithm Benz and Kötzing (2013) used heuristics in its traversal of the same search space as ZZ, enabling it to construct grammars of equal size in fewer parses. Siyari and Gallé (2017) introduced the concept of flexible matching, where terms to be replaced within the input string were not limited to those which exactly repeat, but instead constructed from a pattern uwv where w was allowed to vary so that only the prefix u and suffix v of each pattern were equal. Experiments showed that their method was better able to identify syntactical structure in linguistic data than IRR or ZZ.

As demonstrated by Steedman (1984), grammars can model and delineate musically significant structures. Algorithmic construction of grammars can positively identify musical phrases, such as those found within Bach chorales by Sequitur (Nevill-Manning & Witten, 1997), a hierarchical grammar-based compressor. Abdallah et al. (2016) took a probabilistic approach to structural analysis by grammar, and provided an in-depth review of related methods, alongside a demonstration that such models could effectively segment symbolic scores within a dataset of Bach chorales into groups of repeating symbols represented by the right-hand sides of the grammar's rules. Simultaneously, the symbols used to uniquely identify each of these rules (non-terminals)

were taken as an indication of the class of each segment, so that the structure of the resulting parse tree might be used to infer relationships between them. They noted, however, that Markov models remained superior when applied to a corpus of folksongs gathered by the University of Essen (Schaffrath & Huron, 1995). Previous work by Sidorov et al. (2014) showed that straight-line grammars formed from musical sequences could be applied to tasks such as editing and error detection, and suggested their utility for summarisation, simplification, similarity estimation, and plagiarism detection.

Such works show that compression can be useful in the modelling of data, and that grammars constructed from music and other data can exhibit useful structural properties. Grammars generated by compression therefore have the potential to be an effective representation for use in analytical tasks.

2.2. Classification by compression distance

The size of a compact model of a given input, such as that produced by minimisation of the size of a straight-line grammar, may be considered an approximation of the Kolmogorov Complexity (Kolmogorov, 1963) of that input. This forms the basis for computation of pairwise normalised information distance which may be used to cluster inputs by similarity, and is therefore useful for automatically separating a collection of inputs into their respective classes. In their investigation of clustering by normalised information distance, Cilibrasi et al. (2004) experimented with different compressors, including GZIP (Deutsch, 1996), BZIP2 (Seward, 1996) and the Linux standard command *compress* (Welch, 1984).

Li et al. (2004) also used GZIP in their presentation of a normalised information distance. Louboutin and Meredith (2016) chose LZ78 (Ziv & Lempel, 1978) for their study of classification and pattern discovery.

Cilibrasi and Vitányi presented a method of compression-based clustering by similarity, shown to be generally invariant to the compressor employed (Cilibrasi & Vitányi, 2005).

There is, therefore, a wide range of compression techniques. But Cilibrasi and Vitányi also demonstrated that Normalised Compression Distance (NCD) may be leveraged against a wide variety of tasks, including genetic and viral classification, recovery of linguistic and literary structure, identification of style for musical compositions, character recognition, and astronomic anomaly classification. Numerous other studies have highlighted the power of comparison by compression distance, and the link between accuracy and compression strength, for tasks including image registration (Bardera et al., 2006), network dynamic classification (Nykter et al., 2008) and estimation of stem cell significance (Cohen et al., 2010). As grammar-based compressors are able to exploit structure within musical input data, it is reasonable to suggest they may be effective when applied to classification tasks.

2.3. Automated approaches to music analysis

The discovery of patterns in a musical piece might be attempted using common string processing techniques and tools (Apostolico & Galil, 2013) designed for tasks such as compression and pattern matching within text. Cambouropoulos et al. (2001) examined some of the challenges of representing musical data in a form suitable for processing by such algorithms. He showed that the use of pitch values alone, as opposed to a relative representation such as pitch intervals, prevented the discovery of equivalent musical patterns in different keys, and highlighted that seeking patterns within a sequence of interval values could prevent the discovery of two distinct instances, since, if these exist as a concatenated sequence, a single interval representing the current and subsequent pitch may belong to both. As stated in a separate study (Meredith et al., 2002), encoding the musical events within a score into temporal sequences of symbols is not ideal; simultaneous notes within a voice may be treated as separate events when repeating patterns are sought, and patterns split across voices may not be detected as repetitions. Cambouropoulos' work also presented a contour-based representation, which allowed algorithms which were normally constrained to replacing only terms which exactly repeated within a given input to instead replace terms whose contour exactly matched, even when their original representation did not, thus allowing them to perform flexible matching. The challenges of musicologically-useful pattern discovery through segmentation of the musical score were explored, with the author concluding that a segmentation based purely on exactly repeating terms is likely to produce unsuitable results.

Pearce et al. (2008) investigated the delineation of phrases within melodies, and provided a concise review of existing approaches. The *Generative Theory of Tonal Music* (Lerdahl & Jackendoff, 1983) outlines a set of grouping preference rules, and an implementation based on clustering and statistical learning (Kanamori et al., 2014) was shown to outperform existing methods when detecting 'local grouping boundaries' as defined by the GTTM. Grouper (Temperley, 2004) was presented as part of the Melisma Music Analyzer, and was a dynamic programming melody partitioning algorithm based on three *Phrase Structure Preference Rule* definitions. These were the *Gap Rule*, designed to bias choice of phrase

boundary location generally towards large inter-offset and onset-to-offset intervals, the *Phrase Length Rule*, which set a preference of approximately 10 notes per phrase, and the *Metrical Parallelism Rule*, designed to encourage grouping to occur in a symmetrical fashion. Temperley showed it was able to correctly predict the majority of phrase boundaries for pieces from the *Essen Folksong Collection* (Schaffrath & Huron, 1995). Cambouropoulos' *Local Boundary Detection Model* (Cambouropoulos, 2001) leveraged the magnitude of local note intervals to produce a boundary strength estimation for each score event, and was found to have comparable performance to an existing boundary detection method, the punctuation rule system (Friberg et al., 1998). Pearce et al. (2008) also presented *IDyOM*, an unsupervised inductive learning model which uses a definition of melodic closure to create significant phrase groups. This also exhibited performance comparable to existing algorithms on a subset of the *Essen Folksong Collection*.

An alternative method for pattern discovery was demonstrated with the Pattern Boundary Detection Model (PAT) algorithm (Cambouropoulos, 2006), which was shown to discriminate an appropriate description of the melodic surface based on exact matching, taking high repeat-onset incidence sums as an indication of increased boundary probability. A modified scheme capable of detecting melodically-similar segments was presented, although optimisation of a model generated by its selections was not considered, and further work was deemed necessary for creation of a robust, generalised implementation.

These studies highlight the importance of choice of music representation, and show that it is possible to recover some significant musical information using regularity-based compression of an interval representation, within the limitations of the scheme. They also show that automatic segmentation of a melodic surface is certainly possible, but do not examine whether domain knowledge is required in order to achieve equal success.

Meredith et al. (2002) presented the Structure Induction Algorithm (SIA) family of algorithms, designed specifically for the task of processing musical data. These algorithms were vector-based, and for a given score each musical event was represented as a single point in multi-dimensional space, where patterns which occur at different offsets but whose points have an identical relation to each other may be treated as equivalent. This design allowed the discovery of repeating yet non-sequential patterns from polyphonic music, distinguishing these algorithms from string-based methods in both ability and output. A compression-based version of a SIA algorithm, COSIATEC (Meredith, 2013), was shown to perform strongly on music data for many analytical tasks. COSIATEC represents a markedly different approach to string-based pattern discovery, making any direct comparison between the two classes of algorithm unsuitable in the context of the experiments performed in our study. As such, we have chosen not to consider it for this paper.

Another important aspect of the algorithmic analysis of music is evaluating the algorithm's ability to discern musically significant patterns, for which a reliable expert-defined ground truth is required. The Music Information Retrieval Evaluation eXchange (MIREX) (Choi et al., 2018) invited participants to submit algorithms with which to attempt the discovery of repeated themes and sections from poly- or mono-phonic symbolic representations of a pre-defined set of pieces. A target set of ground truth patterns was provided for each piece, taken from analyses by Barlow et al. (1948), Schoenberg et al. (1967), and Bruhn (1993), as shown in the task description on the MIREX Wiki (Choi et al., 2018). Both dataset and evaluation code were made publicly available, alongside the competition results for previous years, presenting an opportunity for performance comparison against new methods. In this study, we adopt MIREX resources to evaluate the effectiveness of grammar-based compressors on the task.

3. Materials and methods

In order to evaluate the performance of compressors on real-world musical data, a suitable compressor must be selected, and a large collection of digital scores is required which represents a useful sample from a population of scores varying in size, structure, and musical content. In this section, we discuss our selected compressors, describe how the score collection used in our experiments was assembled, and define how its data are represented in symbolic forms suitable for processing with our chosen algorithms.

3.1. Grammar-based compression

For this paper, the following definition has been adopted. A context-free grammar (Carrascosa et al., 2011) $G = \langle \Sigma, N, P, S \rangle$ is composed of a set of terminal symbols Σ (unique symbols present in the input string), a set of non-terminal symbols N (each representing a single production rule), a set of production rules P and a start rule S (both containing symbols from $\Sigma \cup N$). Grammar G is encoded as a sequence of rules (S, P) separated by a special terminator symbol. Conventionally, its size may be calculated as follows:

$$|G| = \sum_i (|P_i| + 1) + |S| + 1. \tag{1}$$

The goal of a smallest grammar algorithm is to minimise $|G|$ by generating a grammar containing only production rules which replace instances of substrings in S, P with their respective references in a combination which produces the smallest possible encoding. Computationally, construction of a smallest grammar is a complex process proven to be NP-Hard (Lehman & Shelat, 2002). However, a close approximation may be discovered with lesser complexity, making algorithms which generate a compact grammar, such as ZZ, suitable for practical use. Existing grammar construction methods vary in complexity and performance. Carrascosa et al. (2010, 2011, 2012) showed the time complexity of IRR schemes to be $O(n^2 \log n)$, compared to $O(n^7)$ for ZZ, where n is the length of the input in symbols. It is important to note that this represents maximum complexity; in practice, convergence usually occurs far earlier.

We have selected ZZ and IRR-MC as well-known grammar-based compressors suitable for our study. IRR-MC is included so that the results obtained from a less powerful compressor may be compared to those obtained using ZZ, to explore our hypothesis that there is a relationship between compressor strength and performance on our chosen experiments.

IRR-MC begins with P empty and N containing only a reference to S. Variable S is initialised with the input string σ, and thus all members of the unique set of its symbols are added to Σ. At each iteration, the most compressive substring of S is added as a new production rule, and its occurrence in S (or P) replaced with a rule reference symbol, which itself becomes a member of N. Once complete, the context-free grammar exactly generates σ alone upon expansion. The result is a greedy minimisation of $|G|$.

In contrast, ZZ traverses a superset lattice of possible substrings, known as constituents, where moving to a different node within the lattice represents the addition or removal of a constituent from the current set, and so of a production rule from P. Traversal ends when no move to a neighbouring node results in the possibility of constructing a more compact grammar from that node's constituent set, thus locally minimising $|G|$.

3.2. Corpus of musical scores

A substantial collection of symbolic music data was gathered for this study, sourced from expert transcriptions in order to avoid the high error rate to be found in Optical Music Recognition (OMR) transcriptions. The collection is comprised of scores from the following sources: the *Acadia Early Music Archive* (Callon, 1998–2009); the *Choral Public Domain Library* (CPDL organisation, 2018); *Musopen*, a repository of free scores and recordings (Musopen organisation, 2018); *Music21*, a toolkit for computer-aided musicology with a large accompanying dataset (Cuthbert & Ariza, 2010); *KernScores*, an online symbolic music library (Sapp, 2005); a digital archive of the 1850 edition of *O'Neill's Music Of Ireland* (Chambers, 2015); the *Meertens Tune Collections*, a database of Dutch folk songs (Meertens Instituut, 2018); and the Johannes Kepler University *Patterns Development Database* (Johannes Kepler University, 2013), itself using data from *KernScores*. In total, 7961 digital scores were gathered, of which 7928 were converted to a suitable common format and included in our corpus.

Table 1. Distribution of pieces by source; three primary sources make up the majority of the corpus.

Source	Proportion
Music21	**37%**
KernScores	**28%**
O'Neill's Music of Ireland	**25%**
Meertens Tune Collections	4%
Miscellaneous	4%
Acadia Early Music Archive	1%
Choral Public Domain Library	1%
Musopen	< 1%
JKU PDD	< 1%

The MTC (Meertens Tune Collections) *Annotated Corpus v2.0.1* (van Kranenburg et al., 2016) contains 360 songs bound to one of 26 'tune families' as defined by musicologists at the Meertens Institute, and described by Volk and Van Kranenburg (2012) during their study of the relationship between musical features and similarity. The collection has been extensively used in classification studies. However, the composer of each song is unknown.

Table 1 shows the distribution of pieces within the corpus by source.

The corpus contains works by 126 known composers, with 3467 pieces whose composers are marked 'unknown'. Of the latter, the majority belong to the two primary folk collections included: *Ryan's Mammoth Collection* (Howe, 1883), and *O'Neill's Music Of Ireland (1850)*, the last of which does credit an individual for each score in the original publication, but a digital copy of this data is not publicly available. Table 2 provides details of the distribution of composers. Pieces are also categorised by period and genre; Tables 3 and 4 show how the collection is divided. The 20th Century category contains folk music almost exclusively, obtained from the *Music21* and *O'Neill's Music Of Ireland* collections.

As an approximation of corpus complexity, times taken to construct grammars using ZZ were stored for each piece. These were used as a sort criterion to allow processing in order of run-time. This was necessary because, in the worst case, grammar construction time

Table 2. Distribution of pieces by composer; Bach and Palestrina make up a large portion of the corpus, but the majority of pieces have no recorded composer.

Composer	Proportion
Unknown	**46%**
Bach, Johann Sebastian	**20%**
da Palestrina, Giovanni Pierluigi	**17%**
Haydn, Joseph	4%
Corelli, Arcangelo	3%
Mozart, Wolfgang Amadeus	3%
Beethoven, Ludwig van	2%
Vivaldi, Antonio	2%
Others ($<$ 1% each)	4%

Table 3. Distribution of pieces by period, the three largest being 20th Century, Baroque and Renaissance.

Period	Proportion
20th Century	**43%**
Baroque	**25%**
Renaissance	**18%**
Classical	7%
Romantic	2%
Undefined	2%
Medieval	1%
19th Century	1%
Georgian	1%
Tudor	$<$ 1%

Table 4. Distribution of pieces by genre; most scores are either classical pieces or folk songs.

Genre	Proportion
Classical	**55%**
Folk	**43%**
Undefined	2%
Jazz	1%

for corpus pieces was too great to allow experiments to complete; processing in order of run-time allowed us to maximise the number of pieces used. Number of notes represented within each piece were also recorded. The relationship between symbolic length and complexity, approximated as the time required to construct a grammar for each piece using ZZ, is shown in Figure 1. This figure highlights the non-linear distribution of build times, and therefore build complexity, which characterises the corpus, and demonstrates the necessity for ordered processing.

3.3. Data representation

Several attributes may be associated with each note in a musical score, and in our experiments we separate selected attributes into individual strings of symbols. For example, the following pair of strings are generated given the two-voice score in Figure 2:

- *Chromatic pitch:* MIDI note values $0 \leq v \leq 127$ as defined in the MIDI 1.0 standard (MIDI Association, 1996), each representing an individual semitone (12 notes per octave, as defined by Western music, where middle C = 60). Rests are ignored.

$$p_{\text{chr}^1} := [60, 62, 64, 65, 67, 65, 64, 69, 62, 67, 67, 69, 67, 65, 64, 65, 64, 62, 60, 62, 60, 59]$$
$$p_{\text{chr}^2} := [67, 69, 71]$$

- *Diatonic pitch:* Note values $0 \leq v \leq 75$ taken directly from the *diatonic pitch* attributes of a Sibelius 7 representation of each piece, as defined in *Sibelius 7: Using the ManuScript language* (Avid Technology Inc, 2011) as 'the number of the "note name" to which this note corresponds, 7 per octave ... 35 = middle C, 36 = D, 37 = E and so on'. Rests are ignored, and accidentals are converted to their base notes.

$$p_{\text{dia}^1} := [35, 36, 37, 38, 39, 38, 37, 40, 36, 39, 39, 40, 39, 38, 37, 38, 37, 36, 35, 36, 35, 34]$$
$$p_{\text{dia}^2} := [39, 40, 41]$$

- *Chromatic/diatonic intervals:* For a string of pitch values p with $n = |p|$, we generate an interval string d with $|d| = n - 1$ with elements

$$d_i := p_{i+1} - p_i$$
$$d_{\text{chr}^1} := [2, 2, 1, 2, -2, -1, 5, -7, 5, 0, 2, -2, -2, -1, 1, -1, -2, -2, 2, -2, -1]$$
$$d_{\text{chr}^2} := [2, 2]$$

- *Note in chromatic/diatonic octave:* For a string of pitch values p, each element becomes

$$p_i := p_i \pmod{12}$$
$$p_{\text{chr}^1} := [0, 2, 4, 5, 7, 5, 4, 9, 2, 7, 7, 9, 7, 5, 4, 5, 4, 2, 0, 2, 0, 11]$$
$$p_{\text{chr}^2} := [7, 9, 11]$$

- *Chromatic/diatonic contour:* For a string of pitch values p with $n = |p|$, we generate a contour string c with $|c| = n - 1$ with elements

$$c_i := \text{sgn}(c_{i+1} - c_i)$$
$$c_{\text{chr}^1} := [1, 1, 1, 1, -1, -1, 1, -1, 1, 0, 1, -1, -1, -1, 1, -1, -1, -1, 1, -1, -1]$$
$$c_{\text{chr}^2} := [1, 1]$$

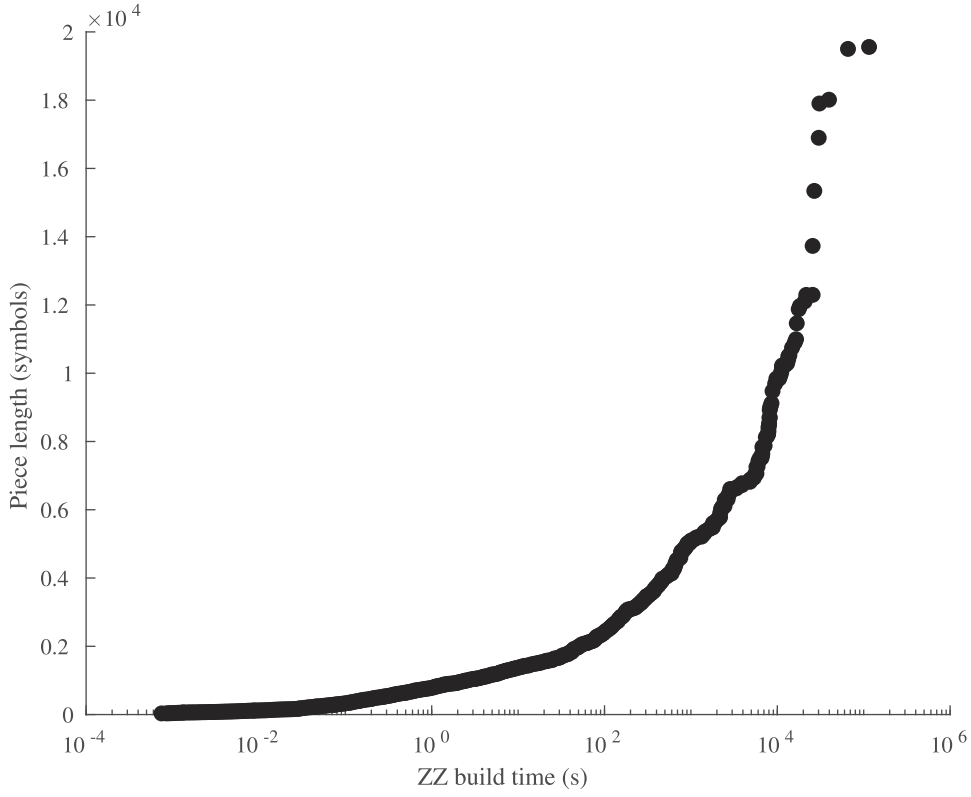

Figure 1. Relationship between symbolic length and approximate complexity of pieces within the corpus.

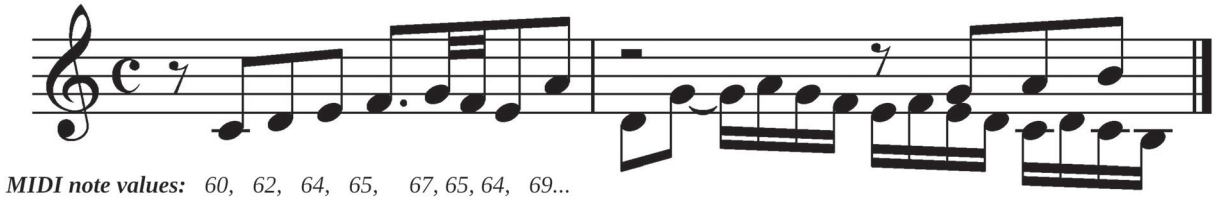

Figure 2. The first two bars from Bach's Fugue No. 1, WTC I., with MIDI note values shown for the first bar.

- *Duration:* Note duration values as defined in *Sibelius 7: Using the ManuScript language* (Avid Technology Inc, 2011), where 1unit $= \frac{1}{256}$ of a crochet.

$$d_1 := [128, 128, 128, 192, 32, 32, 128, 128, 128, 128, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64]$$

$$d_2 := [128, 128, 128]$$

- *Onset intervals:* From a string of note onset values, also defined in *Sibelius 7: Using the ManuScript language* (Avid Technology Inc, 2011) (where a bar has duration of 1024), we generate a string of intervals in the manner described for pitch intervals.

$$o_1 := [128, 128, 128, 192, 32, 32, 128, 128, 128, 128, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64]$$

$$o_2 := [128, 128]$$

Where multiple notes possess the same onset time, they are converted to a sequence, and ordered by note value (for instance, an inversion of the chord C-E-G, with note values 60, 64, 55, will become $[55, 60, 64]$). Where multiple voices are present within a score, regardless of the polyphony of any given voice, they are treated individually and presented to the grammar construction algorithm as separate strings. Where these multiple strings exist within a piece, they are concatenated, but separated by unique termination symbols to prevent matches being made over their boundaries. For instance, an input composed of the chromatic intervals of two voices may be generated as follows, with the symbol \$ chosen here as a unique terminator t_1:

$$S := [d_{\text{chr}^1}, t_1, d_{\text{chr}^2}]$$
$$:= [2, 2, 1, 2, -2, -1, 5, -7, 5, 0, 2, -2, -2, -1, 1, -1, -2, -2, 2, -2, -1, \$, 2, 2]$$

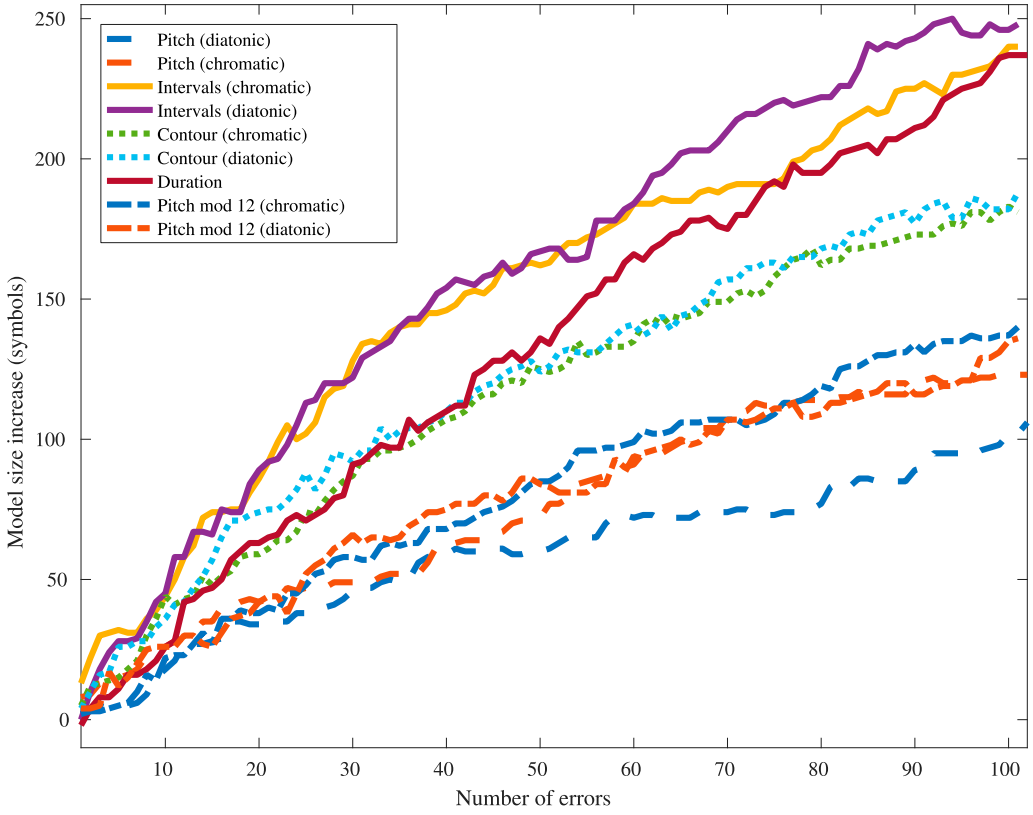

Figure 3. Reaction of grammar-based compressors to an increasing number of errors, in all available symbolic representations, for Bach's Fugue No. 10 from *Das Wohltemperierte Clavier* Book I. On average, diatonic intervals produce the strongest response.

4. Applying grammar-based compression to the prediction of data errors

4.1. Purpose

These experiments are designed to test the hypothesis that a grammar-based compressor may be used to detect the presence of data errors within a musical score. Any such error is likely to degrade the regularities present within the piece, reducing the ability of the compressor to exploit regularity and causing production of a larger grammar. Similarly, correction of an existing error should, in many cases, allow production of a smaller grammar. A musical 'spell-checking' system might be based on this technique, automatically locating incorrect data arising from OMR or human transcription errors. The experiments are constructed to demonstrate a relationship between data errors and compression strength, measured here as model size.

4.2. Method

We compare the performance of ZZ and IRR-MC to three general-purpose compressors, as a basis for what may be expected from standard tools. We selected LZW, Burrows-Wheeler Transform with run-length encoding, and GZIP for this study. BWT and LZ-derived algorithms have been shown to perform well on tasks involving the approximation of Kolmogorov Complexity (Kolmogorov, 1963), and produce a clearly defined symbolic output, the size of which may be easily computed and compared to an encoded grammar. Model size for ZZ and IRR-MC is defined in Equation (1). For LZW, model size is taken as the sum of the length of the alphabet and encoded output, plus a separator symbol between them. For BWT with RLE, model size is taken as the sum of the length of the encoding of each symbol and the number of times it repeats. For GZIP, model size is taken as the number of bytes necessary to store the compressed output on a filesystem. Because these measures are not directly comparable, we instead compute the ratio of compression achieved.

These experiments are computationally expensive. Evaluation of the change in ZZ model sizes to an increasing number of errors (Section 4.3.2) in each representation for Bach's Fugue No. 10 from *Das Wohltemperierte Clavier* Book I, shows that sequences of note intervals in the diatonic scale produce the strongest general response (Figure 3). Based on this result, we have chosen to evaluate performance on a diatonic interval representation of music alone, as described in Section 3.3. A 'ground-truth' model is produced by compressing this diatonic data. A model containing n errors is produced by selecting

n non-boundary symbols within the data, and altering them before generating a compressed model; we change selected values by ± 1 interval, to simulate a common single staff-line transcription error. The subsequent interval value is also changed so that the following notes remain unaffected.

Compressor response is measured by computing the difference in model size for ground-truth and altered data. Ability to correct errors is measured using precision, recall and F-measure, with the following definition:

- *True positive:* a correction of a previously altered value.
- *False positive:* alteration of a previously correct value.
- *True negative:* no alteration of a previously correct value.
- *False negative:* a failure to correct a previously altered value.

4.3. Experiments

4.3.1. Response to a single error

For each piece, we first compress the unaltered data, and measure the size of the model. Then, for each position selected as described in the following paragraph, we introduce a single error into the data, generate a compressed model, and measure its size. The difference in model size is taken as the response of the compressor to this single error.

We perform two variations of the experiment. In the first, all values within the piece are altered, but for a limited number of pieces since experiment run-time in this case is tl for each piece, where t is grammar build time and l is input length in symbols (as shown in Figure 1), making a test of the entire corpus impractical. In the second, only 25% of each piece is considered for candidate error positions, which allows every piece in the corpus to be tested.

Results – The results are presented in Table 5.

The results show that all the compressors tested respond to the presence of a data error, regardless of the difference in compressor strength or output encoding. Hypothesis testing, with the null hypothesis that the result values obtained for each pair of methods may come from the same distribution, confirms that the results for each method belong to Student's t-distribution (Student, 1908) and are statistically distinct from each other given a significance level of 5%. Result distribution is presented in Figure 4, where ZZ and IRR are seen to possess similar characteristics.

Table 5. F-measures: Accuracy of model size response to error.

Experiment	ZZ	IRR-MC	LZW	BWT	GZIP
1 (3107 pieces, asc. comp. time)	0.79	0.77	**0.81**	n/a	0.72
2 (all pieces, 25% of each piece tested)	**0.81**	0.79	0.73	0.60	0.72

The simple run-length encoded Burrows-Wheeler Transform gives the weakest response, followed by GZIP. The grammar-based methods show good performance, with ZZ consistently outperforming IRR, as expected.

Despite LZW being a poorer compressor in comparison to statistical coding techniques (Shanmugasundaram & Lourdusamy, 2011), it exhibits the most sensitivity to errors in the data when applied to short musical sequences. This occurs because LZW is able to take advantage of short, rarely repeating sequences within an input, whereas a grammar requires a greater overall gain before such sequences may be chosen as constituents for the model. Due to this fact, LZW is responsive to single errors within a greater proportion of each piece. Since constituents chosen for inclusion in a grammar might be considered structurally significant, we are primarily interested in the compressor's response to errors within these segments.

ZZ response over tested positions for two scores may be seen in Figures 5 and 6. The rule hierarchy chosen by the compressor is plotted above each note sequence in S, and the change in grammar size resulting from introduction of an error at a given position is plotted at the top of each figure, with standard deviation shown as grey shading.

4.3.2. Response to an increasing number of errors

For each piece of length l, we select a set of nl symbols to alter from a uniform distribution, choosing $n = 0.5$ as early experiments showed no significant response could be clearly detected beyond this threshold. We then introduce errors at $0 \leq p \leq nl$ positions, generating a compressed model at each iteration, and measuring its size. A change in model size from the model for $p = 0$ is taken as the response of the compressor.

Pieces were split by length into three groups, $l =$ 1–200, 201–1000 and $\geq$1001, so that the response to inputs of different lengths could be observed. Since distribution of piece length within the corpus is non-linear, a strong decline in the number of data points comprising the average occurs as e increases. For this reason, only results for smaller counts should be considered an accurate representation of response.

Results – The results are presented in Figures 7–12.

Different compressors produce differently encoded models, and so the sizes of their models may not be directly compared. To aid our attempt to compare compressor performance, we calculate the compression ratio each model achieves. Hypothesis testing confirms that results for each method belong to Student's t-distribution given a significance level of 5%, but not all results within

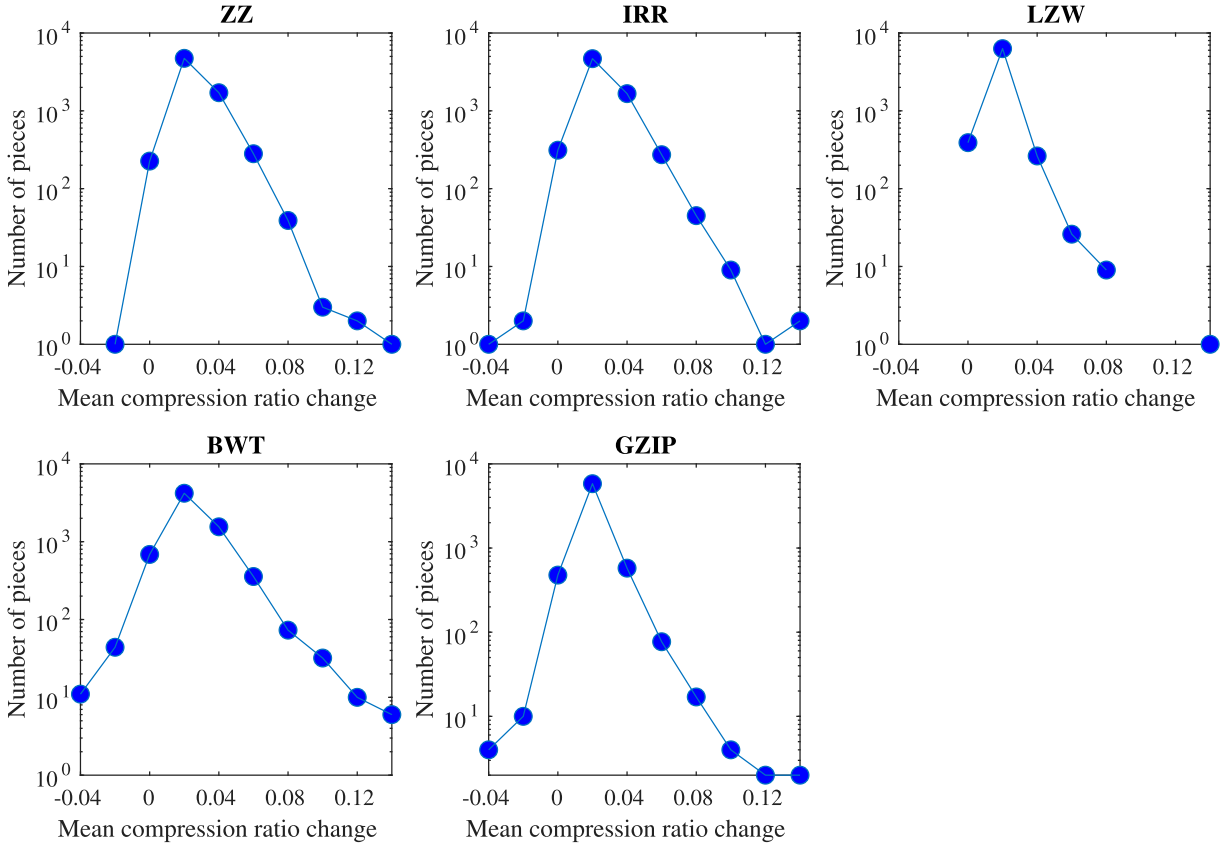

Figure 4. Histograms highlighting the overall trend in response for each compressor. Each figure shows the distribution of mean compression ratio change observed for each piece, following the introduction of a single data error. Note that the LZW figure contains three ratio groups which contained no pieces ($y = 0$).

each group are distinct from each other. For the group 1–200, the null hypothesis is true between ZZ, IRR and BWT. Where pieces are of length ≥1001, ZZ and IRR are not distinct from each other, however it is important to note that the group's sample size is markedly small at 134 pieces, which may account for the failure in differentiation.

All piece groups exhibit a detectable rise in model size as number of errors increases, until approximately $p \geq 0.25l$, where a significant proportion of the input has become corrupted; such response is supportive of the hypothesis. Of the three groups (lengths 1–200, 201–1000, ≥1001, Figures 7–9), the first two then show a reduction in response with the 1–200 group levelling, perhaps as structure within the input data becomes degraded to that of noise. The group formed from pieces of length ≥1001 does not exhibit a decreased response within the range tested, $p \leq 0.25l$. Instead, model size continues to grow, suggesting this larger input data contains more structure which may be compressed, and is therefore more error-sensitive.

All groups show a similar response, but with more instability as group sample size decreases, as might be expected. Each existing error simultaneously raises the probability of encountering noise, and also reduces the available structure from which to discriminate errors. This causes the reduction in response with increasing errors, resulting in a plateau where detection is no longer possible. Determination of the point at which this effect begins for pieces of high complexity is left for a future exercise.

Perhaps the most significant result is the clearly superior performance of GZIP in the group containing pieces of length ≥1001, where it responds with greater sensitivity than all other methods; for pieces of length ≤1000, ZZ offers the best performance. This suggests ZZ and GZIP are best able to compress and therefore leverage structural information from smaller and larger pieces respectively.

Standard deviation is significant, with each method showing a highly similar response; for this reason, plots for ZZ alone are included here (Figures 10–12). Greater variance as p increases is to be expected, since errors in this context may introduce different structure as well as degrade that which exists. However, it is notable that an overall decrease in compression occurs in all cases, demonstrating the response of the compressor to overall and not simply local structures. Variance has an apparent relationship to p, with the greatest effect visible in

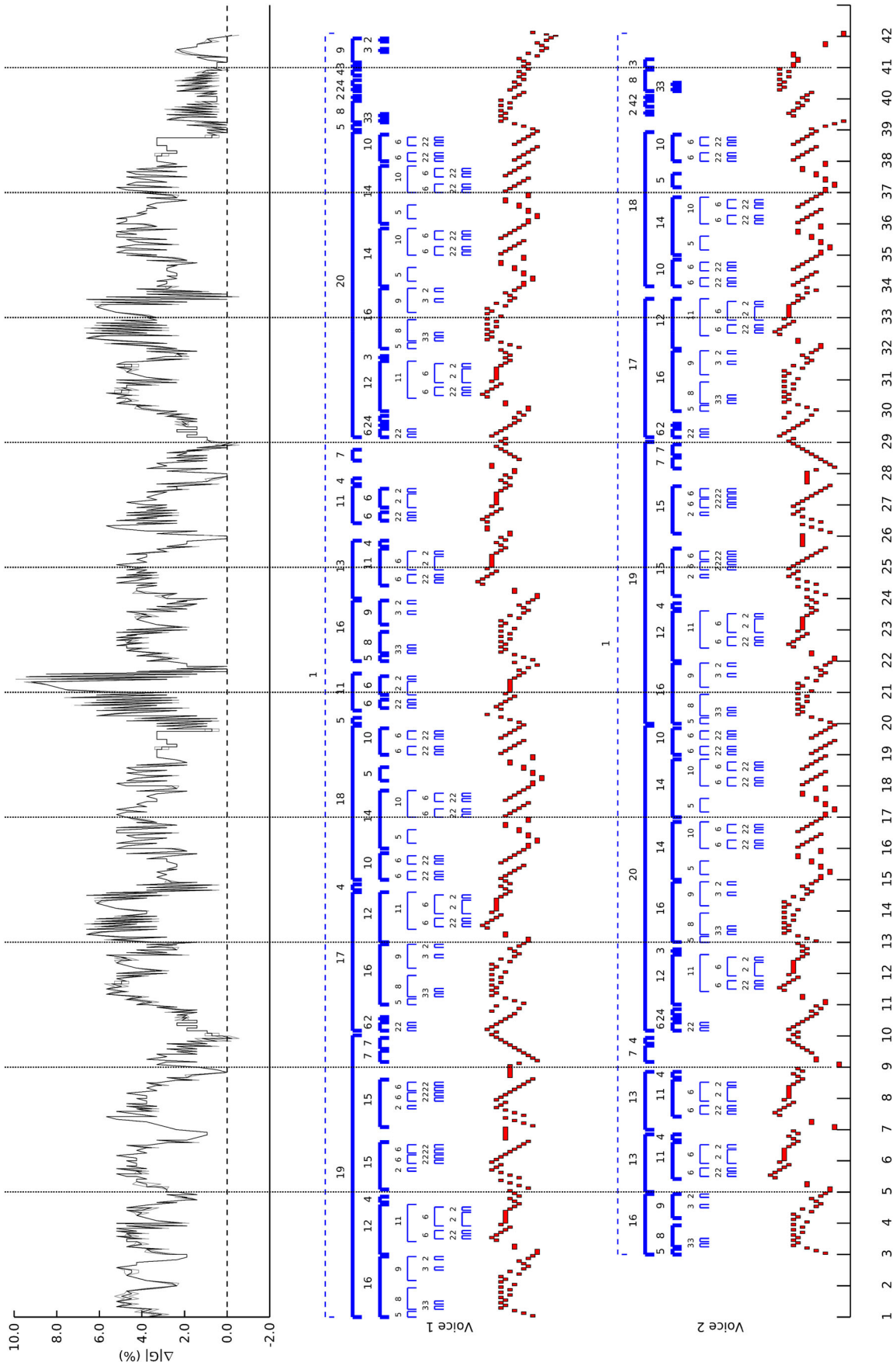

Figure 5. Response to data errors over two passes of all note positions using ZZ, for Bach's Fugue No. 10 from WTC I.

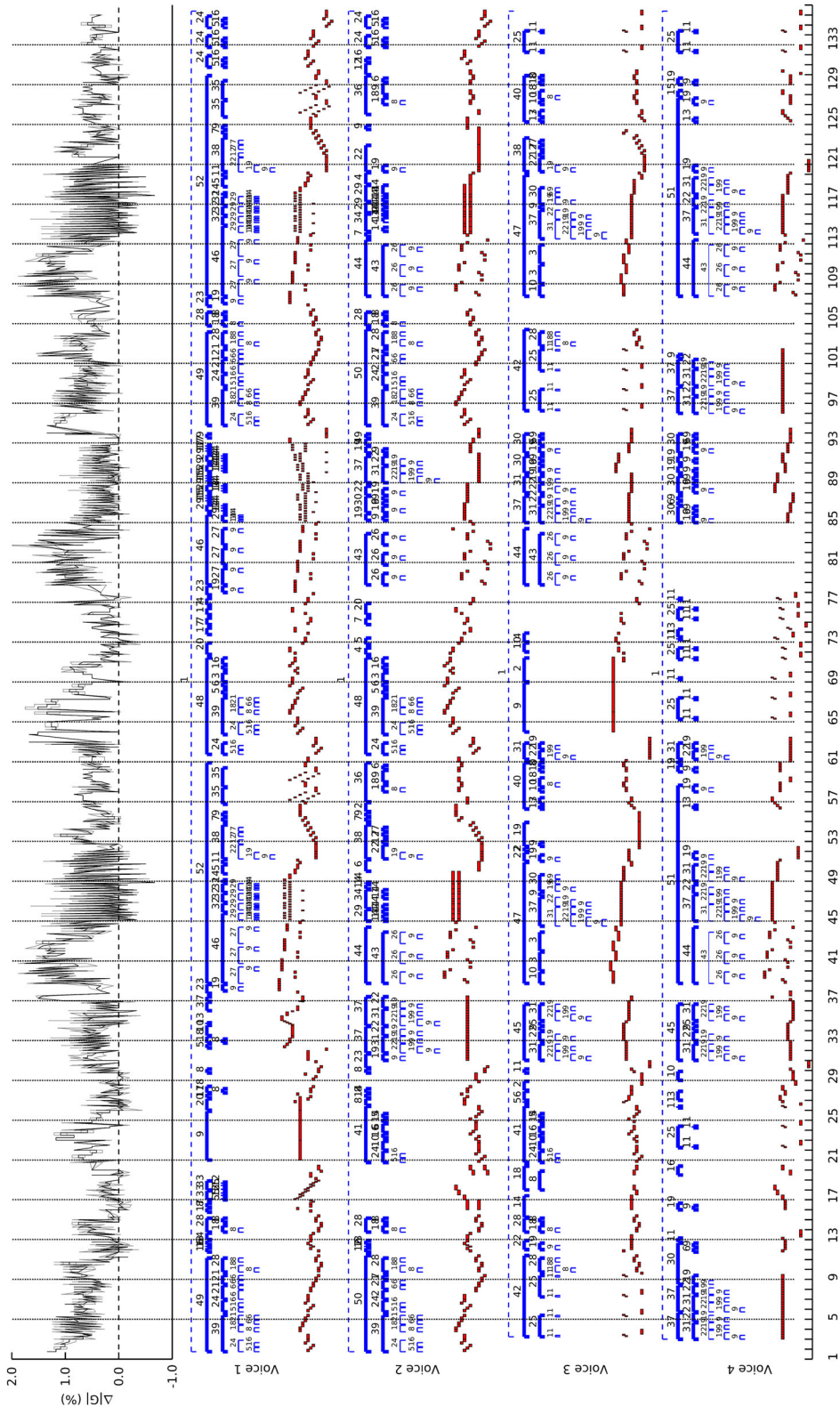

Figure 6. Response to data errors over two passes of all note positions using ZZ, for the Finalé of Haydn's String Quartet No. 22 in G major, Op. 17 No. 5. A dense and shallow hierarchy can be seen in the first voice in bars 85-93, suggesting structure of less significance to an exact compressor during this period. Musical content here is varied and unique within the score, producing an expected reduction in compression, in clear contrast to the majority of the piece.

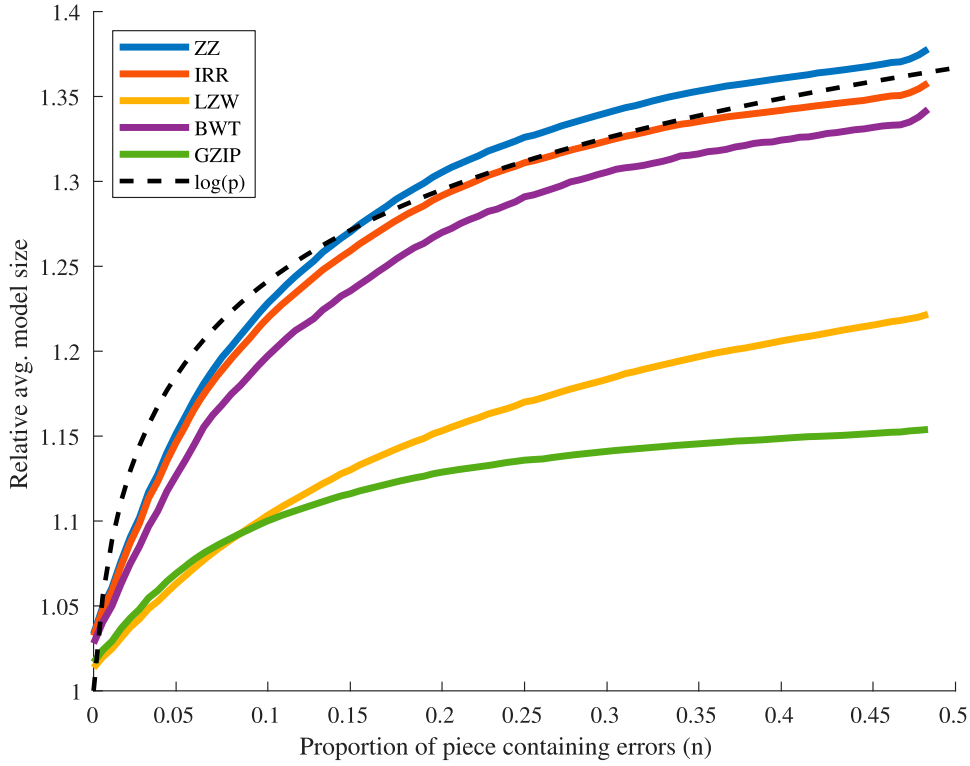

Figure 7. Average response of each compressor to an increase in the proportion of each piece containing errors (group 1: pieces of length 1–200).

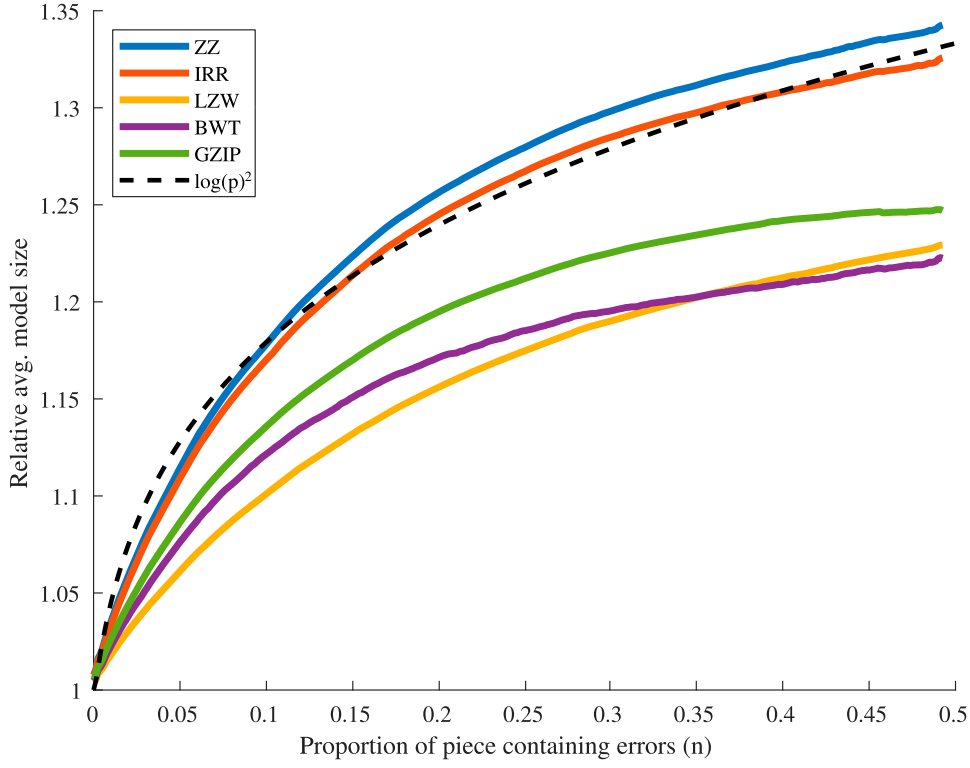

Figure 8. Average response of each compressor to an increase in the proportion of each piece containing errors (group 2: pieces of length 201–1000).

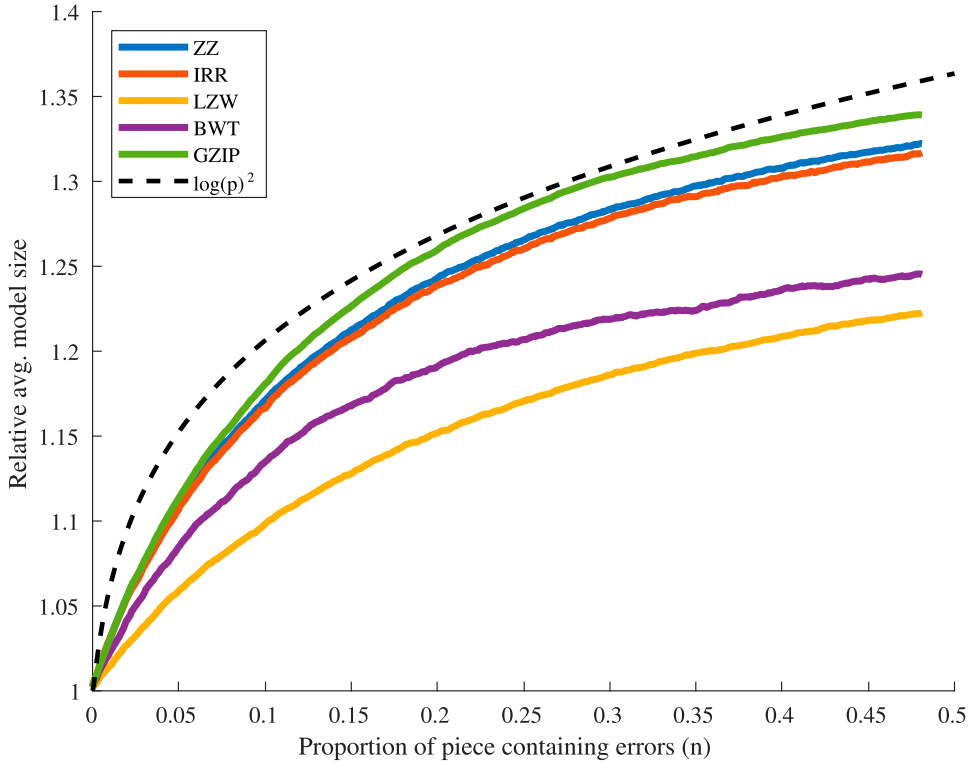

Figure 9. Average response of each compressor to an increase in the proportion of each piece containing errors (group 3: pieces of length 1001+).

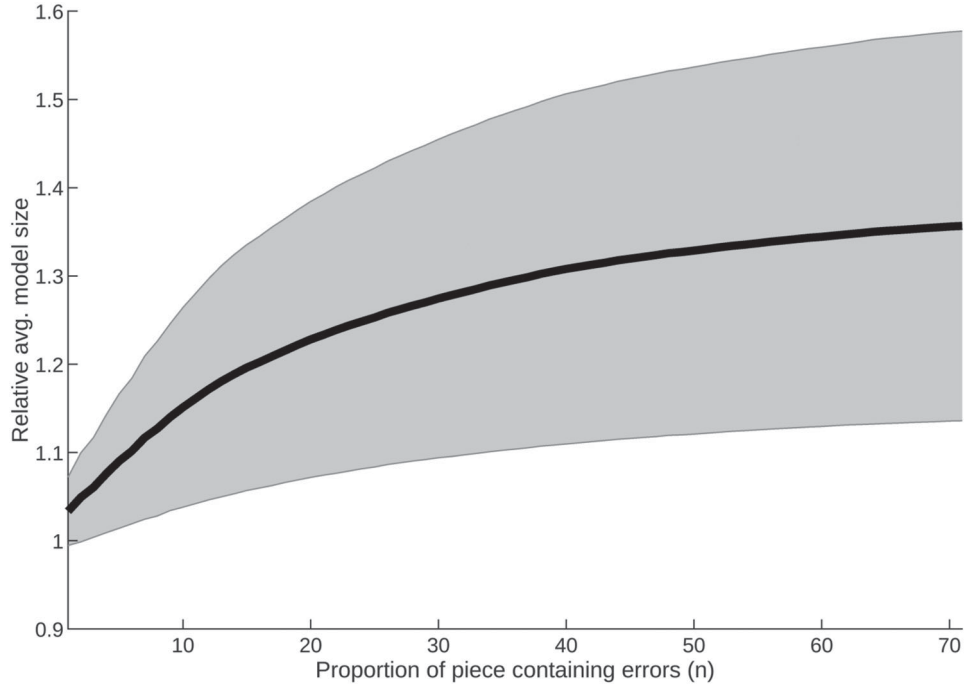

Figure 10. Average response of ZZ to an increase in the proportion of each piece containing errors, with standard deviation (group 1: pieces of length 1–200).

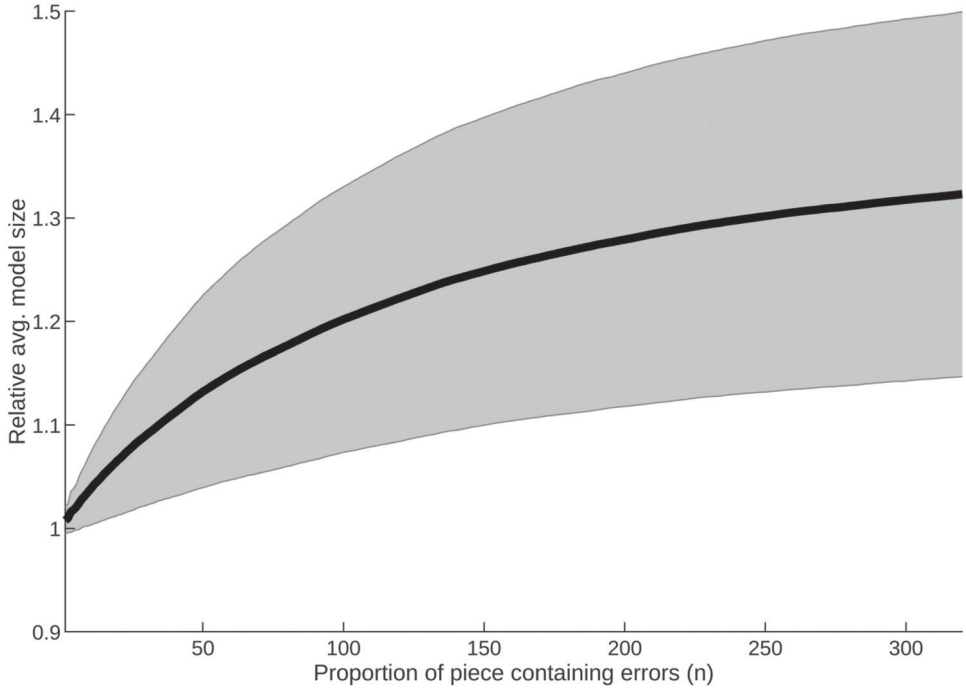

Figure 11. Average response of ZZ to an increase in the proportion of each piece containing errors, with standard deviation (group 2: pieces of length 201–1000).

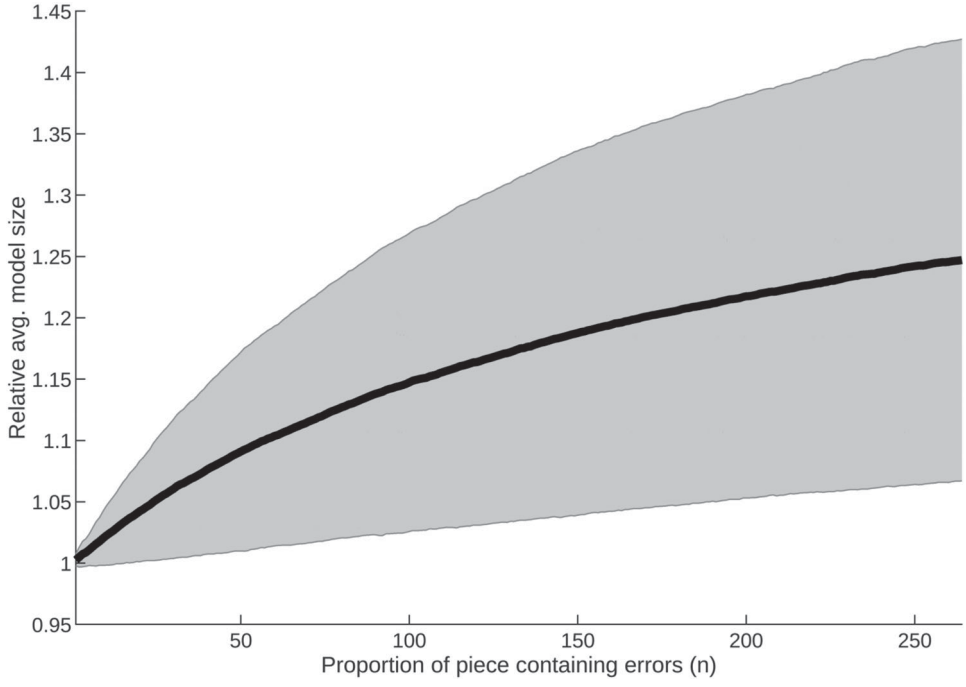

Figure 12. Average response of ZZ to an increase in the proportion of each piece containing errors, with standard deviation (group 3: pieces of length 1001+).

the group of largest pieces. The plot for the 201–1000 group contains some local maxima, correlating with common piece lengths within it. Overall, such large variance means no method in this study may be relied upon to show a strong response to given errors within music data, but since all methods exhibit decreasing compression a response will certainly occur, albeit minor in magnitude.

Table 6. *F*-measures: correct selection of candidate error position; pieces are tested in ascending order of computation time.

Experiment	ZZ	IRR-MC	LZW	BWT	GZIP
$m = n = 1$ (565 pieces, 100% of each piece tested)	**0.22**	0.20	0.19	n/a	0.16
$m = 0.5, n = 0.25$ (565 pieces, 25% of each piece tested)	**0.27**	0.24	0.26	0.15	0.20
$m = 0.5, n = 0.25$ (5735 pieces, 25% of each piece tested)	**0.35**	0.32	0.21	0.12	0.24

4.3.3. Automatic selection of candidiate transcription error positions

Building on the previous experiments, we now present a novel method capable of automatically selecting notes believed to be errors, and designed as an aid to the process of transcription error correction.

For each piece, we create $c = mp$ representations of the piece, each containing a single error in one of the c positions, and a compressed model is then constructed for it, with its size taken as a baseline for a version of the piece containing an error.

For each such version, we alter $p = nl$ positions individually by ± 1 interval. Exactly one such change will correct the error in that version. We construct a compressed model for each potential correction, and measure its size. Following our hypothesis that a musical piece which contains errors has a degraded structure and is therefore less compressible, we compare the size of the resulting model with that of the version containing an error. Any alteration which results in a smaller model size is taken as a likely successful correction, and identification of a candidate transcription error position.

For the c versions of each piece into which an error was introduced, precision, recall and F1 are calculated for each alteration, to evaluate our method's performance. Any size smaller than the baseline is taken as a positive, and any greater than or equal to the baseline as a negative. Two versions of this experiment are conducted, the first with $m = n = 1$, the second with $m = 0.5$ and $n = 0.25$, as preliminary experiments showed this would allow a reasonable proportion of the corpus to be processed within the available time. The second experiment is repeated on the same pieces used for the first, to provide an indication of the difference in observed performance when testing a smaller proportion of each piece.

Results – The results are presented in Table 6.

Hypothesis testing again confirms that results for each method are statistically distinct from each other. Result distribution is presented in Figure 13, where ZZ and IRR can be seen performing consistently well.

Table 7. Average ratio of compressed to uncompressed data for pieces in both experiment groups, for each compressor.

Experiment	ZZ	IRR-MC	LZW	BWT	GZIP
$m = n = 1$ (565 pieces)	0.86	0.88	0.86	n/a	1.13
$m = 0.5, n = 0.25$ (5735 pieces)	0.89	0.89	0.87	0.838	0.728

These figures again suggest a relationship between the strength of each compressor and its performance on this task. When provided with more complex data for the second experiment, the ability of GZIP to compress with greater strength than LZW (Savakis, 2000) is clear, and BWT shows poor accuracy, likely a result of reduced compression. ZZ consistently produces the best result ahead of IRR, which is by comparison a naïve method. Perhaps most interesting is the clear advantage both grammar-based methods exhibit.

An overall increase in *F*-measure can be seen to take place with $m = 0.5$, $n = 0.25$ instead of $m = n = 1$, partly due to the increased probability of correctly selecting a candidate selection from the smaller option set. This limitation is also present in the test results from the larger group of 4390 pieces; actual performance will be poorer than shown if pieces are processed in their entirety.

Although direct comparison of compressors by the length of their encoded representations is not possible without full consideration of encoding differences, it is interesting to note that average piece compression ratios, as may be seen in Table 7, generally support the hypothesis that greater compression results in best performance for this application.

Method practicality – Although theoretically interesting, selection of candidate error positions in this manner is a computationally complex task, and potentially impractical. If t corrections per position are to be tested, given ZZ complexity $g = O(n^5 \times m^2)$, where $n =$ number of symbols in the input sequence and $m =$ number of constituents per node in the lattice (Meredith, 2014), and substring search complexity $s = O(n^2)$, the computational complexity of this method is $O(t(n^7 \times m^2))$. However, this upper bound is rarely reached in practice, and the experiment highlights the superior ability of grammar-based compressors over the tested algorithms to identify musically incorrect structure.

5. Classification

We apply ZZ and IRR-MC to classification of the Meertens Tune Collections *Annotated Corpus v2.0.1* (van Kranenburg et al., 2016) by 'tune family', as defined by expert musicologists from the Meertens Institute. Selecting this widely attempted task provides an opportunity to examine the performance of our method in the context of

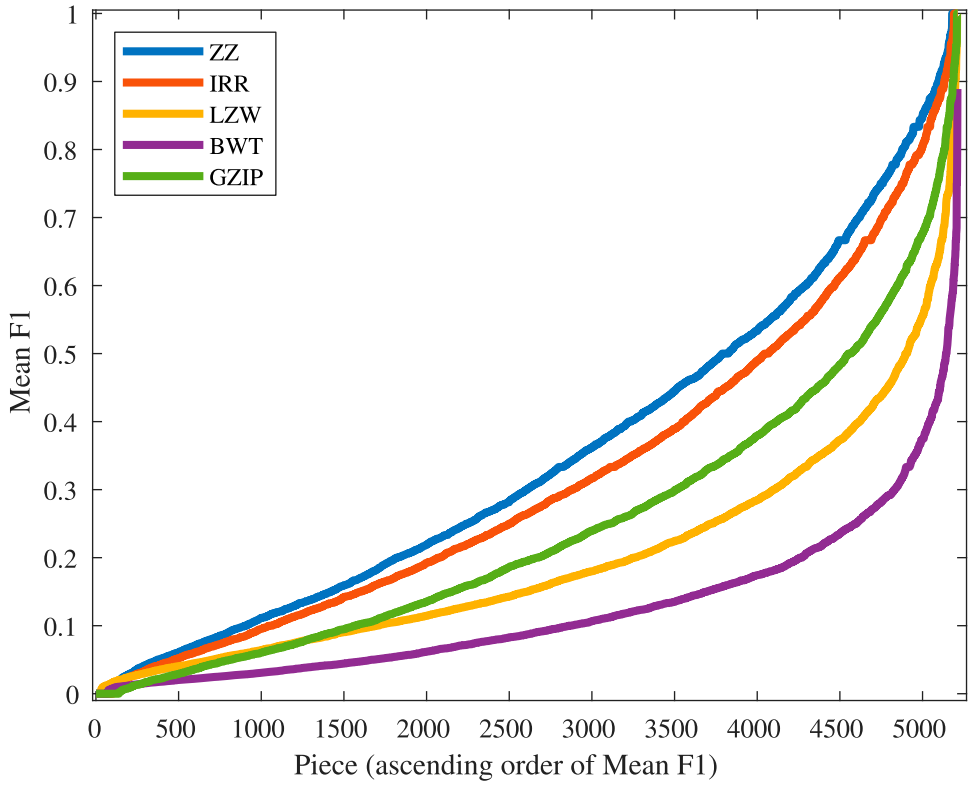

Figure 13. Distribution of mean *F*-measure per piece for each compressor (in ascending order).

many published studies. We evaluate its success for eight individual musical representations, and overall performance when these are weighted and combined to classify each piece in the collection.

5.1. Purpose

This experiment is designed to test the hypothesis that the computed compression distance between grammar-based models for two musical scores may represent an approximation of their similarity, and may therefore be useful in the classification of scores by pairwise distance. An ideal grammar construction algorithm will select the set of patterns which, when replaced within the input, produce the smallest model. Thus, where a pattern exists in both scores, it provides more potential for compression than a pattern unique to one score, and compressing scores with common components is likely to produce smaller models than those generated from dissimilar pieces.

5.2. Method

For a given compressor C, scores are selected in a pairwise fashion, and eight strings $x = d_1$, $y = d_2$ produced, one for each individual representation described in Section 3.3 except *note in diatonic octave*. For each set of strings, three models are constructed: $C(x)$, $C(y)$, and $C(xy)$, where xy represents a concatenation of x and y separated by a unique symbol. A Normalised Compression Distance (Li et al., 2004) is then computed for each pair of scores, as defined in Equation (2):

$$\mathrm{NCD}(x, y) = \frac{C(xy) - \min(C(x), C(y))}{\max(C(x), C(y))} \tag{2}$$

For each representation, 1-Nearest-Neighbour classification (Cover & Hart, 1967) is used to cluster scores by distance, with leave-one-out cross-validation (Kohavi, 1995) used to evaluate 'tune family' prediction accuracy against expert-defined ground truth. The known overall success rates for each representation are then used as weights in combining class predictions for each score, and their sum used to arrive at the final class. Success rate r is calculated as follows, where c is the number of correct predictions, and t is the total number of pieces tested:

$$r = c/t \tag{3}$$

5.3. Results

The results for each individual representation using ZZ are shown in Table 8. Success rates for each compressor when combining weighted representations is shown in Table 9.

Table 8. Per-representation success rates from NCD classification of the MTC-ANN v2.0.1 using ZZ.

Representation	Success rate
Intervals (chromatic)	0.88
Pitch (diatonic)	0.87
Pitch (chromatic)	0.87
Intervals (diatonic)	0.85
Octave note (chromatic)	0.76
Contour (diatonic)	0.69
Contour (chromatic)	0.68
Duration	0.63

Table 9. Rate of successful classification of pieces from the MTC-ANN v2.0.1 for ZZ and IRR-MC.

ZZ	IRR-MC
0.92	0.83

Of the individual music data representations used, chromatic-based pitch interval vectors produces the greatest success rate, with both pitch and pitch-interval vectors generating the strongest response from the types tested. This is perhaps unsurprising; a chromatic representation retains all available pitch detail, and use of intervals offers some degree of invariance to repeating patterns which are transposed within a piece. Since the MTC *Annotated Corpus v2.0.1* contains short strophes (average length is 48 notes), often in a single key, it is reasonable to expect transpositions to be less important than in longer works. This may be reflected in the good performance of both pitch vectors. The duration vectors provide least success during classification; distribution of note lengths within the *Annotated Corpus v2.0.1* is highly skewed, with one duration alone accounting for over half of all instances, and another for 30%. This causes a reduction in information from which to characterise each tune family, a likely reason for this representation's loss of accuracy.

Despite performing well, ZZ is not able to improve upon the techniques used in some existing studies, most notably van Kranenburg et al. (2013) who achieved an accuracy of 0.99 using a combination of Inner Metric Analysis, pitch, and note phrase-offset features. Table 10 provides a comparison of classification results on the *MTC-ANN v2.0.1* for selected studies.

Within the context of these studies, our method performs as may be expected of a sequence-based similarity model without domain knowledge. Kranenburg et al. computed rate of success when interval sequences only were used, achieving an average of 0.92. There, transposition was used to place each pair of scores into a common key, and the Needleman-Wunsch alignment (Needleman & Wunsch, 1970) balanced to minimise the penalty for continued shifting of a pattern segment when alignment is sought. This allowed for flexible pattern matching, akin to human-like recognition of simple musical variations. It is possible that this strategy provided a performance gain similar to that obtained from our use of multiple representations, suggesting that the addition of flexible matching may further improve our method. However, verifying whether this is the case is left for future work.

Table 10. Classification success rates achieved by various methods on the MTC-ANN v2.0.1.

Work	Avg. success rate
van Kranenburg et al. (2013)	0.99
Stober (2011)	0.98
Conklin (2013)	0.97
Goienetxea et al. (2016)	0.96
Louboutin and Meredith (2016)	0.94
Hillewaere et al. (2014)	0.94
Boot et al. (2016)	0.93
This work	*0.92*
Velarde et al. (2013)	0.84
Meredith (2014)	0.84

Stober (2011) adopted a generalised approach, creating a distance measure based on the weighting of various domain-based facets, such as chords and harmonies. In contrast to the average success rate of 0.99 from van Kranenburg et al. (2013), he achieved an average of 0.97 where the class of the query piece is unknown. Our method's lower success rate is expected given the lack of domain knowledge employed, and perhaps it is reasonable to suggest that use of pitch-based facets alone for Stober's method might cause a drop in accuracy similar to that seen with these constraints in the study by van Kranenburg et al. (2013).

Conklin (2013) and Goienetxea et al. (2016) reported average success rates of 0.97 and 0.96 respectively, both higher than our method, and also incorporated domain knowledge in the form of viewpoints, each representing features derived as a function of note pitch, duration and onset time. In particular, Goienetxea et al. (2016) employed a reductive heuristic based on the novelty of common patterns, and both works suggested that features such as motifs, metric, and phrase information are important in addition to pitch for this task. The increase in accuracy our results show when distance is based on a weighted combination of representations supports these assertions, and the greater success of all these studies where higher level musical features are used suggests that such an addition might also improve the accuracy of our method.

Work by Louboutin and Meredith (2016) employed morphetic pitch alone (Meredith, 2006), and reported an average success rate of 0.85 where COSIATEC was used to process a single viewpoint, similar in informational terms to where our study operates on diatonic pitch vectors. Their method outperformed ours where different compressors, including LZ77 (Ziv & Lempel, 1977),

were given multiple viewpoints and the results combined. Their work highlights the suitability of LZ77 in the analysis of polyphonic music, and the potential gain from leveraging compressed models of various types, a strategy which might also improve our method. Hillewaere et al. (2014) employed a flexible approach, using Levenshtein distance for pairwise alignment of both melodic and rhythmic data, but without compressed modelling. They achieve an average success rate of 0.94, compared to 0.92 by van Kranenburg et al. (2013) using the same representation, suggesting a superiority of edit distance over Needleman-Wunsch, and highlighting a link between flexible patterns and musical variations.

Interestingly, Boot et al. (2016) reported best results for sequence alignment of uncompressed data, where 0.93 is achieved using a note-to-note correspondence comparable to that employed by van Kranenburg et al. (2013). It is important to note that all cited studies related to the latter work demonstrate high success rates despite making use of fewer musical representations than our study, showing that higher level modelling of features may be more significant to tune family classification than a wider combination of lower-level data. However, we can confirm our work improves on results from Volk & Haas when using compressed representations, perhaps pointing to the superiority of grammars in isolating patterns significant to classification of the MTC.

Velarde et al. (2013) recognised that their Haar Wavelet analysis method underperformed in comparison to string-matching approaches when classifying the *Annotated Corpus v2.0.1*, and showed that the use of chromatic pitch instead of a scale-based representation such as morphetic pitch limited effectiveness. However, chromatic pitch provides the best success rate in our experiments using grammar-based compressors. It is possible that combining results from wavelet analysis of multiple representations could also present an improvement.

In conclusion, existing studies suggest the performance we achieve on this task is to be expected when using grammar-based compression of such sequences as the basis of a distance metric. The addition of flexible matching which better captures musical structure, higher level feature modelling, and, perhaps, combining grammar-based distances with those of other compressed models is likely to offer a measurable improvement.

6. Segmentation

6.1. Purpose

This experiment is designed to test the hypothesis that, following the Minimum Description Length principle (Rissanen, 1978), generating a model of music data using a grammar-based compressor may cause division of that data to occur in a musicologically significant fashion, resulting in structure which is similar to that which a human expert would define for the piece. The ability to automatically segment a score into meaningful segments could have several applications. Primarily, an accurate analysis of a piece may be obtained very quickly, providing an aid to academics and performers and a potential insight into the intentions of its composer. Indeed, the differences between an automatic and human-made segmentation may themselves highlight overlooked interpretations, or simply an alternative perspective, consideration of which may benefit musicological knowledge. Such a model might also be employed as a compositional aid, allowing alterations in flow and structure to be made to a score at a high level, but using meaningful 'units' of note sequences. These experiments are designed to demonstrate the degree of similarity between computationally-derived structures and human analyses, and the potential for high-level score manipulation, when grammar-based compression is employed.

6.2. Method

We select the diatonic interval representation for our segmentation experiments, encoding each piece as a concatenated sequence of voices as described in Section 3.3. For our examination of grammar-assisted editing, we choose chromatic intervals, note onset intervals and durations, since all notes within a score may be modelled using this combination. In each experiment, ZZ is then used to construct a grammar for each piece, and all sub-rules within S are selected and processed using the following experimental methods.

6.3. Experiments

6.3.1. MIREX 2016 discovery of repeated themes & sections task

Grammars are built from each of the pieces in the Johannes Kepler University Patterns Test Database (Johannes Kepler University, 2013). Each sub-rule of S is passed to the MIREX 2016 code designed to evaluate algorithm performance on the symbolic, polyphonic 'discovery of repeated themes and sections' task. We focus on two of the available metrics: *establishment* and *occurrence*. *Establishment* is a measure of an algorithm's ability to identify any instance of a ground truth pattern, whereas *occurrence* measures its ability to identify all instances within a piece. As defined by the evaluation procedure, matches with a score threshold ≥ 0.75 are selected as positive identifications. F-measures are

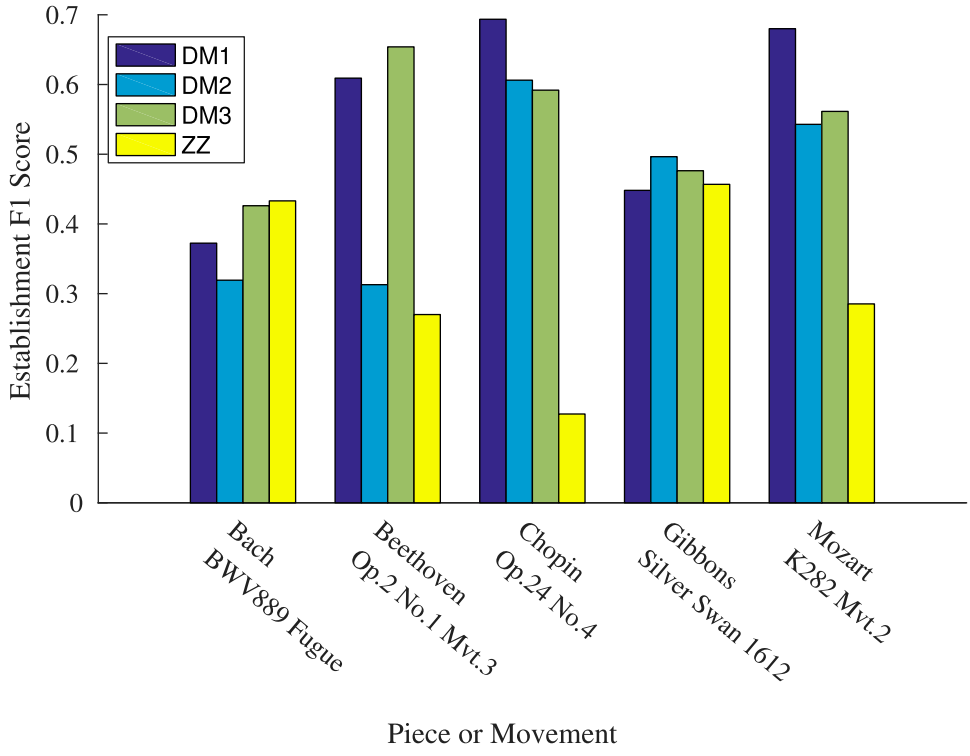

Figure 14. MIREX 2016 Discovery of Repeated Themes & Sections symPoly task, Establishment F1 score for 2016 algorithms DM1, DM2 & DM3 with additional ZZ results.

calculated from each metric, and these are compared to the official results for 2016.

Results – The results are shown in Figures 14 and 15.

ZZ outperforms all other methods in identifying patterns for Bach's Fugue No. 20 from *Das Wohltemperierte Clavier* Book II, and improves on the poorest method when seeking any instance of ground truth patterns within Gibbons' *The Silver Swan*, although it fails in retrieving all instances. On all other pieces in the dataset, it responds most poorly, in particular failing to identify pattern sets within the performance threshold for Beethoven's Op. 2 No. 1 Movement 3, or Chopin's Op. 24 No. 4.

ZZ's strong performance on the Bach Fugue may be attributed to the frequent repetition of its subject, for which exact sequence matching is most suitable. Algorithms DM1–3 are based on SIATEC (Meredith et al., 2002), an algorithm capable of flexible matching when applied to inexact sequences of significant similarity. This makes them more suitable for use with musical data containing variations, such as the pieces by Beethoven and Chopin. The suggestion that grammars built on exactly repeating patterns are unsuitable for such data may be reasonably supported by ZZ's strength in pattern establishment over identification of instances; difference of a single symbol within a pattern instance causes ZZ to split the entire sequence around this symbol, whereas SIATEC simply omits the point from its pattern definition, resulting in the latter's greater ability to retrieve all pattern instances.

6.3.2. Structural analysis of bach's well-tempered clavier

Given the strong response obtained to the Bach Fugue on the MIREX 2016 task, we investigated the response of grammar-based compression to Bach works further, in an attempt to evaluate whether stuctures present within the grammar may directly relate to those deemed significant in a specific musical analysis. We select the first eight pieces from Bach's *Das Wohltemperierte Clavier* Book I for use in this experiment. For each piece, a set of segments is first defined as *(start, end)* offsets into the sequence of intervals used to represent it. Each segment is defined by the following process:

- Where Bruhn (1993) shows a definite start and end point for a given repeating section, the intervals representing these sequences are located within the input data.
- For each instance of these sequences, *(start, end)* pairs are defined for each *exactly* repeating sub-pattern, since our chosen compressor operates only on groups of exactly repeating symbols.

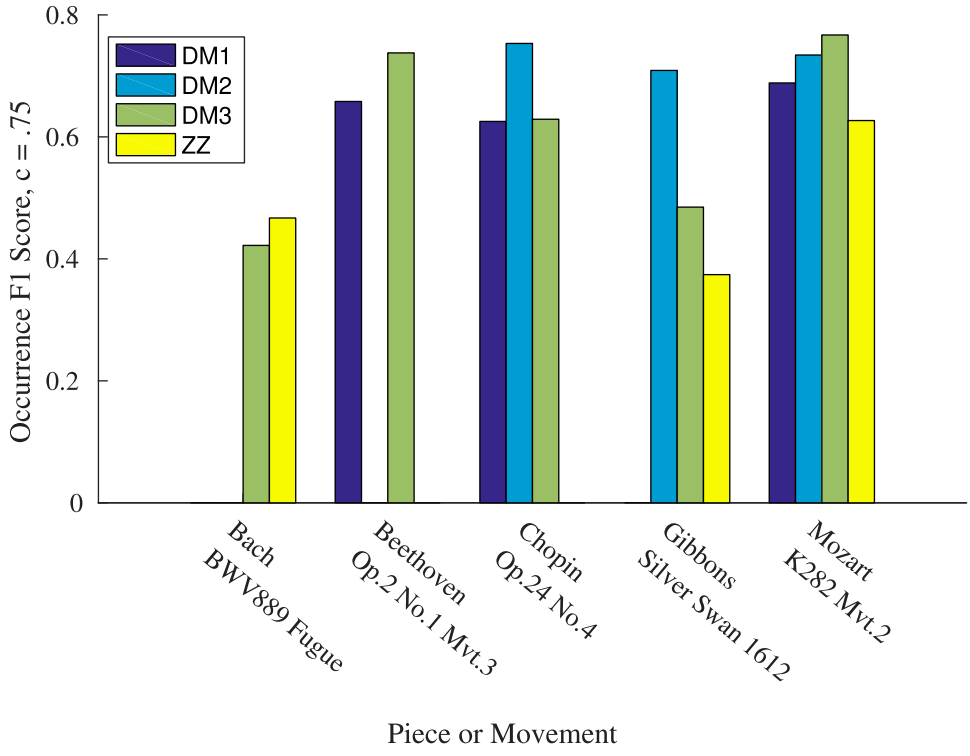

Figure 15. MIREX 2016 Discovery of Repeated Themes & Sections symPoly task, Occurrence F1 score for 2016 algorithms DM1, DM2 & DM3 with additional ZZ results. A score of 0 results from failure of an algorithm to identify at least 75% of the total instances of any pattern – no bar is plotted for these cases.

- Each set of *(start, end)* pairs is labelled following Bruhn's description, and considered a 'ground truth' unit which it is desirable for an automatic segmentation tool to identify.

Grammars are then built for each piece, and a set of *(start, end)* pairs created for each of its sub-rules in *S*, by iteratively expanding the grammar and recording the offsets of each rule occurrence within the input sequence. Where rules may be obviously grouped, such as two consecutive non-terminal symbols occurring beneath a single span specified by Bruhn, an artificial rule containing these group elements is manually added to the grammar.

A score is then greedily computed to represent match degree between the ground truth segments and the grammar's rules. For each ground truth segment, the sum of Jaccard Distance to instances of each grammar rule is calculated, and the rule at minimum distance is considered a unique match to the target segment. Where no match exists, distance is set to maximum. The overall match between a set of expert-derived segments and the grammar rules which most closely represent them is taken as the mean Jaccard Index for all such segments.

Results – The results are presented in Table 11.

Figures 16 and 17 show the hierarchy produced by ZZ for WTC I Fugue No. 2 and Prelude No. 3, in 'piano-roll' format.

Table 11. Mean Jaccard Index to Bruhn's analyses of J.S. Bach's *Das Wohltemperierte Clavier* Book I, No. 1–4.

	WTC I	
No.	Prelude	Fugue
1	0.82	0.86
2	0.87	0.95
3	0.91	0.78
4	0.66	0.62

These figures show matching typical of that returned by the experiment. Rules are selected from various levels of the hierarchy, yet generally where a human-chosen span occurs there exists a grammar rule of markedly similar length and offset, and a strong correlation between them is predominant. Each ground truth section within Fugue No. 2 almost exactly aligns with a grammar rule in the first three layers of the hierarchy, with 64% of matches occurring at the top level. Some low level rules begin with an additional symbol over the target sections, causing the reduction in accuracy. Although such symbols form part of exactly repeating sequences, they exist beyond Bruhn's boundaries, and thus outside what this experiment considers accurate against the ground truth. There are some weak results, in particular for No. 4 Prelude and Fugue.

The grammar for Fugue No. 4 exhibits good correlation, but with important exceptions. For example, several instances of the first subject are not matched to any

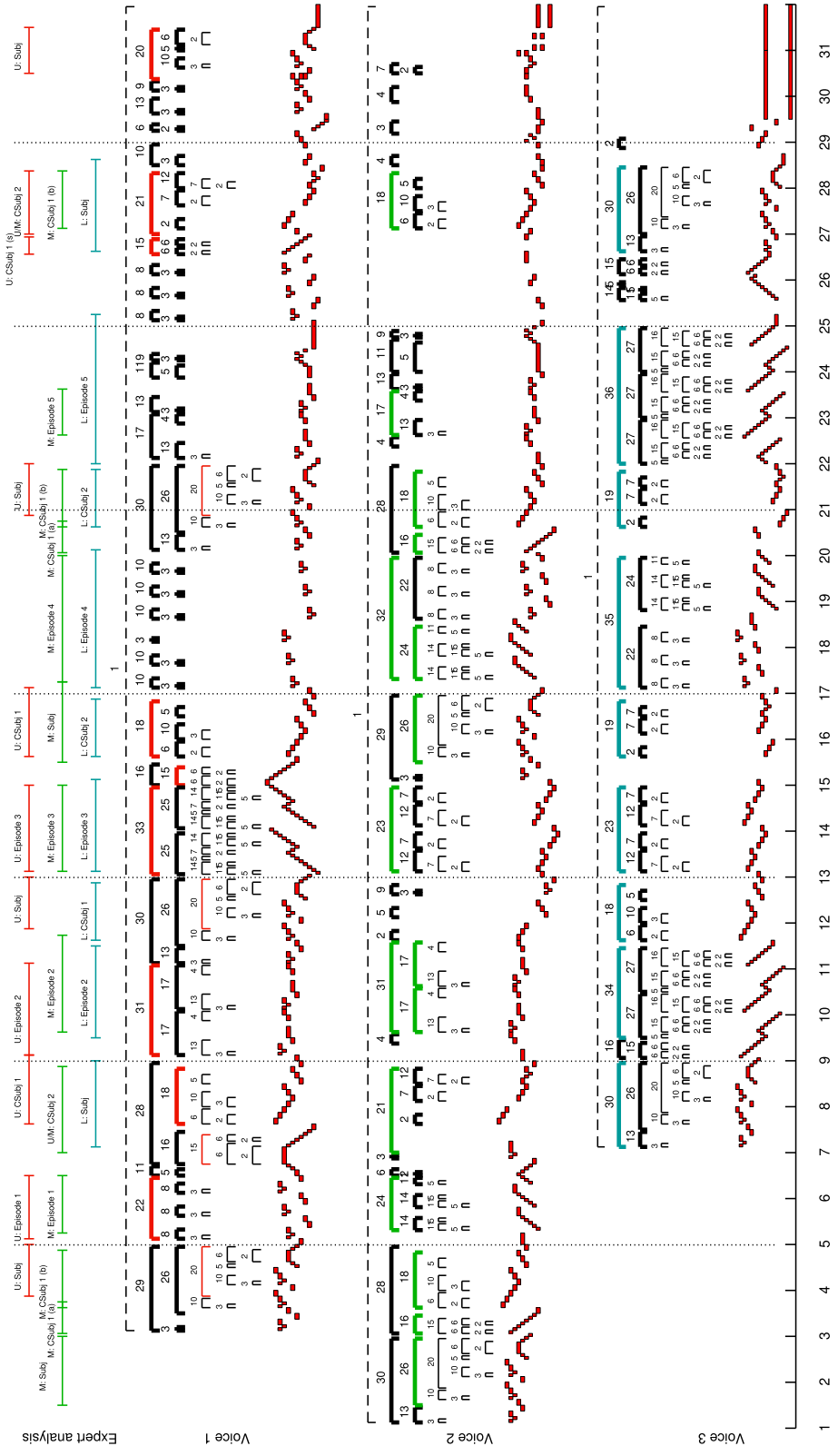

Figure 16. Comparison between musicologist-identified segments and rules within the hierarchy returned by ZZ, for Bach's Fugue No. 2 from WTC I. Jaccard Index for this piece is 0.95. Note the erroneous selection of rule 15 in voice 1, bar 15; simply selecting the rules most strongly matching the target sections can result in inclusion of non-matching segments, where the penalty for this choice does not outweigh rejection of the rule. However, both rule 15 instances do indeed match partial counter-subject motifs, suggesting that algorithmic identification of a repeat not highlighted by Bruhn has in fact occurred.

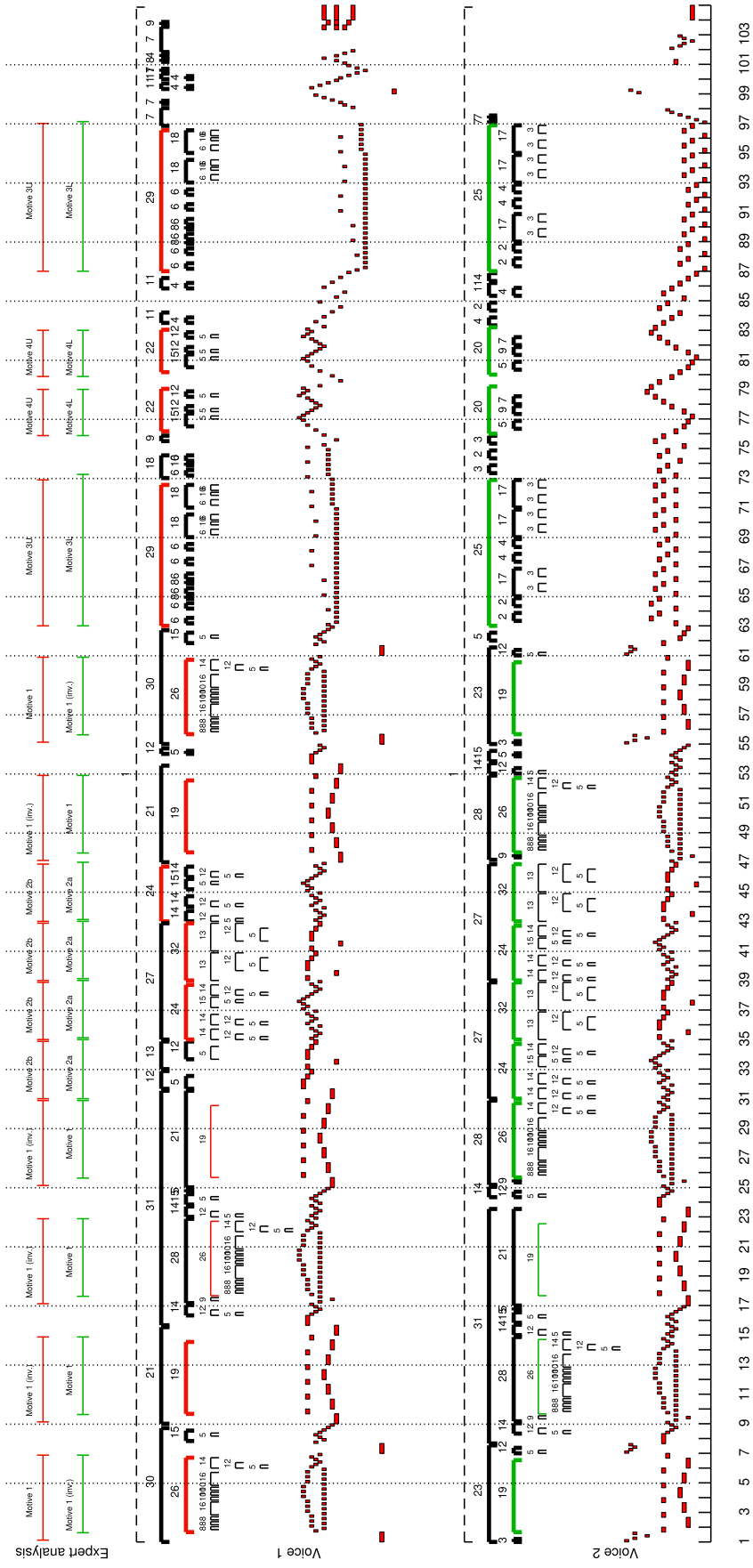

Figure 17. Comparison between musicologist-identified segments and rules within the hierarchy returned by ZZ, for Bach's Prelude No. 3 from WTC I. Jaccard Index for this piece is 0.91. Voice 3 contained only closing notes, and is omitted from this figure.

candidate rule. Two distinct circumstances explain these omissions. In some cases, intervals within the subject exposition differ, either because of a change in musical scale or by deliberate variation, such as a transitive note linking a subject to the following phrase. This shortcoming might be addressed by incorporating domain knowledge into the grammar construction process, so that sequences containing variations may also be considered as repeating. In other instances, the compressor has not chosen a rule matching a subject span; instead, a rule producing greater compression has been formed using a symbol also present within the subject, thus preventing use of an existing rule which defines the subject alone. It is conceivable this might occur because a smallest model was not produced by ZZ. However, it may be more reasonable to suggest that overlapping structural explanations exist for this passage, such as may be seen in Bruhn's analysis of Book I Prelude No. 1 (Bruhn, 1993). This is a more serious shortcoming: the branching of a grammar's hierarchy prevents the modelling of intersecting structures, which may represent important complementary explanations in a musical analysis.

Our greedy approach to choosing grammar rules does not restrict matches to a single level of the hierarchy, and the most closely correlating rules are not always those from the top level. For example, in Figure 16, rules 16 and 18 are selected together as the strongest candidates for counter-subject 1, part b (bars 3–5 and 20–22). However, rule 28 exists at a higher level and contains both chosen rules, suggesting that extension of Bruhn's segment by a single interval might be a good representation of this counter-subject instance. Mismatches such as these may indicate a failure of the model to recover a musicologically ideal segmentation, or present an alternative explanation of a score's structure. The rule hierarchy may indeed provide an advantageous view of a given segment at various levels of abstraction, and hint at the manner in which a composer employs compound motifs and techniques when creating the piece.

It is important to note the small sample presented here. Digitisation of expert analyses is time consuming and open to interpretation, and access to a wide collection of digital interpretations from various analytical schools against which to evaluate alogrithmic methods is currently unavailable – such a resource could significantly benefit similar research. Accepting this limitation, our results suggest the level of correlation between grammar rules and expert-defined segments is notable, and likely aided by the highly-structured nature of the analysed pieces, making identification of exactly-matching repeats relevant and useful.

Table 12. Changes applied to rules of the grammar modelling the chromatic intervals of J.S. Bach's Prelude No. 1 from *Das Wohltemperierte Clavier* Book I.

Rule	Instances	Original intervals	Edited intervals
2	9	3, −3	−9, 9
3	22	3, 5	12, −4
4	7	4, −4	−8, 8
6	4	−7, 4, 3	−3, −4, 7
7	3	−6, 3, 3	−3, −3, 6
9	7	0, 0, 0	0, −7, 7
10	4	3, −10, 7	−7, −2, 9
11	3	3, −8, 5	−5, −4, 9
13	4	5, 7, −12	12, −7, −5
15	2	−6, 2, 4, −6, 2	12, −4, −2, −6, −4
19	10	5, 4, −9, 5, 4	9, −4, −5, −3, 12
20	4	5, 5, −10, 5, 5	10, −5, −5, 10, 0
21	8	7, 5, −12, 7, 5	12, −5, −7, −5, 17
22	8	−10, 4, 6, −10, 4, 6	−6, −4, 10, −6, −4, 10

6.3.3. Grammar-assisted editing

To demonstrate some benefits and disadvantages of a grammar-based editing system, we developed a simple editing tool, choosing Bach's Prelude No. 1 from *Das Wohltemperierte Clavier* Book I as input since it has a clearly-defined structure against which alterations can be easily distinguished. A single grammar is built for this piece, and its non-*S* rules are altered with the intention of producing a musically reasonable output. In this case, we attempt to simply reverse the ascending motif used throughout. Finally, the grammar is expanded to produce edited score data, which may be represented visually or played.

Edit operations are restricted to the substitution of individual terminal symbols within rules representing pitch intervals, to demonstrate the effect of changes to individual pitch values only, and avoid the alterations in structure which are likely to occur if non-terminal symbols are changed. Edits were further constrained so that the sum of each rule remains unchanged, to avoid introduction of a pitch offset for subsequent notes.

Results – The result of the editing process is shown in Figure 18.

Terminal symbols in 14 of the 29 rules of the pitch grammar are changed; Table 12 provides details of the edits made.

The frequencies with which rules 2, 3 and 19 occur in the expanded score are high, indicating these are important foundational elements. Indeed, altering them has a strong effect across the entire piece, highlighting that many related changes may be made simultaneously using this method. As a human editor, care must be taken to preserve the context of each rule. For example, a rule containing only scale degrees 1 and 5 may appear in a major and minor setting, and introduction of a 3rd can cause

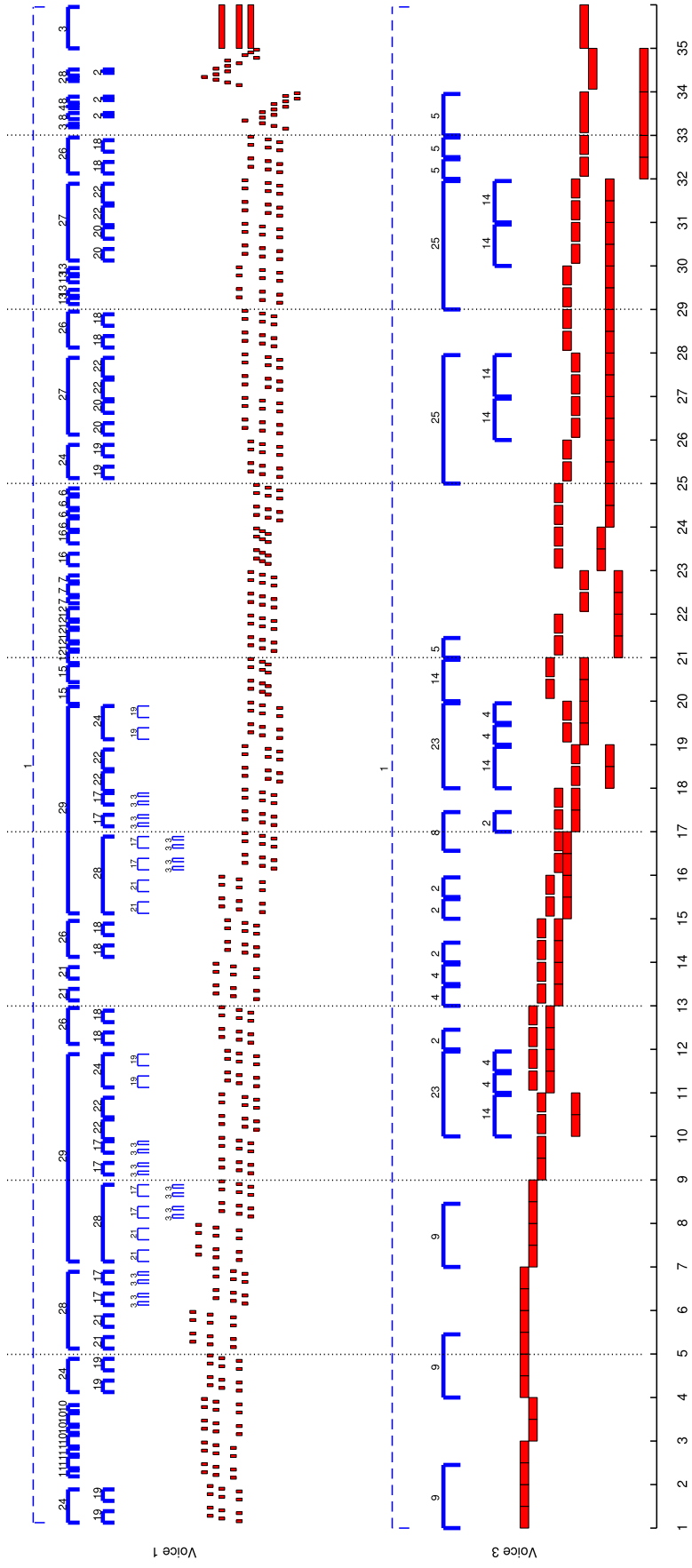

Figure 18. Structure of grammar for Bach's Prelude No. 1 from WTC I, built from chromatic intervals by ZZ.

dissonance where the rule coincides with the opposite scale. Auditioning after each rule is edited will allow a manual check to be made, and undesirable changes to be reversed.

These simple edits allow fast production of a believable score, where the goal of reversing ascending figures is mostly achieved. However, for two-symbol rules such as 2 and 3 this is not possible, since only one pitch change may

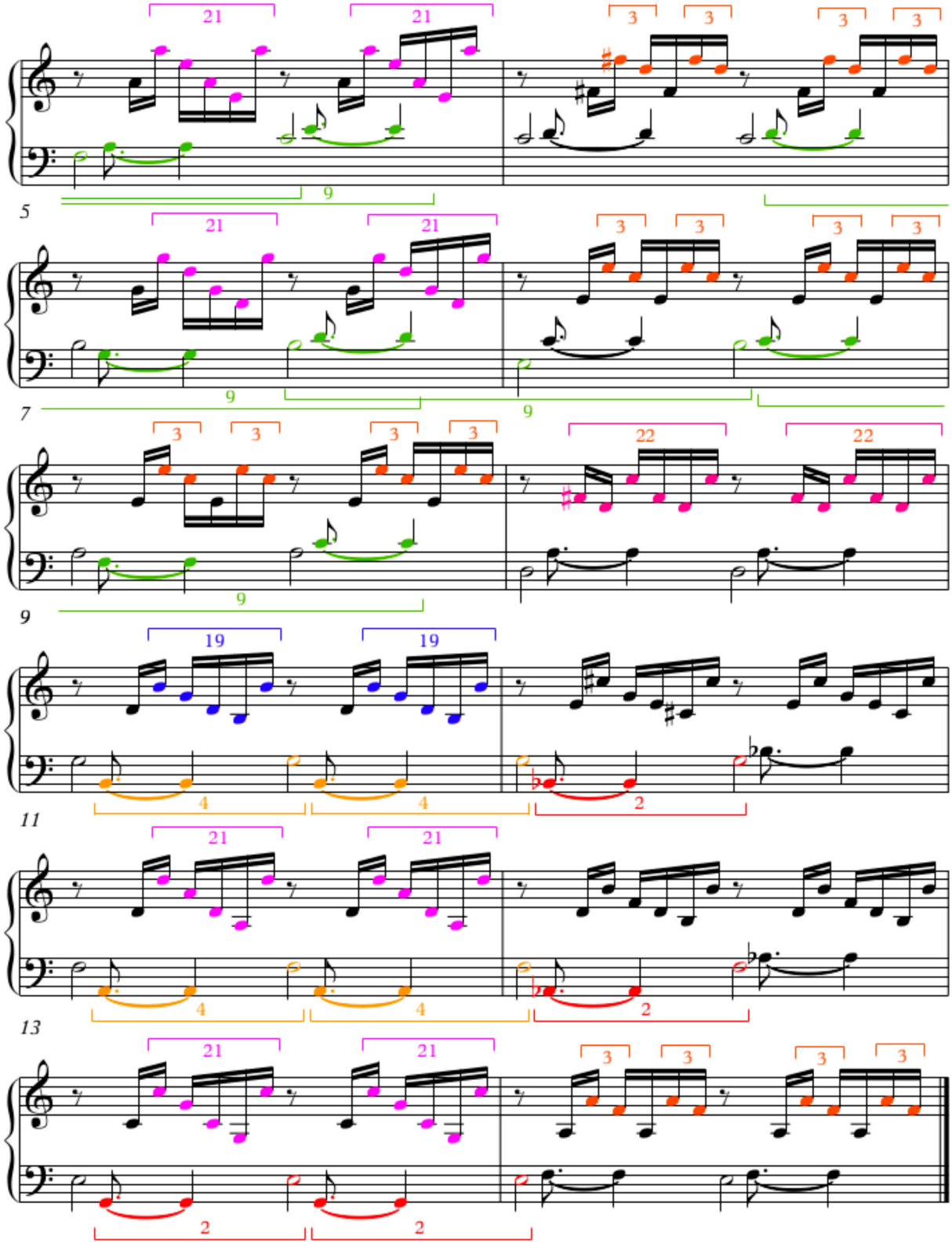

Figure 19. Bars 5-16 of Bach's Prelude No. 1 from WTC I after grammar-assisted editing.

be made before returning to the original note; in these cases, inversion of the motion was chosen. Enforcing the constraint that the rule's sum must not alter for rule 20 also requires a compromise: its final interval cannot be a negative value, which breaks the descending pattern it otherwise maintains. Rule 9 represents four identical bass notes, and a 5th was substituted for the third note in the sequence to show additional movement is easily possible whenever the change does not violate the context of the rule's occurrence. A segment of the resulting score is shown in Figure 19.

Several disadvantages to this approach can be seen. Since pitch and rhythm are represented separately as sequences in *S*, insertion or deletion of a single element results in misalignment between note attributes, and will occur as many times as the rule is used during grammar expansion. Encoding voices in separate sequences as described in Section 3.3 does not support the modelling of harmonic relationships such as chords, and where related notes are edited these relationships must be manually preserved. The ability to make multiple changes from editing a single symbol may also result in dissonant combinations which are not immediately obvious, as a single rule can exist in several different musical contexts. Care must be taken not to allow the effect of altering a rule to impact subsequent rules and values, especially where interval-based data representations are chosen.

However, grammar assisted editing naturally enables changes specifically relating to content or structure to be made, either individually as in this experiment, or in combination. Where the musical context of a rule is known, for example the scale its pitch values belong to, it may be altered to contain anything within that context. Large-scale modifications to a score are possible via rule editing, from alteration of low-level building blocks containing only terminal symbols to strong structural changes through manipulation of rules containing a deep hierarchy. Binding information such as pitch, onset and duration together symbolically could help address some of the disadvantages affecting such operations. Since, as shown in the previous experiments, a grammar's rules may represent a musically significant segmentation of a piece, it is reasonable to suggest our method allows edits to occur within a contextual, musical framework.

7. Conclusions

In this paper, we have investigated the application of grammar-based compressors to six practical musical applications, comparing their performance to that achieved by the use of other popular compression algorithms. We have examined the responsiveness of each method to errors, and measured their performance when applied to location of errors, classification of folk music by tune family, and discovery of expert-defined musicological patterns.

When tasked with detection of a common transcription error, LZW proved most responsive for pieces too small to compress by standard grammar, but beyond this margin ZZ proved most sensitive. All methods showed a logarithmic response to an increasing number of errors, with GZIP outperforming ZZ as input length became significant. ZZ was most successful in correctly identifying the position of a single error, with an F-measure of 0.22−0.35. Strong variation in response was measured for all methods, showing none can be relied upon to respond correctly in each individual case. However, every method generated a larger model in the majority of cases.

When grouping pieces from the Meertens Tune Collections by tune family, ZZ was able to perform moderately by comparison with existing studies, when the results from multiple representations were weighted and used in nearest-neighbour classification. We also applied ZZ to the discovery of expert-defined patterns from the polyphonic *Discovery of Repeated Themes & Sections* task presented in MIREX 2016, where it bettered all submitted methods for a highly-structured Bach fugue, but gave unstable results for the other four scores, highlighting the greater flexibility of SIATEC-based algorithms in discovering inexactly repeating patterns. Finally, we compared exactly repeating structures identified by musicologist Siglind Bruhn within eight works from Bach's *Das Wohltemperierte Clavier* Book I to rules within grammar-based compressors produced by ZZ for each piece. Although results showed wide variation, strong correlation existed at high levels of the hierarchy, a notable achievement considering that ZZ possesses no domain knowledge.

In conclusion, our results generally support the link between strength of compression and the information recovered, as suggested by the Minimum Description Length principle. We have shown that ZZ can outperform several popular compressors when applied to detect degradation in musical structure and classification of Dutch folk tunes, in the latter case when provided with attribute-rich note data. However, exact grammars cannot rival current techniques when seeking expert-defined patterns containing variations, and can fail to generate desirable rules where an overlapping explanation, and therefore rule, exists. Our findings highlight the significance of intersection in the analysis of musical compositions, and support suggestions by existing studies that the ability to abstract musical features and pattern templates using domain knowledge is important to algorithmic analysis; such additions are likely to improve the performance of ZZ on these applications. Even without

such enhancements, our results demonstrate that grammars are a tool that perform at a useful level in the field of music analysis.

The data that support the findings of this study are openly available from Cardiff University via URL http://doi.org/10.17035/d.2020.0098047203.

8. Future work

A number of possibilities exist for further development of this work, and we suggest the following directions:

- An heuristic could be designed which retains the sensitivity of grammar-based compressors to errors (degraded musical structure), but does not require exhaustive exploration of the search space. Such an heuristic might select only patterns which, when altered, allow significantly increased compression, or use high-level abstraction to structure the search and terminate branches unlikely to result in improvement on subsequent iterations. This may allow error detection by grammar to become a practical option.
- Investigation into reduction of the variation observed with increasing number of errors could be carried out, in an attempt to smooth and enhance error response, and to isolate false-negative conditions. Potentially, input pre-processing to higher level structures may produce a more consistent compressor response.
- From the observation that combining representations offers an accuracy gain when classifying the MTC, grammar-based compressors may be combined with other compression algorithms, such as those studied here, to produce a weighted output. Leveraging significant properties of individual compressors in this manner is likely to result in an overall improvement to the applications presented.
- Grammars may be augmented with parametric structures providing greater potential for modelling sequences that are repeated with variations. Addition of simple flexible matching, in a form where encoding overhead does not prohibit its use, is likely to increase tune family classification success rate. Various matching schemes could be tested, from simple alignment- or distance-based transforms to those incorporating domain knowledge, such as motif, rhythm or phrase templates. Existing feature-centred techniques, such as facets (Stober, 2011), viewpoints (Conklin, 2013; Goienetxea et al., 2016), or form definitions such as described by Giraud and Staworko (2015), might be added to the construction process, and direct comparison made against these studies. We hypothesise that more compact grammars would be possible, resulting in improved performance for musical applications.
- We suggest a large-scale, methodical musicological study of the analysis and segmentation of a large corpus of digital scores would be a benefit to the research community. Ground truth, hierarchical definitions from various schools of analysis, including scale annotations, transformations and transitions might be used to develop more powerful and accurate algorithms for score compression and processing. We observed additional notes prefixed to expert-identified segments in some cases; evaluation of how musically admissible such extensions may be is possible given expert consensus. Potentially, generalised grammars capable of covering a given form or style might be programmatically generated and used to optimise compression.

Acknowledgments

We also thank Prof. Meredith for his kind correspondence and encouragement.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

We gratefully acknowledge the support of the *Supercomputing Wales* project, which is part-funded by the European Regional Development Fund (ERDF) via Welsh Government, and whose facilities were used throughout. David Humphreys would also like to thank Cardiff University, UK, for fully funding three years of his PhD via stipend, without which this study would not have been possible.

References

Abdallah, S., Gold, N., & Marsden, A. (2016). Analysing symbolic music with probabilistic grammars. In *Computational music analysis* (pp. 157–189). Springer.

Apostolico, A., & Galil, Z. (2013). *Combinatorial algorithms on words* (Vol. 12). Springer Science & Business Media.

Avid Technology Inc (2011). *Sibelius 7: Using the manuscript language*. Retrieved September 21, 2018, from http://resources.avid.com/SupportFiles/Sibelius/ManuScript_Language_Guide_2018.6.pdf

Bardera, A., Feixas, M., Boada, I., & Sbert, M. (2006). *Compression-based image registration*. In 2006 IEEE international symposium on information theory (pp. 436–440). IEEE.

Barlow, H., Morgenstern, S., & Erskine, J. (1948). *A dictionary of musical themes*. Crown.

Benz, F., & Kötzing, T. (2013). *An effective heuristic for the smallest grammar problem*. In Proceedings of the 15th annual conference on genetic and evolutionary computation (pp. 487–494). ACM.

Boot, P., Volk, A., & de Haas, W. B. (2016). Evaluating the role of repeated patterns in folk song classification and compression. *Journal of New Music Research*, *45*(3), 223–238. https://doi.org/10.1080/09298215.2016.1208666

Bruhn, S. (1993). *JS Bach's well-tempered clavier: In-depth analysis and interpretation*. Mainer.

Burrows, M., & Wheeler, D. J. (1994). *A block-sorting lossless data compression algorithm* (SRC Research Report 124).

Callon, G. J. (1998–2009). *Acadia early music archive*. Retrieved September 20, 2018, from http://www.acadiau.ca/ ∼ gcallon/www/archive/

Cambouropoulos, E. (2001). *The local boundary detection model (LBDM) and its application in the study of expressive timing*. In ICMC (p. 8). ICMC, Instituto Cubano de la Musica.

Cambouropoulos, E. (2006). Musical parallelism and melodic segmentation: A computational approach. *Music Perception*, *23*(3), 249–268. https://doi.org/10.1525/mp.2006.23.3.249

Cambouropoulos, E., Crawford, T., & Iliopoulos, C. S. (2001). Pattern processing in melodic sequences: Challenges, caveats and prospects. *Computers and the Humanities*, *35*(1), 9–21. https://doi.org/10.1023/A:1002646129893

Carrascosa, R., Coste, F., Gallé, M., & Infante-Lopez, G. (2010). Choosing word occurrences for the smallest grammar problem. In *International conference on language and automata theory and applications* (pp. 154–165). Springer.

Carrascosa, R., Coste, F., Gallé, M., & Infante-Lopez, G. (2011). The smallest grammar problem as constituents choice and minimal grammar parsing. *Algorithms*, *4*(4), 262–284. https://doi.org/10.3390/a4040262

Carrascosa, R., Coste, F., Gallé, M., & Infante-Lopez, G. (2012). Searching for smallest grammars on large sequences and application to DNA. *Journal of Discrete Algorithms*, *11*, 62–72. https://doi.org/10.1016/j.jda.2011.04.006

Chambers, J. (2015). *John Chambers' clone of the O'Neill's project files and web pages*. Retrieved September 25, 2018, from http://trillian.mit.edu/ ∼ jc/music/book/oneills/1850

Choi, K., Hao, Y., X. L. Zhang, Dzhambazov, G., & Collins, T. (2018). *Music Information Retrieval Evaluation eXchange (MIREX) home*. Retrieved September 30, 2018, from http://www.music-ir.org/mirex/wiki/MIREX_HOME

Chomsky, N. (1956). Three models for the description of language. *IEEE Transactions on Information Theory*, *2*(3), 113–124. https://doi.org/10.1109/TIT.1956.1056813

Cilibrasi, R., & Vitányi, P. M. B. (2005). Clustering by compression. *IEEE Transactions on Information Theory*, *51*(4), 1523–1545. https://doi.org/10.1109/TIT.2005.844059

Cilibrasi, R., Vitányi, P. M. B., & de Wolf, R. (2004). Algorithmic clustering of music based on string compression. *Computer Music Journal*, *28*(4), 49–67. https://doi.org/10.1162/0148926042728449

Cohen, A. R., Gomes, F. L. A. F., Roysam, B., & Cayouette, M. (2010). Computational prediction of neural progenitor cell fates. *Nature Methods*, *7*(3), 213–218. https://doi.org/10.1038/nmeth.1424

Conklin, D. (2013). *Fusion functions for multiple viewpoints*. In Proceedings of the 6th international workshop on machine learning and music. MML.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

CPDL organisation (2018). *Choral public domain library*. Retrieved September 20, 2018, from http://www2.cpdl.org/

Cuthbert, M., & Ariza, C. (2010, January). *Music21: A toolkit for computer-aided musicology and symbolic music data*. In ISMIR (pp. 637–642). ISMIR.

Deutsch, L. P. (1996, May). *GZIP file format specification version 4.3* (No. 1952). RFC 1952. RFC Editor. Retrieved September 23, 2018, from https://rfc-editor.org/rfc/rfc1952.txt

Friberg, A., Bresin, R., Frydén, L., & Sundberg, J. (1998). Musical punctuation on the microlevel: Automatic identification and performance of small melodic units. *Journal of New Music Research*, *27*(3), 271–292. https://doi.org/10.1080/09298219808570749

Gallé, M. (2011). Searching for Compact Hierarchical Structures in DNA by means of the Smallest Grammar Problem (Unpublished doctoral dissertation). Université Rennes 1.

Giraud, M., & Staworko, S. (2015). Modeling musical structure with parametric grammars. In *International conference on mathematics and computation in music* (pp. 85–96). Springer.

Goienetxea, I., Neubarth, K., & Conklin, D. (2016). *Melody classification with pattern covering*. In 9th International workshop on music and machine learning (MML 2016). MML.

Gritten, A., & King, E. (2006). *Music and gesture*. Ashgate Publishing, Ltd.

Hillewaere, R., Manderick, B., & Conklin, D. (2014). Alignment methods for folk tune classification. In *Data analysis, machine learning and knowledge discovery* (pp. 369–377). Springer.

Howe, E. (1883). Ryan's mammoth collection: 1050 reels and jigs, hornpipes, clogs, walk-arounds, essences, strathspeys, highland flings and contra dances, with figures, and how to play them; Bowing and fingering marked, together with forty introductory studies for the violin, with explanations of bowing, etc. *New Hibernia Review/Iris Éireannach Nua*, *6*(3), 9–22.

Johannes Kepler University (2013). *JKU patterns development database* (August 2013 version). Retrieved September 25, 2018, from http://tomcollinsresearch.net/research/data/mirex/JKUPDD-Aug2013.zip

Kanamori, K., Hamanaka, M., & Hoshino, J. (2014). *Method to detect GTTM local grouping boundaries based on clustering and statistical learning*. In ICMC. ICMC.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In*Proceedings of the 14th international joint conference on artificial intelligence* (Vol. 2). Morgan Kaufmann Publishers Inc.

Kolmogorov, A. N. (1963). On tables of random numbers. *Sankhyā: The Indian Journal of Statistics, Series A*, 25(4), 369–376.

Lehman, E., & Shelat, A. (2002). *Approximation algorithms for grammar-based compression*. In Proceedings of the thirteenth annual ACM-SIAM symposium on discrete algorithms (pp. 205–212). Society for Industrial and Applied Mathematics.

Lerch, A., Arthur, C., Pati, A., & Gururani, S. (2019). *Music performance analysis: A survey*. Preprint, arXiv:1907.00178

Lerdahl, F., & Jackendoff, R. (1983). *A generative theory of tonal music*. MIT Press.

Li, M., Chen, X., Li, X., Ma, B., & Vitányi, P. M. B. (2004). The similarity metric. *IEEE Transactions on Information Theory*, *50*(12), 3250–3264. https://doi.org/10.1109/TIT.2004.838101

Louboutin, C., & Meredith, D. (2016). Using general-purpose compression algorithms for music analysis. *Journal of New Music Research*, *45*(1), 1–16. https://doi.org/10.1080/09298215.2015.1133656

Meertens Instituut (2018). *The Meertens tune collections: MTC-ANN-2.0.1*. Retrieved September 25, 2018, from http://www.liederenbank.nl/mtc/

Meredith, D. (2006). The ps13 pitch spelling algorithm. *Journal of New Music Research*, *35*(2), 121–159. https://doi.org/10.1080/09298210600834961

Meredith, D. (2013). *COSIATEC, 2013 MIREX competition on discovery of repeated themes and sections submission*. Retrieved September 23, 2018, from http://titanmusic.com/software/mirex2013/Mirex2013CosiatecRaw.zip

Meredith, D. (2014). *Using point-set compression to classify folk songs*. In International workshop on folk music analysis (pp. 29–35). Computer Engineering Department, Boğaziçi University.

Meredith, D., Lemström, K., & Wiggins, G. A. (2002). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, *31*(4), 321–345. https://doi.org/10.1076/jnmr.31.4.321.14162

MIDI Association (1996). *MIDI 1.0 detailed specifications*. Retrieved September 28, 2018, from https://www.midi.org/specifications-old/item/the-midi-1-0-specification

Musopen organisation (2018). *Musopen – free sheet music, royalty free music, and public domain resources*. Retrieved September 20, 2018, from https://musopen.org/

Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, *48*(3), 443–453. https://doi.org/10.1016/0022-2836(70)90057-4

Nevill-Manning, C. G., & Witten, I. H. (1997). Identifying hierarchical structure in sequences: A linear-time algorithm. *Journal of Artificial Intelligence Research*, *7*, 67–82. https://doi.org/10.1613/jair.374

Nykter, M., Price, N. D., Larjo, A., Aho, T., Kauffman, S. A., Yli-Harja, O., & Shmulevich, I. (2008). Critical networks exhibit maximal information diversity in structure-dynamics relationships. *Physical Review Letters*, *100*(5), Article ID 058702. https://doi.org/10.1103/PhysRevLett.100.058702

Pearce, M. T., Müllensiefen, D., & Wiggins, G. A. (2008). *A comparison of statistical and rule-based models of melodic segmentation*. In ISMIR (pp. 89–94). ISMIR.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*(5), 465–471. https://doi.org/10.1016/0005-1098(78)90005-5

Sapp, C. S. (2005). *Online database of scores in the humdrum file format*. In ISMIR (pp. 664–665). ISMIR.

Savakis, A. E. (2000). *Evaluation of lossless compression methods for gray scale document images*. In 2000 International conference on image processing, 2000. Proceedings (Vol. 1, pp. 136–139). IEEE.

Schaffrath, H., & Huron, D. (1995). *The Essen Folksong collection in the Humdrum Kern format*. Center for Computer Assisted Research in the Humanities.

Schenker, H., & Salzer, F. (1969). *Five graphic music analyses (F nf Urlinie-Tafeln)*. Courier Corporation.

Schoenberg, A., Stein, L., & Strang, G. (1967). *Fundamentals of musical composition*. Faber & Faber.

Senthil, S., & Robert, L. (2011). Text compression algorithms - a comparative study. *ICTACT Journal on Communication Technology*, *2*(4), 444–451. http://doi.org/10.21917/ijct.2011.0062

Seward, J. (1996). *BZIP2 and LibBZIP2*. Retrieved September 23, 2018, from http://www.bzip.org/

Sidorov, K. A., Jones, A., & Marshall, A. D. (2014). *Music analysis as a smallest grammar problem*. In ISMIR (pp. 301–306). ISMIR.

Siyari, P., & Gallé, M. (2017). *The generalized smallest grammar problem*. In International conference on grammatical inference (pp. 79–92). PMLR.

Steedman, M. J. (1984). A generative grammar for jazz chord sequences. *Music Perception: An Interdisciplinary Journal*, *2*(1), 52–77. https://doi.org/10.2307/40285282

Stober, S. (2011). *Adaptive distance measures for exploration and structuring of music collections*. In Audio engineering society conference: 42nd international conference: Semantic audio. AES.

Student. (1908). The probable error of a mean. *Biometrika*, *6*(1), 1–25. https://doi.org/10.1093/biomet/6.1.1

Temperley, D. (2004). *The cognition of basic musical structures*. MIT press.

van Kranenburg, P., Janssen, B., & Volk, A. (2016). *The meertens tune collections: The annotated corpus (MTC-ANN) versions 1.1 and 2.0.1* (Vol. 2016, No. 1). Meertens instituut KNAW.

van Kranenburg, P., Volk, A., & Wiering, F. (2013). A comparison between global and local features for computational classification of folk song melodies. *Journal of New Music Research*, *42*(1), 1–18. https://doi.org/10.1080/09298215.2012.718790

Velarde, G., Weyde, T., & Meredith, D. (2013). An approach to melodic segmentation and classification based on filtering with the Haar-Wavelet. *Journal of New Music Research*, *42*(4), 325–345. https://doi.org/10.1080/09298215.2013.841713

Volk, A., & Van Kranenburg, P. (2012). Melodic similarity among folk songs: An annotation study on similarity-based categorization in music. *Musicae Scientiae*, *16*(3), 317–339. https://doi.org/10.1177/1029864912448329

Welch, T. A. (1984). Technique for high-performance data compression. *Computer*, *17*(6), 8–19. https://doi.org/10.1109/MC.1984.1659158

Ziv, J., & Lempel, A. (1977). A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, *23*(3), 337–343. https://doi.org/10.1109/TIT.1977.1055714

Ziv, J., & Lempel, A. (1978). Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, *24*(5), 530–536. https://doi.org/10.1109/TIT.1978.1055934