

Beyond Bars: Distribution of Edit Operations in Historical Prints

Music & Science

Volume 9: 1–19

© The Author(s) 2026

DOI: 10.1177/20592043261470534

journals.sagepub.com/home/mns



Adrian Nachtwey¹ , Fabian C. Moss² and Anna Viktoria Katrin Plaksin¹

Abstract

A persistent challenge for musicological corpus studies is the scarcity of existing, high-quality corpora for computational research. Here, we propose a method to selectively encode musical scores based on a specific sampling technique and illustrate the approach using a corpus of piano works, namely Beethoven's Bagatelles op. 33. In a case study, we perform a phylogenetic analysis on six editions of the Bagatelles. We first analyze the full-length encodings and then sampled subsets of bars and compare the results. We evaluate three sampling approaches to create the subsets, each approach based on distinct assumptions about the distribution of differences between musical prints. The results show that sampled subsets can approximate the overall number of differences with high accuracy. In particular, random sampling performs as well as, or better than, more constrained approaches. These findings show that sampling methods are a viable option to enable quantitative musicological analyses, for example on historical editorial practices, using phylogenetic approaches.

Keywords

Computational musicology, corpus studies, editorial practice, music digitization, sampling

Received: August 13, 2025; accepted: July 2, 2026

Action Editor: Mark Gotham, Kings College London.

Peer Review: Mark Saccomano, Paderborn University. One anonymous reviewer.

Introduction

In recent years, musicological research based on large datasets, so called corpus studies, has gained popularity. Researchers have engaged in producing rich, high-quality datasets of music scores for computational analysis, for example annotated training data for audio-chord-recognition algorithms (Burgoyne et al. 2011), sets of piano variations by Mozart and Beethoven including harmonic annotations (Devaney et al. 2015), encoded scores with analytic annotations (Hentschel et al. 2025), and harmonic annotations of Beethoven's string quartets (Neuwirth et al. 2018). However, musicological corpus studies often face the problem of acquiring their data (see Lewis et al. 2023). Optical Music Recognition (OMR) techniques are, despite recent advances (see Mayer et al. 2024; Ríos-Vila et al. 2023, 2024), limited in terms of the quality of their output or their scope of application (for a general state of the art see Calvo-Zaragoza et al. 2023). Therefore, they are rarely a viable option for automatically encoding the material, and musical texts have to be encoded manually in time-consuming work.

Consequently, musicologists often rely on already existing corpora, but this also causes problems. First, they either

have to fit the research question of another project by chance, or researchers have to make their questions fit the corpora. This makes them only useful for a handful of projects. Additionally, this reinforces the use of music by already popular composers and musicians, thereby preventing more diverse musicological research.

Moreover, corpus studies such as those mentioned above typically only require a single encoding for each of multiple compositions. Phylogenetic analyses (Windram et al. 2014, 2022), in contrast, rely on particularly special corpora since they need multiple encodings of the same pieces in different versions, such as different prints. Phylogenetic analyses compare these versions to identify differences, thereby establishing

¹KreativInstitut.OWL, Paderborn University, Detmold, Germany

²Institut für Musikforschung, Julius-Maximilians-Universität Würzburg, Würzburg, Germany

Corresponding Author:

Adrian Nachtwey, KreativInstitut.OWL, Paderborn University, Detmold, Germany.

Email: adrian.nachtwey@uni-paderborn.de



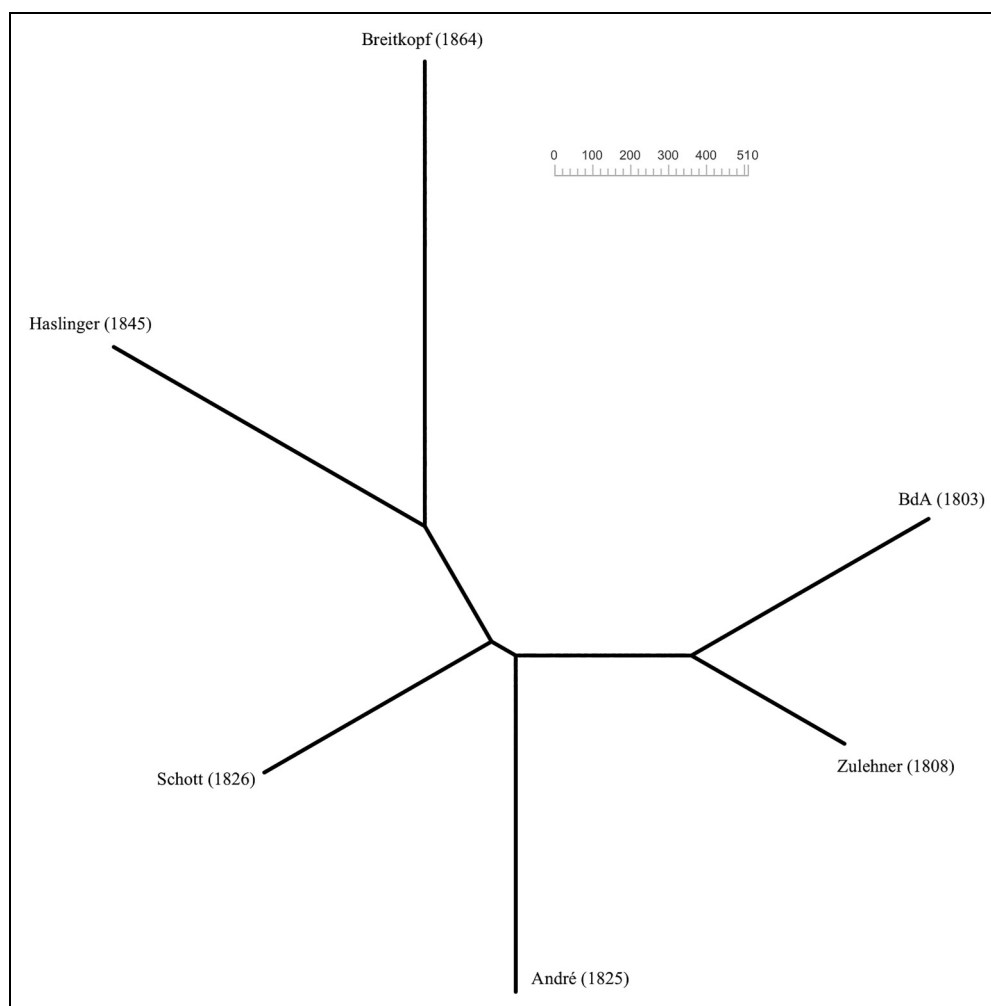


Figure 1. Neighbor-joining tree of the total differences between the full-length encodings of six editions of Beethoven's Bagatelles Op. 33, created using SplitsTree (Huson and Bryant 2024).

connections between them. The results obtained are trees resembling the stemmata created by traditional philological work (see Figure 1). Given these particular requirements, the reuse of pre-existing corpora is not a viable option in this instance.

To reduce the encoding workload, Saccomano et al. (2024) propose sampling as a solution. However, the authors' aim is to present methods for capturing information about the sampled parts and the gaps in an encoding to make the data useful for further research. They do not address how sampling can be applied in specific research contexts.

In this article, we demonstrate how sampling can be used in the context of the phylogenetic analysis of historical music editions. We focus especially on the question of how to create representative samples. The corpus of this case study consists of six editions of Ludwig van Beethoven's Bagatelles Op. 33. The objective is to conduct a phylogenetic analysis of these editions without having to encode all of them in full length. Instead, we sample individual bars from the

pieces. It is crucial that the sampled bars are representative with respect to the differences between the editions. The goal of this study therefore is to find the best way of sampling individual bars from a piece while ensuring that the sample is suited for the phylogenetic analysis of multiple of its editions.

Research Question and Hypothesis

The research question (RQ) of this article is:

RQ: Is it possible to conduct statistically reliable phylogenetic corpus studies while only encoding parts of the corpus?

The goal is to identify a method for drawing samples that are representative of the editions with respect to the differences between them. To this end, we evaluate three sampling methods, each reflecting different assumptions about the distribution of differences in our corpus.

We expect the differences between the scores to fall into two categories. The first category comprises differences that

occur because of mistakes made by the editors or engravers. These should be *local*, because they only affect individual musical symbols. Examples are wrong pitches or missing symbols. In addition there should be differences that affect groups of elements due to editors' choices. For example, changing the stem direction of the first note of a group of 16th-notes affects the stem direction of the following notes, too. These are *propagating* differences. We believe that differences from the second category have a greater impact on the overall number of differences, because different editors make different choices. In addition we assume that the probability of finding differences in a bar is correlated to the number of musical symbols it contains. If a bar contains a comparatively large number of musical symbols the probability of finding differences in this bar should be statistically higher. For drawing samples representative of the number of differences between several sources this means that the number of musical symbols in a piece, rather than the bar, is the relevant unit. Our working hypothesis therefore is:

Hypothesis: A sampling algorithm that considers the number of musical symbols in a bar will be more reliable in drawing representative samples than an algorithm that selects bars at random.

Differences between Music Encodings

First, we need to define what constitutes a difference in this study and how differences are counted. We are interested in all visual differences between two prints, regardless of their impact on performances of the work. An example of various differences is shown in Figure 2. In this example, the allocation of the notes to the two staves differs between the edition of Breitkopf & Härtel and Schott, as does the placement of slurs and staccato dots (above the notes in the example on the left, below on the right). Also, the numbers indicating triplets are omitted in the Schott edition, and the two editions use different symbols to represent quarter-note rests.

All these differences have no impact on the semantics of the musical text, that is, the performance would be the same regardless of which print is used. Still, these differences can provide clues about the connection between three or more sources: If a third print represents the music in the same way as the Schott edition, it is more likely that it relied on the same sources or even the Schott edition itself as a template than that it drew on the Breitkopf edition.

Besides these merely visual differences there are also semantic differences between musical prints, such as missing clefs or accidentals. Regardless of their impact on performances or the correctness of the musical text, all differences are weighed equally in our study.

Sometimes the prints are ambiguous. Sometimes the position of a musical symbol is arbitrary. For example, the exact starting and end point of hairpins can be unclear because the two voices are not perfectly aligned. Also, the difference between staccato dots and wedges is not obvious:

how “long” may a dot be before it becomes a wedge? However, these are difficulties every musicologist has to deal with when collating musical sources. We did not rely on a reference edition to resolve such ambiguities but encoded each case based solely on internal evidence within the edition at hand.

Method

Data

As a case study, we selected relatively short pieces by Beethoven, namely the seven Bagatelles Op. 33. These shorter works were chosen because they can easily be encoded in full length.

Our test corpus comprises the Bagatelles Op. 33 in the following six editions. The publication dates are rough estimations as provided in the description of the editions in the Beethoven-Haus catalog (editions 1–5) or the Petrucci Library (edition 6), respectively.

1. the first print by *Bureau d'Arts [sic!] et d'Industrie* (Vienna 1803, hereinafter: 'BdA'),²
2. prints by Zulehner (Mainz 1808),³
3. André (Offenbach a.M. 1825),⁴
4. Schott (Mainz 1826),⁵
5. Haslinger (Vienna 1845, 'Hasl.'),⁶ and
6. Breitkopf & Härtel (Leipzig 1864, 'Breitk.').⁷

We used the MEI encoding of this Op. 33 by Goebel (2021)⁸ as a basis for our work. All scripts and full-length encodings are available as a Docker Image on Zenodo.⁹

Sampling

The data generation model in this study can be illustrated as follows. Suppose I opaque jars each contain marbles of different colors. Each jar contains only marbles of one color. Each color represents a specific type of difference observed between two musical prints. For example, one color would represent the differing allocation of notes on the staves, another the changed placement of slurs, etc. The number of marbles corresponds to the frequency with which a specific type of difference occurs in the comparison of two prints.

Let the number of marbles in a jar be n_i and $N = \sum_i n_i$ denote the total number of marbles. Each jar may contain a different number of marbles. We then randomly draw $\tilde{n}_i$ marbles from jar i until a sample of size S is obtained. The condition of a sample being representative of the jars now means that the distribution of marbles of different colors in S is the same as in the jars, in other words, the ratios $\frac{\tilde{n}_i}{S}$ and $\frac{n_i}{N}$ are approximately equal.

There are several options for drawing the samples, for example, drawing equally from all jars, assuming that



Figure 2. Example for differences between the editions of the 5th Bagatelle, bars 25 and 26, by Breitkopf & Härtel and Schott.¹ They differ in the placement of the right hand melody, of the articulations (slur and staccato) of this melody, and of the numbers to indicate triplets. Also, different symbols for quarter-note rests are used.

differences are evenly distributed. Alternatively, if we believe that one color (or kind of difference) occurs more often (i.e., there are more marbles of one color than of the others), we could bias our sampling process so that it is more likely to draw from one of the jars than from another. In this case study the distribution of marbles in the jars corresponds to the distribution of differences in the scores, and the three sampling methods under investigation represent three assumptions about this distribution.

Comparing Music Encodings

We created six files with Goebli's MEI encoding and modified each file to match one of the six editions under investigation. We then compared the editions to each other using the Python tool *musicdiff*¹⁰, obtaining the number of edit operations $\delta_{e,e'}$ (with $e, e' \in \{1, 6\}$ denoting the editions and $e \neq e'$) necessary to convert one edition into the other. *musicdiff* is ideal for our purpose because it focuses on finding *visual* differences between two scores rather than only differences that affect the performance. This includes changed pitches and missing or added accidentals, dynamics, or articulations. Additionally, differences in the appearance of symbols that do not change the semantics, including changed stem direction or different placement of slurs, are captured. They prescribe the same performance but represent the same musical content in different but equivalent

ways. It also does not matter whether properties of the musical symbols or the symbols themselves are encoded as MEI elements or attributes. The *musicdiff* output does not distinguish between MEI elements and attributes.

musicdiff follows the comparison workflow described by Foscarin et al. (November 2019). Crucial for determining the edit operation to be performed to obtain the least costly sequence alignment are the costs for each edit operation. For the *musicdiff* algorithm the costs of deletion and insertion depend on the “size” of the element to be deleted or inserted. The size of a symbol, in turn, depends on the number of properties it possesses. For example, a rest has a size of 1 (its only property is its duration), a note has a size of 2 (pitch name and duration; the octave number is included in the pitch name, as in the case of D4, for example) and a note with a tie has size 3. Whether the tie is encoded as an attribute of the note or as an element does not matter. The number of properties does not depend on the encoding. The cost for deleting or inserting a complete bar is the sum of the sizes of all the symbols in that bar.

musicdiff produces text files containing a chronological list of edit operations. Each edit operation is listed with the location of its appearance in the format @@ measure n , staff $1/2$, beat $x@@$. We counted the number of these lines as the total number of edit operations.

Full-Length Comparisons Figure 1 presents the results of the pairwise comparisons of the six full-length encodings as a Neighbor-Joining tree (Saitou and Nei 1987). To calculate the distances we used the total number of differences between each pair of editions, summed over all Bagatelles. The tree was generated using *SplitsTree* (Huson and Bryant 2024). The distances we used to construct the tree in Figure 1 are shown in Table 1. We only used the results for Bagatelles No. 1–6 because we had to exclude No. 7 from our analysis, see below Drawing samples.

In the tree, each edition is located at the tip of a branch. The branch lengths correspond to the pairwise distances, reflecting the degree of relatedness between editions.

Table 1. Total number of differences for the pairwise comparisons of the Bagatelles' Nos. 1–6 full-length encodings.

	BdA	Zulehner	André	Schott	Hasl.	Breitk.
BdA	0	1195	2035	2022	2745	2698
Zulehner		0	1878	1591	2443	2559
André			0	1690	2276	2500
Schott				0	1849	2437
Hasl.					0	2187
Breitk.						0

Editions that are grouped more closely together exhibit greater similarity.

As it happens, the structure of the tree corresponds to the temporal order of the editions in our corpus. The editions by BdA (1803) and Zulehner (1808) are clustered together at one end. At the opposite end of the tree, the editions by Hasl. (1845) and Breitk. (1864) are clustered. The former are the earliest prints in our corpus; the latter are the most recent editions. The Breitk. edition – the most recent – has the longest distance to the first edition by BdA. In total, there are 2,698 differences between these editions, while there are only 1,195 differences between BdA and Zulehner. Between these two groups the editions by André (1825) and Schott (1826) are positioned in intermediate locations.

Sampling Algorithms

We drew the samples using three algorithms that selected bars from the full encodings. For this case study we set the sample size to 10% of the bars in each Bagatelle. We chose bars as the basic unit because the comparison with `musicdiff` requires valid MEI files. Moreover, musical notation only makes sense in context. A single note without at least a clef and key signature is meaningless. Thus, differences cannot be correctly identified without considering their surrounding context.

All scripts were written in Python. For mathematical operations and data handling the `numpy` and `pandas` libraries were used. `musicdiff` depends on the `music21` and `converter21` libraries. The algorithms take a Python dictionary called “measures” as input. The keys are the bar numbers, given as @n-attributes in the MEI files, and the values are the number of musical symbols inside the bars. These symbols are not to be confused with the MEI elements used to encode the music, since, for example, it is possible to encode accidentals as attributes of a `note`-element as well as a child element of this `note`-element. We decided to encode all musical symbols as elements wherever possible, because this way they can be counted more easily. In the following, we will refer to MEI elements as elements and to printed musical symbols as symbols.

To count the elements and copy the selected bars to new files we used the Python module `xml.etree.ElementTree`. Since MEI is an XML-based encoding, it can be parsed by `ElementTree`. The elements to be considered were given as a list. This list defines the MEI elements to be counted by the parser, for example `note`, `rest`, and `slur`. It covers the MEI representations of all musical symbols relevant in the editions under study.

Algorithm 1: Random selection (randSel) First, we used an algorithm that randomly selects bars from the piece without considering the contents of the bars (see Algorithm 1). This means that the algorithm randomly draws numbers out

of a list of numbers ranging from 0 (incomplete pickup beats were encoded with @n=0) to the total number of bars in the piece.

Algorithm 2: Selection based on bar number and symbol count (barSymCount) The second algorithm bases the selection on the number of musical symbols inside the bars. While we still controlled the number of selected bars, this algorithm requires that the selected bars should contain 10% of the symbols in the entire piece (see Algorithm 2). For more flexibility the number of symbols was not fixed exactly. The samples were allowed to be 5% smaller or larger than the target. If 10% of the symbols equaled 100, we would allow 95–105 symbols.

Algorithm 3: Selection based only on symbols (onlySym) The third algorithm selects bars based only on the number of musical symbols. The percentage of selected bars is not preset. As a result, the number of bars may be (considerably) smaller or larger than the ~10% of musical symbols (see Algorithm 3). As with the previous algorithm, the number of symbols in the sample may vary between 95 and 105% of the required number.

Drawing Samples After performing the full-length comparisons and after determining $\delta_{e,e'}$, we drew the samples. The first print by *Bureau d'Arts et d'Industrie* was always used as the basis for sampling. For a discussion of potential biases of using a particular edition as a basis, see A Closer Look: onlySym Algorithm. We repeated this sampling process 10,000 times per Bagatelle and algorithm, resulting in 30,000 samples per Bagatelle, 210,000 in total. For each sample, we created new files. The first of the seven Bagatelles contains 97 bars; with the first two algorithms each sample contained 10 bars. For each of the editions we thus created 10,000 valid MEI files that only contain the 10 sampled bars each for the *randSel* and the *barSymCount* algorithm and an arbitrary number of bars for the *onlySym* algorithm. These sample files were then compared pairwise using `musicdiff`. With 6 editions under consideration, we get 15 comparisons per sample, resulting in 450,000 comparisons per Bagatelle. For each sample s we obtain the number of differences between each pair of editions, e and e' , $\delta_{e,e'}(s)$. We tested whether the samples represented the whole encoding by comparing the number of differences in the samples to the theoretical expectation value $\mu_{e,e'}$.

In other words, a sample of size $x\%$ of the bars is expected to include $x\%$ of the edit operations compared to the full piece, or $\delta_{e,e'}(s) \approx \frac{x}{100} \delta_{e,e'}$, where $x \in [0, 100]$. From the total number of differences we calculated the theoretical mean, that is, 10% of the total, which serves as the theoretical expectation value $\mu_{e,e'}$, the number of differences the samples should contain in order to represent the full piece.

Unfortunately, we had to exclude Bagatelle No. 7 from the analysis. The comparisons failed because the Breitkopf & Härtel edition has more bars than all the others:

In this print, bar 86 is part of the first and second endings of a repeated section. All the other prints start with the endings in bar 87. For this reason, all subsequent bars have differing @n-attributes in the Breitkopf edition. Since the files with the sampled bars are generated automatically, we would be comparing bars to each other that do not contain the same music. This is not reasonable, much as manually creating 30,000 sample files would not be feasible.

For the 6th Bagatelle almost 4,000 comparisons failed in a first run. Most of these were the comparisons of the editions by Schott with Breitkopf (nearly 3,000 missing comparisons). The failures were caused by a slur in the penultimate bar in the Schott edition, which is a tie in the other editions. The problem only occurs when comparing Schott to Breitkopf in that order. When the order of comparison is changed, *musicdiff* works as expected. The same is true for the other missing comparison with one exception. This exception is the comparison of the edition by André and Breitk. for sample No. 1659, drawn with the *barSymCount* algorithm. We excluded this sample from the analysis and completed the comparisons.

However, this raises questions about *musicdiff*'s performance in general. Whether score A is compared to score B or vice versa should not affect the result. The fact that this is not the case here would need further investigation, which is beyond the scope of this study.

It would also be possible to extract the distribution of differences using the comparisons of the full-length encodings. However, since we are interested in the performance of three potential sampling algorithms, this does not suffice. Even if we knew this distribution, it would not tell us how well the algorithms perform on real data. In particular, we would not know the overall accuracy with which the algorithms draw samples. We can only know this through repeated sampling, which allows us to calculate the mean accuracy as well as the standard deviations.

To ensure that the results are not distorted by one or more identical samples we checked the generated samples. In particular, for the algorithm described in Algorithm 2: *Selection based on bar number and symbol count* (*barSymCount*) it was not obvious whether there were enough possible combinations.¹¹ However, in only two cases, both for Bagatelle No. 5, with the *randSel* and the *onlySym* algorithms, was one sample drawn twice. In all the other cases we obtained 10,000 distinct samples. Therefore, our results are not affected by counting the same comparisons more than once.

Analysis

For each pair of editions we visualized the results such that the x-axis represents the number of edit operations and the y-axis the frequency of a particular number of edit operations (see Figure 3). We obtained unimodal distributions skewed to the right. We used the Anderson–Darling test

to assess whether the data are normally distributed. The critical value for $p < .01$ is 1.092, and the values for the data all exceed 5.36, with a maximum of 645.16 for the comparison between BdA and Breitk. for Bagatelle No. 5 (for full results see Tables A2–A7). Thus, all our data deviate significantly from the normal distribution.

Similar to the analysis of the full-length encodings, we constructed Neighbor-Joining trees for the comparisons results. Figure 4 shows the Neighbor-Joining tree for Bagatelle 1, *onlySym* algorithm. We used the data from Table A8. Again, BdA and Zulehner are grouped together as are Haslinger and Breitkopf. However, unlike in Figure 1, the Schott edition is no longer positioned between these two groups but is now more closely related to Zulehner's edition. Highlighted in red is the path connecting the editions of BdA and Zulehner. Its length corresponds to the empirical mean in Figure 3.

We know the theoretical expected mean number of differences ($\mu_{e,e'} = 0.1 * \delta_{e,e'}$) we aim to obtain using our three algorithms. We now compared the results of the sampling process, namely the empirical means $\bar{\delta}_{e,e'}$, with the theoretical values.

We calculated the difference between the expected number of differences for each Bagatelle and each pair of editions and the corresponding empirical means obtained from the algorithms. Based on this, we computed the average of these differences,

$$\Delta = \left(\frac{6}{2}\right)^{-1} \sum_{\substack{e,e'=1 \\ e \neq e'}}^6 |\mu_{e,e'} - \bar{\delta}_{e,e'}|, \quad (1)$$

as well as the mean of the standard deviation (see Tables 2 and 3).

We used the results to evaluate the performance of the algorithms. The main criterion is Δ , which should be as small as possible. In addition, we took into account the 95% confidence intervals. Since our data are not normally distributed, we did not use the standard deviation as decisive criterion.

Results

The performance of the three algorithms only differs slightly. As Table 2 shows, the first algorithm (*randSel*), choosing bars completely at random, differs the least from the theoretical mean; on average, it deviates by 2.72 differences. The second algorithm (*barSymCount*) deviates by 2.73 differences. The third algorithm differs the most with a deviation of the mean of 3.81 differences. Table 2 also shows the 95% confidence intervals of the Δ values, calculated using Bootstrap resampling with 10,000 iterations. The confidence intervals for algorithms *randSel* and *barSymCount* are again very similar, with slightly smaller intervals for the *barSymCount* algorithm. The intervals for the *onlySym* algorithm do not overlap with the intervals

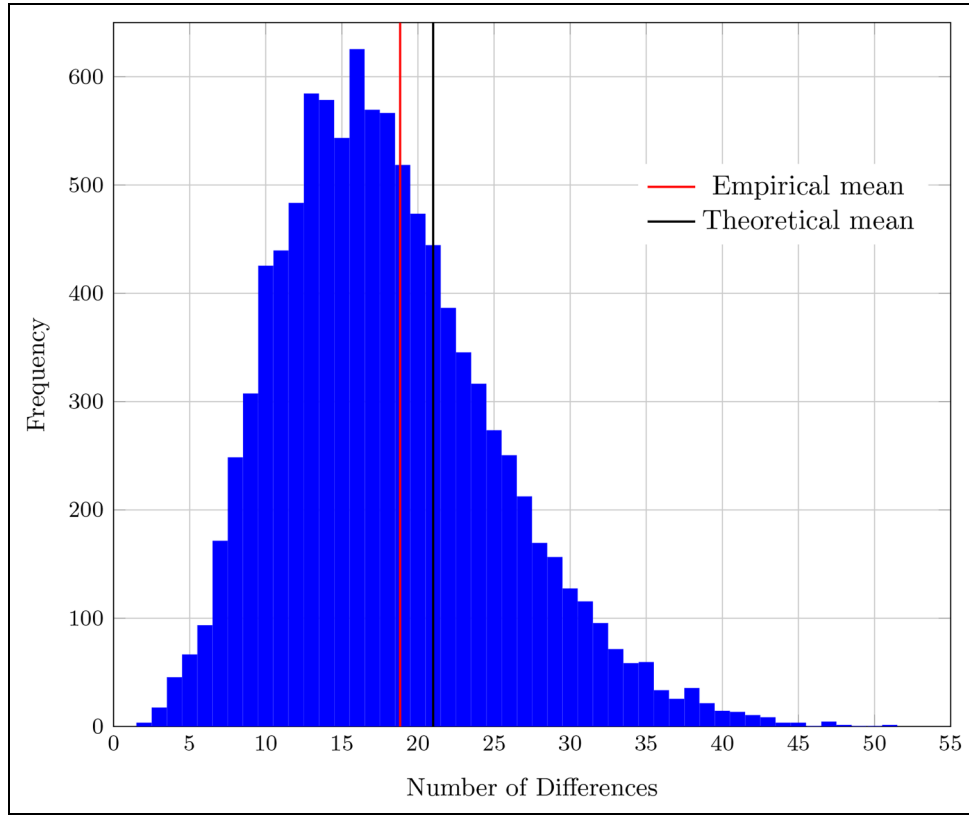


Figure 3. Example plot of the results for the comparison of the editions by BdA and Zulehner, *onlySym* algorithm, Bagatelle I. The red line shows the empirical mean $\bar{\delta}_{e,e'}$ and the black line the theoretical mean $\mu_{e,e'}$.

of the other two algorithms in half of the cases (Bagatelles Nos. 1, 2, and 6). Based on these results, the *barSymCount* algorithm appears to be the most suitable for our endeavor. It deviates about the same as the *randSel* algorithm but has narrower confidence intervals. However, Table 2 shows that the difference between these two algorithms is small: For three Bagatelles, Nos. 2, 3, and 5, the *randSel* algorithm has the smaller deviation, whereas for the remaining three, the *barSymCount* algorithm performs slightly better. Tables A8–A13 show how close the results of these two algorithms are. Thus, the choice can be considered arbitrary. As a decisive reason to choose the *randSel* algorithm over the *barSymCount* algorithm we take into account the complexity of the algorithms and consider the random selection approach to be computationally less costly.

Discussion

A Closer Look: Bagatelles Nos. 3 and 5

It is worth noting that in almost all cases, the empirical and theoretical means are consistent within the standard deviation of the empirical values. There are only three comparison pairs where all the algorithms deviate from the theoretical values: for Bagatelle No. 3, the comparisons of

Zulehner’s edition with Hasl. and Breikt.; for Bagatelle No. 4, again the comparison of Zulehner’s edition with the one by Hasl. In the case of the *onlySym* algorithm, there is a fourth such comparison, Bagatelle No. 5 comparing the editions by Zulehner and André.

In the case of Bagatelle 3 the samples overestimate the number of differences (see Table A10). This appears to be due to the comparison of the full files not producing the right output. Many bars were not compared in detail; instead, they were recognized as missing. This results in a much lower number of differences in the comparison of the complete files and consequently in a much lower $\mu_{e,e'}$. The reason why the comparisons did not work as expected is unclear. The assignment of costs to the edit operations should have led to editing individual symbols in the bars, rather than entire bars.

In general, the results of the algorithms mostly underestimate the “real” number of differences. The exceptions are only with Bagatelles Nos. 3 and 5. These exceptions are as follows:

- Algorithm *randSel*:
 - Bag. 3, 4 comparisons.
 - Bag. 5, 11 comparisons.

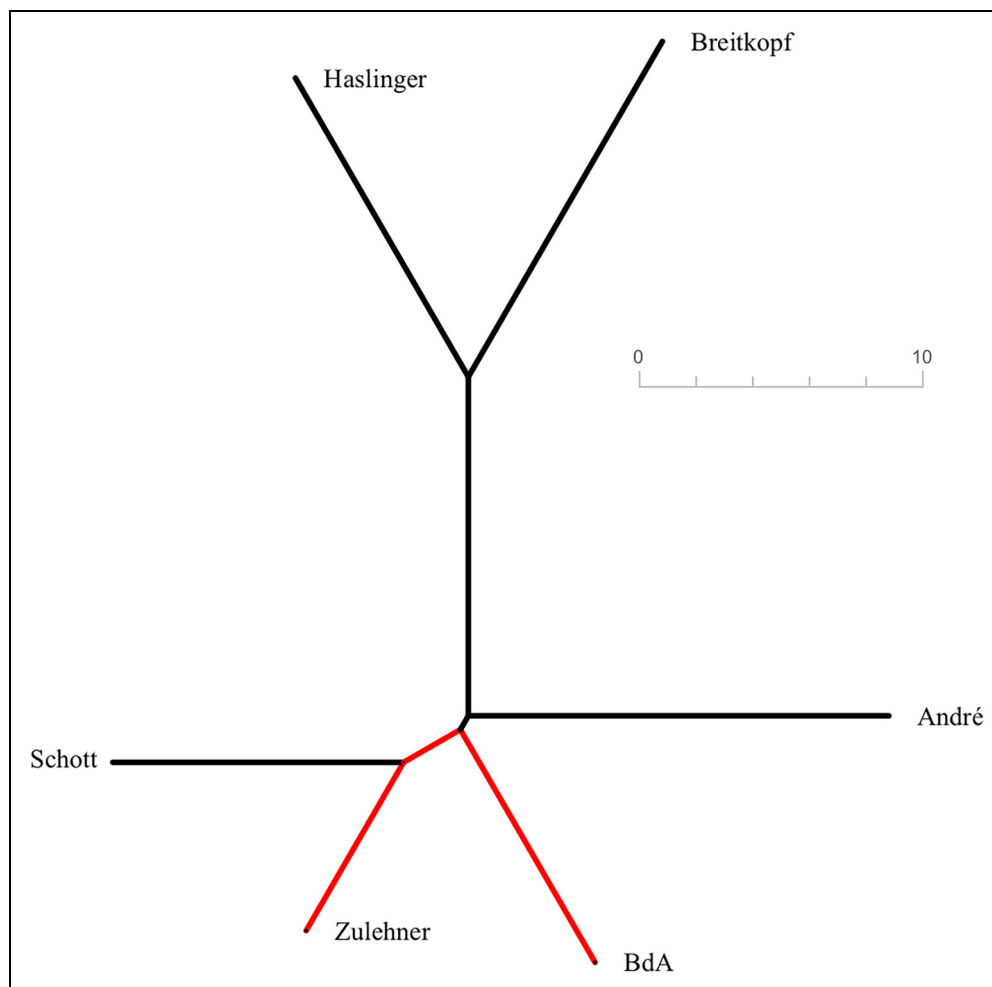


Figure 4. Neighbor-joining tree for the *onlySym* algorithm, Bagatelle I. Highlighted in red is the path connecting the BdA and Zulehner editions. The tree is calculated with the data shown in Figure 3.

- Algorithm *barSymCount*:
 - Bag. 3, 6 comparisons (including all the ones from *randSel* algorithm).
 - Bag. 5, 11 comparisons (the same as for *randSel* algorithm).
- Algorithm *onlySym*:
 - Bag. 3, 2 comparisons.
 - Bag. 5, 3 comparisons.

We have already discussed the results for Bagatelle No. 3. We now turn to Bagatelle No. 5. As shown in Figure 5, there are some comparisons whose results are not only non-normally distributed but bimodal. These are the comparisons involving the edition of Breitkopf & Härtel. The patterns are similar across all algorithms. Still, the empirical means are very close to the theoretical means, indicating that the algorithms still perform well. Since the deviations of the data are relatively large, the standard

deviations are also higher compared to the standard deviations for the other Bagatelles (see Table 3).

The reason for the distribution of the results appears to be primarily two bars, Nos. 63 and 64. The former is shown in Figure 6; the latter looks similar. Only in the edition by Breitkopf & Härtel (see Figure 6b), are the 16th notes beamed together, with the rests in between somehow included in the beams. In all the other editions this is printed as can be seen in the example from the first edition. Since *musicdiff* compares visually, the comparison detects different beaming for every note or chord and every rest, resulting in an extremely high number of differences in these two bars. The second peak visible in some of the results in Figure 5 probably corresponds to all the samples containing these two bars. In a qualitative analysis of the comparison results one would probably count this as only a single variant; however, since *musicdiff* compares all the symbols to each other it treats the missing beam as a difference for all the notes and rests in the bars. This is a methodological decision on how to compare two sequences.

Table 2. Average deviation of the empirical means (Δ) and 95% confidence intervals of the three algorithms per Bagatelle. The values were rounded to two decimal places.

No.	$\Delta_{randSel}$	95% CI	$\Delta_{barSymCount}$	95% CI	$\Delta_{onlySym}$	95% CI
1	1.95	[1.49, 2.38]	1.87	[1.42, 2.28]	3.32	[2.76, 3.84]
2	2.67	[1.72, 3.73]	2.75	[1.80, 3.73]	5.19	[4.14, 6.38]
3	3.22	[0.77, 6.66]	3.25	[0.77, 6.68]	4.24	[2.39, 6.83]
4	3.92	[3.12, 4.84]	3.71	[2.94, 4.64]	4.15	[3.39, 5.06]
5	2.54	[1.65, 3.58]	2.80	[1.92, 3.83]	2.96	[1.73, 4.46]
6	2.02	[1.53, 2.50]	2.01	[1.54, 2.49]	3.01	[2.50, 3.51]
Ø	2.72	—	2.73	—	3.81	—

In summary, despite the noted conspicuous cases, all algorithms perform well. The large differences between the theoretical and the empirical data can be readily explained and assessed in terms of their impact on this study.

A Closer Look: *onlySym* Algorithm

Since the third algorithm *onlySym* selects bars based on the number of musical symbols in the piece, another question arises: Which edition should we use as our basis for sampling? Usually, all editions contain the same number of bars (except for cases like the one discussed for Bagatelle No. 7, where the Breitkopf edition contained one extra bar), but the number of symbols in the editions naturally differs. However, analysis shows that the differences in the number of musical symbols in the six editions are marginal, as shown in Table 4.

The numbers highlighted in red indicate cases where prints contain a strikingly large number of symbols compared to the others, resulting in a comparatively large relative difference of more than 14% and 12%, respectively, opposed to values smaller than 8%. The 2nd Bagatelle is another example where the number of bars differ in the editions: In the two editions by Haslinger and Breitkopf & Härtel a section is printed that repeats an earlier section. In all the other editions this is indicated by *Scherzo Da Capo Senza ripetizione e dopo el trio*. Excluding the additional bars, the two editions contain 2197 (Haslinger) and 2182 (Breitkopf) musical symbols, yielding a maximal

relative difference of about 5.5%, which is much more consistent with the relative differences for the other Bagatelles.

The large difference in the number of musical symbols in the Haslinger edition of the third Bagatelle is a direct consequence of the difference in notation between the editions. In the Haslinger print many more symbols indicate articulations like *legato* or *staccato* (see Table 5). Nevertheless, because the number of musical symbols in the editions only vary by percentages below 8%, this is a first indication of the nature of differences between the editions: It appears that the number of differences occurring because of added or deleted symbols is likely much smaller than the number of differences occurring because of changed appearance of symbols. This concerns, for example, altered stem directions, modified length of slurs, or changed placement of dynamics and articulations.

Based on these results it should not matter which encoding is used as a basis for the *onlySym* algorithm; for consistency, we again chose the first print.

Distribution of Differences in Musical Prints

Each algorithm represents different assumptions about the distribution of differences between editions of musical pieces. The underlying assumption for the first algorithm (*randSel*) is that the differences are distributed completely evenly. This means that the probability of finding a given number of differences is the same for every location in the score. The second (*barSymCount*) and third (*onlySym*) algorithms assume that the number of differences correlates with the number of musical symbols; in other words, bars containing more musical symbols are more likely to contain differences than bars with fewer symbols.

These assumptions also reflect different views on the process of creating an edition of a musical work. If the number of differences is correlated with the density of musical symbols, this could indicate that most differences result from deliberate editorial choices, leading to what we have called “propagating differences.” Additionally, this can hint at the engraver’s susceptibility to errors: If the number of differences and the symbol density correlate, one might assume that more complex parts of the score are more prone to errors by the engraver.

Table 3. Standard deviation of the result data. The values were rounded to two decimal places.

No.	$\sigma_{randSel}$	$\sigma_{barSymCount}$	$\sigma_{onlySym}$
1	9.62	9.46	9.24
2	8.97	8.91	9.68
3	9.47	9.17	9.00
4	7.57	7.52	7.67
5	14.63	14.96	14.97
6	5.80	5.76	6.74
Ø?	9.78	9.30	9.55

Table 4. Number of musical symbols in the six editions of the Bagatelles Op. 33. The numbers highlighted in red show cases where prints contain a strikingly large number of differences in comparison to the other prints.

No.	BdA	Zulehner	André	Schott	Hasl.	Breitk.	Mean	$Diff_{max}$	$Diff_{rel}$
1	2152	2151	2124	2151	2186	2211	2162.5	87	0.0402
2	2163	2159	2127	2152	2453	2446	2250	326	0.1449
3	1311	1312	1288	1313	1451	1356	1338.5	163	0.1218
4	1483	1481	1466	1489	1491	1579	1498.17	113	0.0754
5	3162	3156	3102	3121	3107	3162	3135	60	0.0191
6	1380	1377	1364	1377	1399	1405	1383.67	41	0.0296
7	2299	2295	2300	2296	2369	2430	2331.5	135	0.0579

If, conversely, differences are distributed randomly, one could assume that editorial choices mostly produce local changes rather than propagating differences and that the probability of errors by the engraver is independent of the complexity of the score.

The *barSymCount* algorithm represents a compromise between both possibilities. Since it is limited in the selection process, it incorporates both potential sources for differences. First, because of the condition that a certain number of symbols must be matched, it reflects the assumed correlation between differences and symbol density. Second, because of the condition that a certain number of bars needs to be chosen, it reflects the assumption of a uniform distribution of differences. Since we consider the first hypothesis to be more plausible, we expected the *onlySym* algorithm to work best.

This hypothesis is not supported by our findings. On the contrary: The algorithm that selects bars solely based on the number of symbols performs worst, while the completely random selection yields the best results. The *barSymCount* algorithm, which combines the methods of the other two, performs equally well.

We believe this to be due to the differing impact of the two kinds of differences presented: Most of the differences are local differences, while only a few are propagating. The

relatively poor performance of the third algorithm seems to confirm this: It is designed to find propagating differences that affect groups of symbols—differences that do not seem to have a strong effect on the overall number of differences. Furthermore, since there seems to be no or little correlation between differences and symbol density, one can assume that the number of errors made by the engravers is low and largely independent of symbol density.

The example of the 5th Bagatelle particularly illustrates these observations. For all the comparison pairs and all Bagatelles, the results generally resemble a right-skewed distribution, and the Anderson–Darling test confirms that the data are in fact not normally distributed. The 5th Bagatelle shows that this skewness is produced by the mentioned superimposition. Most of the differences are randomly distributed, constituting the bulk of the distributions, while a few differences are induced by propagating, deliberate choices. These are indeed correlated with the symbol density and thus only have an effect if bars containing the affected symbols are selected for the sample. Only in the 5th Bagatelle is this effect strong enough to be clearly visible in the results because the two bars heavily increase the number of visual differences (see section Results and Figure 5). However, this example shows how small the effect of propagating differences is. In bars 63 and 64 there are many propagating differences; in this case, one might expect that the *barSymCount* algorithm performs better than the *randSel* algorithm. As Table 9 shows, this is not the case: The mean deviation is actually larger for the *barSymCount* algorithm.

In other words: Most differences between two editions of the same work affect only individual symbols, and only rarely is a whole group of symbols affected. This is not to say that the impact of these differences on performances is small, but visually, they appear minor.

However, there are some limitations to our results. First, some of the differences arise from the encoding process. As in all processes of music copying, errors can occur in the encoding.

Second, we only used a very limited set of musical works for our research. A natural next step would therefore be to extend the corpus of works under investigation. Third, we only considered six different editions of the works, which

Table 5. Number of selected MEI elements in the Bagatelle Op. 33 No. 3. Highlighted in red are strikingly large numbers for Artic- and Slur-elements in the editions of Haslinger and Breitkopf & Härtel.

Element	BdA	Zulehner	André	Schott	Hasl.	Breitk.
beam	221	223	222	223	222	224
note	850	851	849	851	850	850
rest	23	23	24	23	23	23
artic	30	32	26	32	131	32
tempo	-	-	1	1	1	1
dynam	50	49	41	50	49	49
dir	-	1	1	1	1	-
slur	51	50	46	47	98	89
chord	17	17	16	17	12	19
accid	54	51	51	53	52	54
tie	15	15	11	15	12	15

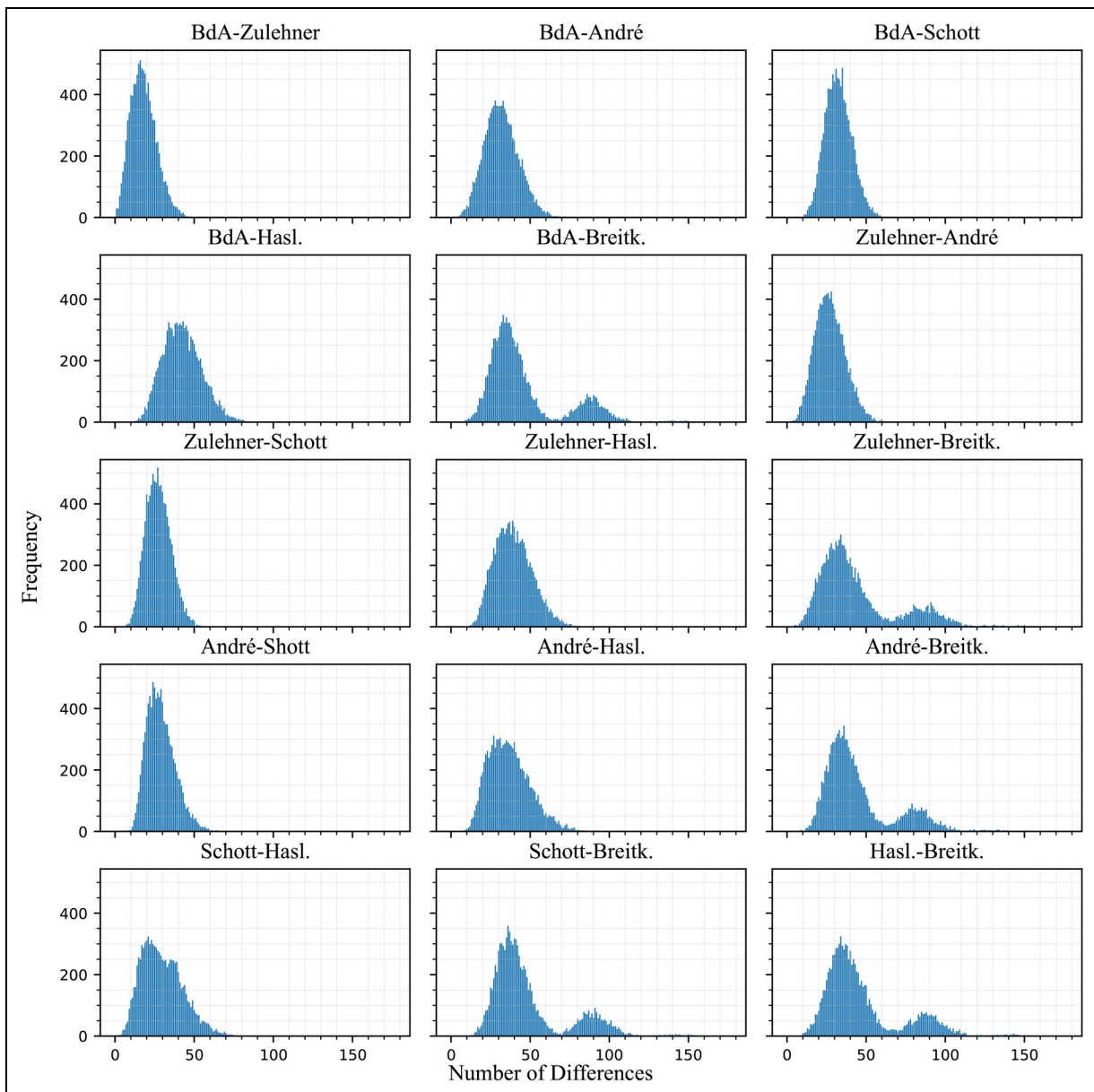


Figure 5. Results of the comparisons of Bagatelle No. 5 for the samples drawn with the *onlySym* algorithm. All plots share the same x- and y-axis.

should also be expanded in future work. Another crucial limitation is that we used the default settings of the *musicdiff* tool. In particular, we did not change the costs for edit operations. The costs of each edit operation heavily influence the results of alignment algorithms. Changing costs based on musicological considerations can change the outcome of the investigation (see Merkl 2022). However, even putting aside these methodological issues, Bagatelle No. 6 showed that *musicdiff* may still require further development to ensure more reliable performance. Especially, it should not matter if score A is compared to score B or vice versa. We also did not assign different weights to specific types of differences but counted all

differences as one difference. In the tradition of stemmatology, one could consider defining some kinds of differences as “more important” than others. To do so, it would be necessary to analyze the results of the comparisons not only quantitatively but also qualitatively. Still, since we were first and foremost interested in quantitative analysis, our findings allowed us to conduct the desired research. Also, the distribution of differences is not affected by musicological evaluation of their meaning; that is, the position of differences remains unchanged when they are qualitatively analyzed.

Despite these limitations, we consider our study to be a first step toward understanding editorial practice in

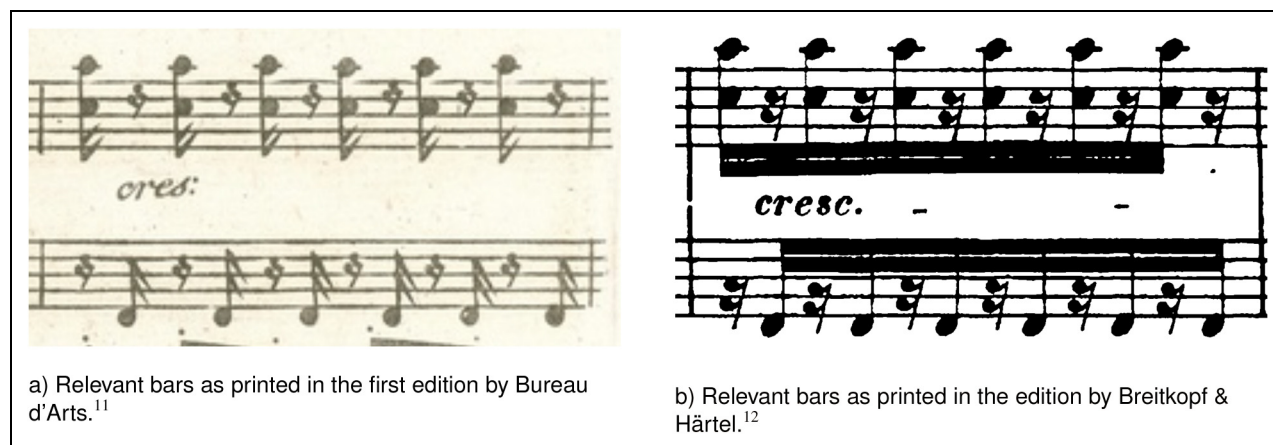


Figure 6. Bar 63, one of the bars responsible for the results in Figure 5.

19th-century German-speaking countries. The results lead us to reject our initial hypothesis, which assumed a correlation between symbol and edit operation distribution, based on the idea of deliberate editorial decisions. This notion does not appear to hold. However, as discussed, further investigation on a larger corpus is needed.

This further investigation may now be conducted more easily, because we have answered our initial research question and shown that analyzing a randomized subset of data can yield results for musicological research. Even though the development of sampling methods for symbolic music corpora is still in its early stages, we believe that this approach is promising for other corpus studies as well, for example those applying phylogenetic methods to musicology, such as Ballen et al. (2024), Windram et al. (2014), Windram et al. (2022), Plaksin (2021), or projects like “*Das Handwerk des Verlegers*” (“The Editor’s craft”) conducted at the Beethoven-Haus Bonn (2021–2024). This finding shows that the use of sampling methods may in the future provide a way to reduce the workload of preparing music encodings significantly. This will enable research that is not dependent on preexisting corpora, so musicological questions do not have to fit these corpora but rather the corpora can fit the questions.

Acknowledgements

We used OpenAI’s GPT-5-Mini model to help improve the writing of this article. This work was realized as a part of KreativInstitut.OWL. It is a project in research cooperation of the OWL University of Applied Sciences and Arts, Detmold University of Music and Paderborn University, funded by the Ministry of Economic Affairs, Industry, Climate Action and Energy of the State of North Rhine-Westphalia.

ORCID iDs

Adrian Nachtwey  <https://orcid.org/0009-0000-1214-5547>

Fabian C. Moss  <https://orcid.org/0000-0001-9377-2066>

Anna Viktoria Katrin Plaksin  <https://orcid.org/0000-0002-9969-0608>

Ethical Approval

This research did not require ethics committee or IRB approval. This research did not involve the use of personal data, fieldwork, or experiments involving human or animal participants, or work with children, vulnerable individuals, or clinical populations.

Funding

We acknowledge support for the publication costs by the Open Access Publication Fund of Paderborn University.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Notes

1. The Breitkopf edition is available at the Petrucci Library, the Schott edition can be found at the Beethoven-Haus Bonn: <https://www.beethoven.de/de/media/view/5114772269826048/scan/0>
2. <https://www.beethoven.de/de/media/view/4956805541134336/scan/0>
3. <https://www.beethoven.de/de/media/view/4697437075668992/scan/0>
4. <https://www.beethoven.de/de/media/view/4908887509565440/scan/0>
5. <https://www.beethoven.de/de/media/view/5114772269826048/scan/0>
6. <https://www.beethoven.de/de/media/view/5673265322262528/scan/0>
7. <https://imslp.org/wiki/Special:ReverseLookup/58117>
8. https://github.com/trompamusic-encodings/Beethoven_Op33_BreitkopfHaertel.git
9. see <https://doi.org/10.5281/zenodo.21510943>.

10. The documentation can be found at <https://github.com/gregchapman-dev/musicdiff>, see Foscarin et al. (November 2019).
11. To be found at the Petrucci Library: <https://imslp.org/wiki/Special:ReverseLookup/58117>

References

- Ballen, G. A., Mühlová, K. H., & Hajič, J. (2024). What did the dove sing to pope gregory? ancestral melody reconstruction in gregorian chant using bayesian phylogenetics. *bioRxiv* <https://doi.org/10.1101/2024.08.19.608653>. URL <https://www.biorxiv.org/content/early/2024/08/19/2024.08.19.608653>.
- Burgoyne, J. A., Wild, J., & Fujinaga, I. (2011). An expert ground truth set for audio chord recognition and music analysis. In *Proceedings of the 12th international society for music information retrieval conference* (pp. 633–638). <https://doi.org/10.5281/zenodo.1417547>.
- Calvo-Zaragoza, J., Martínez-Sevilla, J. C., Penarrubia, C., & Ríos-Vila, A. (2023). Optical music recognition: Recent advances, current challenges, and future directions. In M. Coustaty & A. Fornés (Eds.), *Document analysis and recognition – ICDAR 2023 workshops* (pp. 94–104). Springer Nature. ISBN 978-3-031-41498-5.
- Devaney, J., Arthur, C., Condit-Schultz, N., & Nisula, K. (2015). Theme and variation encodings with roman numerals (tavner): A new data set for symbolic music analysis. *Proceedings of the 16th International Society for Music Information Retrieval Conference. ISMIR*, pp. 728–734. <https://doi.org/10.5281/zenodo.1417497>.
- Foscarin, F., Fournier-S'Niehotta, R., & Jacquemard, F. (November 2019). A diff procedure for music score files. Computation and visualization of the differences between two music score files. 6th International Conference on Digital Libraries for Musicology (DLfM). Den Haag, pp. 7–13. URL <https://inria.hal.science/hal-02267454v2>.
- Goebel, W. (2021). MEI encoding: Sieben Bagatellen, Opus 33 by Ludwig van Beethoven URL <https://zenodo.org/records/4730608>.
- Hentschel, J., Rammos, Y., Neuwirth, M., & Rohrmeier, M. (2025). A corpus and a modular infrastructure for the empirical study of (an)notated music. *Scientific Data*, 12, 1–10. <https://doi.org/10.1038/s41597-025-04976-z>
- Huson, D. H., & Bryant, D. (2024). The splitsree app: Interactive analysis and visualization using phylogenetic trees and networks. *Nature Methods*, 21, 1773–1774. <https://doi.org/10.1038/s41592-024-02406-3>
- Lewis, D., Shibata, E., Hankinson, A., Kepper, J., Page, K., Rosendahl, L., Saccomano, M., & Siegert, C. (2023). Supporting musicological investigations with information retrieval tools: an iterative approach to data collection. *Proceedings of the 24th International Society for Music Information Retrieval Conference. Milan, Italy*, pp. 795–801. <https://doi.org/10.5281/zenodo.10265407>.
- Mayer, J., Straka, M., Hajič, J., & Pecina, P. (2024). Practical end-to-end optical music recognition for piano form music. In E. H. Barney Smith, M. Liwicki, & L. Peng (Eds.), *Document analysis and recognition -ICDAR 2024* (pp. 55–73). Springer Nature. ISBN 978-3-031-70552-6.
- Merkel, R. (2022). *Bioinformatik. Grundlagen, Algorithmen, Anwendungen* (4 edition). Wiley-VCH.
- Neuwirth, M., Harasim, D., Moss, F. C., & Rohrmeier, M. (2018). The annotated beethoven corpus (abc): A dataset of harmonic analyses of all beethoven string quartets. *Frontiers in Digital Humanities*, 5, 1–5. <https://doi.org/10.3389/fdigh.2018.00016>
- Plaksin, A. V. K. (2021). Modelle zur computergestützten Analyse von Überlieferungen der Mensuralmusik. Empirische Textforschung im Kontext phylogenetischer Verfahren, hrsg. von Ralf Martin Jäger (= Wissenschaftliche Schriften der WWU Münster, Reihe XXVI. Schriften zur Musikwissenschaft aus Münster, volume 27).
- Ríos-Vila, A., Calvo-Zaragoza, J., & Paquet, T. (2024). Sheet music transformer: End-to-end optical music recognition beyond monophonic transcription. In E. H. Barney Smith, M. Liwicki, & L. Peng (Eds.), *Document analysis and recognition -ICDAR 2024* (pp. 20–37). Springer Nature. ISBN 978-3-031-70552-6.
- Ríos-Vila, A., Rizo, D., Iñesta, J. M., & Calvo-Zaragoza, J. (2023). End-to-end optical music recognition for piano form sheet music. *International Journal on Document Analysis and Recognition (IJDAR)*, 347–362. <https://doi.org/10.1007/s10032-023-00432-z>
- Saccomano, M., Rosendahl, L., Lewis, D., Hankinson, A., Kepper, J., Page, K., & Shibata, E. (2024). Selective encoding: Reducing the burden of transcription for digital musicologists. *Journal of the Text Encoding Initiative [online]*. <https://doi.org/10.4000/13x2d>
- Saitou, N., & Nei, M. (1987). The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4), 406–425. <https://doi.org/10.1093/oxfordjournals.molbev.a040454>
- Windram, H., Charlston, T., Tomita, Y., & Howe, C. (2022). A phylogenetic analysis of two preludes from J.S. Bach's well-tempered clavier II. *Early Music*, 50(3), 373–393. <https://doi.org/10.1093/em/caac027>.
- Windram, H. F., Charlston, T., & Howe, C. J. (2014). A phylogenetic analysis of Orlando Gibbons's prelude in G. *Early Music*, 42(4), 515–528. <https://doi.org/10.1093/em/cau100>.

Appendix

Table A1. Shorthands for the comparison pairs. In the following the comparisons are labeled by numbers. Each number represents a pair of editions.

#Comp	Edition Pair
1	BdA–Zulehner
2	BdA–André
3	BdA–Schott
4	BdA–Haslinger
5	BdA–Breitkopf
6	Zulehner–André
7	Zulehner–Schott
8	Zulehner–Haslinger
9	Zulehner–Breitkopf
10	André–Schott
11	André–Haslinger
12	André–Breitkopf
13	Schott–Haslinger
14	Schott–Breitkopf
15	Haslinger–Breitkopf

Table A2. Results of the Anderson–Darling test for normal distribution, Bagatelle 1. Critical value 1.092, $p < .01$; values rounded to two decimal places.

#Comp	<i>randSel</i>	<i>barSymCount</i>	<i>onlySym</i>
1	78.70	63.81	78.70
2	29.34	25.26	29.34
3	44.53	35.95	44.53
4	25.96	20.97	25.96
5	13.56	8.83	13.56
6	21.00	17.92	21.00
7	25.75	27.34	25.75
8	19.25	20.24	19.25
9	10.77	8.62	10.77
10	16.42	13.56	16.42
11	38.96	32.68	38.96
12	32.40	27.46	32.40
13	19.87	20.79	19.87
14	12.95	9.67	12.95
15	15.02	15.04	15.02

Table A3. Results of the Anderson–Darling test for normal distribution, Bagatelle 2. Critical value 1.092, $p < .01$; values rounded to two decimal places.

#Comp	<i>randSel</i>	<i>barSymCount</i>	<i>onlySym</i>
1	20.51	22.65	31.34
2	11.39	12.24	12.36
3	7.02	7.43	11.35
4	12.76	18.14	23.84
5	12.64	17.68	22.30
6	10.66	8.78	9.99
7	9.29	8.94	11.97
8	20.66	23.48	29.51
9	24.27	30.51	41.18
10	10.50	8.58	11.43
11	15.64	22.62	24.32
12	16.97	22.20	27.34
13	13.91	16.58	19.88
14	13.48	17.24	20.83
15	19.88	18.5	19.45

Table A4. Results of the Anderson–Darling test for normal distribution, Bagatelle 3. Critical value 1.092, $p < .01$; values rounded to two decimal places.

#Comp	<i>randSel</i>	<i>barSymCount</i>	<i>onlySym</i>
1	70.12	105.95	114.05
2	8.08	14.09	17.87
3	7.02	10.2	9.96
4	21.91	36.40	42.18
5	5.36	10.04	9.61
6	7.11	9.93	11.82
7	7.83	9.94	9.95
8	26.47	37.75	42.01
9	5.96	10.99	10.77
10	25.73	23.70	26.32
11	28.60	33.59	37.16
12	20.14	19.74	18.57
13	41.71	45.11	54.07
14	16.79	17.13	18.99
15	40.21	35.15	43.69

Table A5. Results of the Anderson–Darling test for normal distribution, Bagatelle 4. Critical value 1.092, $p < .01$; values rounded to two decimal places.

#Comp	<i>randSel</i>	<i>barSymCount</i>	<i>onlySym</i>
1	15.12	18.57	20.05
2	22.81	26.66	24.79
3	17.87	16.09	21.16
4	18.68	21.23	17.36
5	17.74	18.32	18.89
6	26.68	24.66	26.72
7	59.59	47.50	46.42
8	14.86	18.08	15.88
9	8.87	11.53	14.41
10	29.60	25.85	26.63
11	18.15	21.49	14.54
12	11.46	11.43	12.44
13	23.49	33.05	20.92
14	25.94	28.96	23.19
15	11.75	12.80	14.81

Table A7. Results of the Anderson–Darling test for normal distribution, Bagatelle 6. Critical value 1.092, $p < .01$; values rounded to two decimal places.

#Comp	<i>randSel</i>	<i>barSymCount</i>	<i>onlySym</i>
1	36.05	38.23	37.61
2	21.19	24.00	20.54
3	27.37	27.84	29.02
4	16.73	21.59	16.88
5	13.18	17.54	11.90
6	30.45	34.10	25.77
7	41.97	35.08	41.07
8	17.88	17.45	16.51
9	16.97	19.97	12.89
10	31.59	34.25	30.71
11	14.33	14.23	13.05
12	14.56	16.69	11.20
13	18.40	16.50	23.98
14	15.10	17.99	17.01
15	29.26	37.23	40.45

Table A6. Results of the Anderson–Darling test for normal distribution, Bagatelle 5. Critical value 1.092, $p < .01$; values rounded to two decimal places.

#Comp	<i>randSel</i>	<i>barSymCount</i>	<i>onlySym</i>
1	50.13	35.12	40.32
2	9.02	7.28	16.24
3	24.69	15.79	16.17
4	23.65	19.92	21.49
5	544.51	579.70	645.16
6	12.08	10.86	24.20
7	35.05	29.02	22.28
8	36.07	32.44	28.99
9	514.24	510.47	477.88
10	47.60	45.58	63.55
11	66.63	66.49	57.61
12	461.69	479.04	501.89
13	103.70	97.80	66.68
14	506.27	531.52	619.60
15	417.83	443.74	519.50

Table A8. Overview of the results for Bagatelle No. 1.

#Comp	$\mu_{e,e'}$	$\bar{\delta}_{randSel}$	$\bar{\delta}_{barSymCount}$	$\bar{\delta}_{onlySym}$
1	21.00	18.83	18.93	18.00
2	27.70	25.95	25.94	24.81
3	25.90	23.87	23.98	22.94
4	38.40	38.17	38.17	36.61
5	37.10	35.59	35.69	34.07
6	26.70	25.14	25.12	24.15
7	18.20	17.99	18.11	17.20
8	39.80	37.52	37.48	35.82
9	41.60	38.19	38.43	36.44
10	31.60	30.29	30.34	29.02
11	42.00	39.52	39.64	37.74
12	46.50	44.04	44.14	42.13
13	38.30	36.39	36.35	34.75
14	43.00	40.06	40.25	38.29
15	29.90	26.94	27.02	26.00

Table A9. Overview of the results for Bagatelle No. 2.

#Comp	$\mu_{e,e'}$	$\bar{\delta}_{randSel}$	$\bar{\delta}_{barSymCount}$	$\bar{\delta}_{onlySym}$
1	22.70	19.83	19.57	18.44
2	42.70	41.82	41.80	39.36
3	44.50	41.84	41.65	39.19
4	63.60	57.17	57.21	53.79
5	53.80	51.58	51.37	48.52
6	38.20	36.53	36.50	34.29
7	36.40	35.18	35.11	32.97
8	58.80	52.63	52.66	49.56
9	51.00	49.67	49.53	46.83
10	38.80	38.77	38.61	36.34
11	50.30	45.09	45.18	42.47
12	41.80	41.32	41.30	39.05
13	43.10	38.87	38.79	36.54
14	48.50	47.48	47.31	44.67
15	42.40	38.80	38.78	36.67

Table A12. Overview of the results for Bagatelle No. 5.

#Comp	$\mu_{e,e'}$	$\bar{\delta}_{randSel}$	$\bar{\delta}_{barSymCount}$	$\bar{\delta}_{onlySym}$
1	17.80	19.98	19.69	18.01
2	36.40	35.49	35.49	31.87
3	36.50	36.01	35.83	32.42
4	42.40	46.77	47.11	42.43
5	49.30	51.19	51.62	46.25
6	38.20	30.27	30.44	27.31
7	29.20	30.44	30.23	27.37
8	40.00	43.39	43.76	39.37
9	47.50	49.65	50.44	44.70
10	34.10	32.11	32.10	28.90
11	39.60	40.51	40.92	36.74
12	49.60	50.05	50.72	45.23
13	29.50	33.10	33.39	29.83
14	52.20	54.37	54.77	49.08
15	47.80	52.24	52.88	47.40

Table A10. Overview of the results for Bagatelle No. 3.

#Comp	$\mu_{e,e'}$	$\bar{\delta}_{randSel}$	$\bar{\delta}_{barSymCount}$	$\bar{\delta}_{onlySym}$
1	15.30	13.77	13.62	13.15
2	45.30	44.53	44.58	42.35
3	44.80	43.58	43.52	41.44
4	60.40	61.01	61.22	57.99
5	61.80	61.72	61.83	58.81
6	43.30	43.01	43.10	40.98
7	40.40	39.59	39.56	37.66
8	38.80	56.38	56.68	53.61
9	39.00	57.71	57.78	54.95
10	18.60	15.60	15.62	14.92
11	33.40	32.28	32.59	30.74
12	35.10	33.48	33.53	31.98
13	25.50	25.46	25.60	24.14
14	29.20	28.84	28.92	27.49
15	39.20	39.80	40.00	37.83

Table A13. Overview of the results for Bagatelle No. 6.

#Comp	$\mu_{e,e'}$	$\bar{\delta}_{randSel}$	$\bar{\delta}_{barSymCount}$	$\bar{\delta}_{onlySym}$
1	15.80	14.10	14.11	13.43
2	20.50	17.88	17.89	17.01
3	17.90	16.26	16.23	15.49
4	28.90	27.05	27.10	25.90
5	26.60	25.57	25.48	24.45
6	18.30	16.63	16.71	15.87
7	13.60	13.36	13.41	12.74
8	28.90	25.92	26.07	24.83
9	30.90	27.97	27.96	26.68
10	20.10	17.45	17.40	16.65
11	30.30	26.86	26.93	25.78
12	32.20	29.81	29.81	28.49
13	24.70	20.95	20.92	20.04
14	28.10	27.40	27.26	26.08
15 2	21.90	21.26	21.25	20.15

Table A11. Overview of the results for Bagatelle No. 4.

#Comp	$\mu_{e,e'}$	$\bar{\delta}_{randSel}$	$\bar{\delta}_{barSymCount}$	$\bar{\delta}_{onlySym}$
1	26.90	25.26	25.40	25.10
2	30.90	26.26	26.39	26.06
3	32.60	28.93	29.09	28.67
4	40.80	36.83	37.00	36.54
5	41.20	37.65	37.96	37.24
6	23.10	20.39	20.54	20.26
7	21.30	18.74	18.86	18.55
8	38.00	29.27	29.49	29.07
9	45.90	42.13	42.44	41.76
10	25.80	21.06	21.29	20.98
11	32.00	27.37	27.62	27.22
12	44.80	43.17	43.45	42.77
13	23.80	19.86	20.09	19.83
14	42.70	37.27	37.54	36.96
15	37.50	34.27	34.43	34.04

Algorithms

Algorithm 1 Random selection

```

1: procedure RANDOM SELECTION
    random selection(measures, sample size = None)
    Check if variable sample size was given by user:
2: if sample size is not None then
3:     if sample size > number of measures then
4:         Error
5:     else
6:         required measures = sample size
7:     end if
8: else
9:     required measures = ROUND UP(10% of the number of measures)
10: end if
    Declare variable valid measures:
11:    valid measures = List of keys in the 'measures'-Dictionary
    Select measures:
12:    selected measures = random selection of #required measures from List valid measures without replacement
    return selected measures
    END
13: end procedure

```

Algorithm 2 Selection based on number of measures and elements.

```

1: procedure SELECTION OF BARS AND ELEMENTS
    Define function measures elements taking measures and sample size as arguments. Default value of sample size set to None
2:    measures elements(measures, sample size = None)
    Declare lists measure selection, selected elements, and available measures
3:    measure selection = empty list
4:    selected elements = empty list
5:    available measures = List of keys in the 'measures'-Dictionary
    Check if variable sample size was given by user:
6:    if sample size is not None: then
7:        if sample size > number of measures then
8:            Error
9:        else
10:            required measures = sample size
11:        end if
12:    else
13:        required measures = ROUND UP(10% of the number of measures)
14:    end if
    Define percentage share of required measures of all measures
15:    measure ratio = DIVIDE required measures by number of all measures
    Define number of required elements based on measure ratio
16:    required elements = Number of all elements * measure ratio
    Declare variable max attempts, so that the algorithm stops at some point if no solution can be found
17:    max attempts = Number of measures
18:    attempts = 0
    Set boundaries for accepted results:
19:    lower bound = 95% of required elements
20:    upper bound = 105% of required elements
    Declare variable current sum:
21:    current sum = sum of elements in the selected measures
    Select measures as long as number of required measures isn't reached or number of elements is smaller than lower bound:
22:    while length of measure selection  $\neq$  required measures OR current sum < lower bound do
23:        selected measures = add random selection out of List available measures
24:        add number of elements in selected measure to selected elements

```

```

25:         update current sum
      Remove measures from measure selection if number of required measures is exceeded
26:       if length of measure selection > required measures then
27:         randomly remove one measure from measure selection
28:         remove the number of elements in this bar from selected elements
29:         update current sum
30:       end if
31:       raise attempts by 1
      If upper bound is exceeded, remove measures until current sum < upper bound
32:     while current sum > upper bound do
33:       randomly remove one measure from measure selection
34:       remove the number of elements in this bar from selected elements
35:       update current sum
36:     end while
37:     if attempts == max attempts then
38:       break
39:     end if
40:   end while
41:   if current sum < lower bound then
42:     print warning
43:   else if current sum > upper bound then
44:     print warning
45:   else
46:     print celebration
47:   end if
  return selected measures
48: end procedure

```

Algorithm 3 Selection based only on element distribution.

```

1: procedure SELECTION OF BARS ONLY BASED ON NUMBER OF ELEMENTS
   only elements(measures)
   Declare lists measure selection, selected elements, and available measures
2:   measure selection = empty list
3:   selected elements = empty list
4:   available measures = List of keys in the 'measures'-Dictionary
   Define number of required elements
5:   required elements = ROUND UP(10% of all elements in the piece)
   Declare variable max attempts, so that the algorithm stops at some point if no solution can be found:
6:   max attempts = Number of measures
7:   attempts = 0
   Set boundaries for accepted results:
8:   lower bound = 95% of required elements
9:   upper bound = 105% of required elements
   Declare variable current sum:
10:  current sum = sum of elements in the selected measures
   Choose measures until current sum reaches lower bound or attempts > max attempts
11:  while current sum < lower bound AND attempts < max attempts do
12:    x = randomly chosen measure from available measures
13:    ADD x to measure selection
14:    ADD number of elements in x to selected elements
15:    REMOVE x from available measures
16:    UPDATE current sum
17:    attempts += 1
18:  end while
   Randomly remove measures from the selection if current sum is larger than upper bound
19:  while current sum > upper bound do
20:    r = randomly chosen measure from measure selection
21:    REMOVE r from selected measures
22:    REMOVE number of elements in r from selected elements

```

```
23:         UPDATE current sum
24:     end while
25:     if attempts == max attempts then
26:         RETURN SORTED(measure selection)
27:     end if
28:     if current sum < lower bound OR current sum > upper bound then
29:         PRINT warning
30:     else
31:         return sorted measure selection
32:     end if
32: end procedure
```
