AIMC 2023

Statistical evaluation of abc-formatted music at the levels of items and corpora

Laura Cros Vila[1] **Bob L. T. Sturm**[1]

[1]**KTH Royal Institute of Technology**

Published on: Aug 29, 2023

URL: https://aimc2023.pubpub.org/pub/tv38rj87

ABSTRACT

This paper explores three distance measures and three statistical tests for the comparison of music expressed in abc format. We propose a methodology that allows for an analysis at the level of corpora (is the "style" represented in a corpus the same as that in the another corpus?) as well as at the level of item (is the "style" of an item that of the "style" represented in a corpus?). We estimate distributions of distances between item pairs within and between corpora, and test hypotheses that the distributions are identical. We empirically test the impact of distance measure and statistical test using a corpus of Irish traditional dance music and a collection of tunes generated by a machine learning model trained on the same. The proposed methodology has a variety of applications, from computational musicology, to evaluating machine generated music.

Introduction

Musical style imitation (SI) systems analyze a collection of music exhibiting a particular style and then generate new music in that same style. The engineering of such systems has been an active area of research in artificial intelligence and music since the 1950s [1]. Early systems, like David Cope's Experiments in Musical Intelligence (EMI)[2] and Kemal Ebcioğlu's CHORAL [3], rely on the recombination of pre-composed material, or hand-crafted rules, to capture the style of composers such as Bach and Mozart. More recent systems apply data-driven approaches, such as deep learning techniques [4][5][6]. The development and use of such systems are also attracting a significant amount of investment to meet the need for royalty free and adaptive music in games and online videos for social networks.[1]

The output of SI systems can be evaluated in various ways, from comparing statistics [7][8][9][10][11], to music analysis and in situ testing [12][13][14], to listening tests [15][16]. Since human labor is expensive, and since the scale of material to be assessed can be orders of magnitude larger than possible for human scales of memory and attention, *and* since the development timeline of SI systems can be rapid, automating the "critique" of SI system output becomes necessary. One approach is *CAEMSI* [17], which compares two music corpora via permutation testing with a domain-independent distance metric, i.e., the Normalized Compression Distance. Yang and Lerch [8] proposes probabilistic measures of a variety of musically motivated features for comparing collections of MIDI-formatted music. While all of these methods focus on comparing collections, *StyleRank* [10] is able to assess the stylistic similarity of a particular item to a collection expressed in MIDI via machine learning models trained on features describing melodies and chords.

Other works related to symbolic folk music similarity include the study by Carvalho et al. [18], where they introduce a novel approach to encoding, analyzing, and modeling Iberian folk music. They propose a similarity-based interface that allows for an intuitive exploration of a database by visualizing songs on a 2-D plot using a dimensionality reduction algorithm. The similarity measures they use follow musical criteria such as melody, rhythm, and structure. Janssen et al. [19] conduct a study comparing various similarity measures appearing in music research across different domains. Their aim is to accurately identify melodic segments in folk songs. The measures they examine include correlation distance, city block distance, Euclidean distance, local alignment, wavelet transform, and structure induction. To evaluate the measures, they compare the generated phrase annotations against annotated phrase occurrences in a corpus of Dutch folk songs, using a majority vote from three annotators to determine agreement. Additionally, they investigate how the choice of music representation influences the success of these measures and assessed the robustness of the most effective ones when applied to subsets of the data.

In contrast to the above, we wish to design a methodology for determining what items of a music collection can be considered outliers, or in some sense uncharacteristic, of the styles exhibited by itself or another collection. In our approach, we adapt the domain-agnostic approach of CAEMSI to answer questions about items in collections. We focus in particular on the 365 double jigs of O'Neill's *The Dance Music of Ireland: 1001 Gems* (1907) [20], and several thousand "imitations" generated by the SI system folk-rnn (v2) [21][22]. The features used in [10] are not relevant to this kind of music, which does not explicitly feature chords or harmony. We instead stay in the domain of representations used by folk-rnn, i.e., either abc notation[2] or a 411-dimensional vectorization of the internal representation of the folk-rnn (v2) model for an output. In this paper, our exploration is limited to comparing abc-notated strings or the internal representation of folk-rnn (v2). However, it is important to note that our proposed method is not limited to these specific representations. It is designed to be domain-agnostic and can be applied to other popular representations such as MIDI or musicXML. We examine three distance measures and three statistical tests for comparing collections and their items. We present a variety of results, and discuss future directions of this work – both in the domain of music generation, but also computational musicology.

Methodology

Let $\mathcal{A} = (a_i, i = 1, \dots, n)$ be a collection of n tokenized tunes from O'Neill's 1001 [20], and $\mathcal{B} = (b_i, i = 1, \dots, m)$ be a collection of m tokenized tunes generated by folk-rnn (v2), each cast as a string. One item from $\mathcal{A}$ is "The Jolly Joker", tune no. 229 in O'Neill's 1001:

M:6/8 K:Cmaj C > c c c E F | G A F E F D | C > c c c E G | F A G F E D | C > c c E > c c | D > c c C > c c | e c A G F E | D A G F E D :| |: E D E C z C | C E G G F D | E D E C 3 | e c A A B c | E D E C z C | C E G G F E | F > A F E > G E | D > A G F E D :|

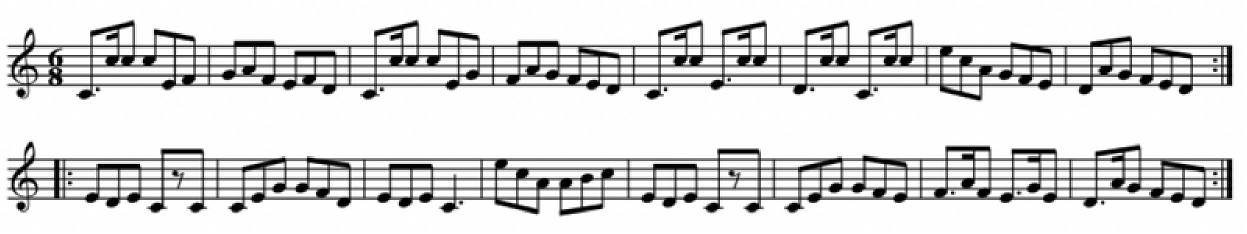

Image 1
“The Jolly Joker” from O’Neill’s “1001” (transposed)

and one from $\mathcal{B}$ is “No. 8091”:

M:6/8 K:Cmix |: G c c G 2 F | G c B G 2 F | D E F D E F | G F D F 3 | G c c G 2 F | G c B G 2 F | D E F B G F | D C B, C 3 :| |: c 2 d e c A | G A B G 3 | F D D B 2 G | F D D F D B, |c 2 d e c A | G A B G 3 | F D D B 2 D | D C B, C 3 :|

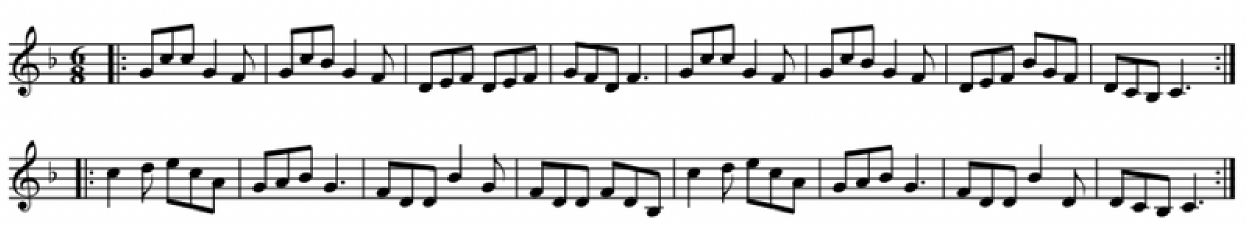

Image 2
folk-rnn (v2) tune No. 8091

Is this item of $\mathcal{A}$ characteristic of the rest of that collection? And is this item from $\mathcal{B}$ characteristic of what is in $\mathcal{A}$? Answers for these questions are hinted at in an analysis of these collections by four human experts at *The Ai Music Generation Challenge 2020* [14]. In particular, “The Jolly Joker” is deemed a very poor tune from O’Neill’s 1001 – a collection that “has plenty of poor and dull tunes” said one of the judges. And the folk-rnn (v2) tune No. 8091 won the first place award, with one judge saying, “If you heard [the tune] in a session, it wouldn’t stick out; it would feel comfortable with all the old tunes that are played”.

To answer these questions, we first define a distance metric between any pair of items within and between these collections, and then consider statistical methods for testing membership. Of course, the relationship between the musical content of these strings and any distance computed between them is debatable, but this is not necessarily a problem in our case since the folk-rnn model is merely modeling the syntax of tokenized symbolic transcriptions found in O’Neill’s and similar music, and it is the syntax exhibited by these collections and their items what we wish to test.

Possible distances between items and collections

The normalized compression distance (NCD) [23] can be used to compare the similarity between two strings. Essentially, it measures the length of the shortest binary program that can compute one string from another and vice versa. This is defined by

$$N(x,y) = \frac{K(x+y) - \min(K(x), K(y))}{\max(K(x), K(y))} \tag{1}$$

where $x+y$ is the concatenation of strings x and y, and $K(x)$ is the length of the shortest program that can compress string x. $K(x)$ is often computed using compression algorithms, such as zlib or lzma. Intuitively, if x and y are very similar, then the concatenation of the two should require a program not much larger than that required to compress either one of them. In this case, the numerator of $N(x,y)$ should be close to zero. The role of the denominator is to normalize the range of $N(x,y)$ to values within [0,1]. The NCD is technically not a distance metric since $N(x,x) \neq 0$ for any non-zero length x, but this is often considered inconsequential for making comparisons. It is also not symmetric. In our case, we symmetrize the measure by computing $[N(x,y) + N(y,x)]/2$. As an example, the distance between the two sequences above (with all spaces removed) is 0.5 for zlib and 0.64 for lzma. The distance between themselves is less than 0.06 for each algorithm.

The Levenshtein distance (also known as the edit distance) is a measure of the difference between two sequences [24]. It is the minimum number of single-character insertions, deletions, and substitutions required to transform one string into the other, where each operation is has a cost (in our case, we set each cost to be one). The normalized Levenshtein distance $L(x,y)$ is obtained by dividing the distance by the maximum length of strings x and y. For the two sequences above (with all spaces removed), the normalized Levenshtein distance is 0.55.

The cosine distance measures the angle between two non-zero-length vectors of the same dimensionality. The cosine distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as:

$$C(\mathbf{x},\mathbf{y}) = 1 - \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2} \tag{2}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. A value of 0 means that the two vectors point in the exact same direction, and a value of 1 means they are orthogonal. In our experiments, we create vectorized representations of a sequence by processing it with folk-rnn (v2) and saving the 137-dimensional softmax output for each step. We then compute the average, maximum and standard deviation of each dimension over the output steps and finally stack these statistics to create a vector of dimensionality 411. For the two sequences above, the cosine distance between them is 0.12.

Comparing collections

We now propose ways to compare collections. Denote $\mathbf{d}_{\mathcal{A}}$ as a vector of $n(n-1)/2$ distances of all unique pairs of items in $\mathcal{A}$, and $\mathbf{d}_{\mathcal{B}}$ a vector of $m(m-1)/2$ distances of all unique pairs of items in $\mathcal{B}$. Finally, denote $\mathbf{d}_{\mathcal{AB}}$ a length-mn vector of distances between all pairs of items of $\mathcal{A}$ and $\mathcal{B}$. We will compare these distances using various statistical tests of distributions fit to them. We will use the term ‘sample’ to refer to items in the distance vectors throughout the following paragraphs.

The Mann-Whitney U test (MW-test), also known as the Mann-Whitney-Wilcoxon or the Wilcoxon rank-sum test, is a non-parametric test used to compare two independent groups of samples to determine if they come from the same population [25]. Specifically, the test is used to examine if the two groups have the same median or if one group tends to have larger values than the other. An in-depth formulation of the MW-test is found in [26]. The MW-test makes four assumptions: the data must be ordinal, the two groups must be independent, each sample must be mutually independent, and the two distributions should have the same shape (although this last assumption is not a strict requirement). The null hypothesis is that the two groups come from the same population, which is equivalent to saying that the medians of the two groups are equal. This null hypothesis implies that the two groups have the same probability distribution, without specifying the common distribution.

The Kolmogorov-Smirnov test (KS-test), on the other hand, is used as a goodness-of-fit test to check if a certain set of continuous values follows a certain theoretical distribution or to compare if samples from two groups have the same distribution [27]. This test relies on the maximum absolute difference between the cumulative distribution functions of samples between the two groups. Unlike the Mann-Whitney U test, the K-S test can tell if a sample follows a certain distribution or not. When comparing two samples, the K-S test is sensitive to both the shape and location of the two samples, making it robust for comparing if the two distributions are the same. Under the null hypothesis the maximum absolute difference between the cumulative distribution functions follows a Kolmogorov-Smirnov distribution.

The key difference between the Mann-Whitney U and K-S tests is that the Mann-Whitney U test compares the two groups on the basis of a measure of central tendency (usually the median), while the K-S test compares on the basis of statistical distance and is usually used as a goodness-of-fit test to check if the data follows a certain distribution. While the Mann-Whitney U test is applicable when the data is ordinal and independent, the K-S test can be used when the data is continuous and can follow any distribution.

Finally, the Kruskal-Wallis H test (KW-test), considered an extension of the Mann-Whitney U test, is a non-parametric statistical test used to compare the medians of two or more independent samples [28]. It ranks the samples in all groups combined and calculates a test statistic that measures the difference between the ranked medians of the groups, without requiring the assumption of normality or equal variances.

Comparing an item to a reference collection

We now examine how an item x can be tested as belonging to the collection $\mathcal{A}$. We first estimate the *reference* distribution of elements in $\mathbf{d}_{\mathcal{A}}$ using kernel density estimation. Next, we calculate the pairwise distances between x and all the items in $\mathcal{A}$ which we refer to as $\mathbf{d}_{\mathcal{A}x}$ and estimate the *calculated* distribution of these distances using the same kernel density estimation technique. To determine whether x belongs to $\mathcal{A}$ we compute the probability mass of the intersection of the two distributions. Image 3 shows an example using the two sequences above, compared using the normalized Levenshtein distance.

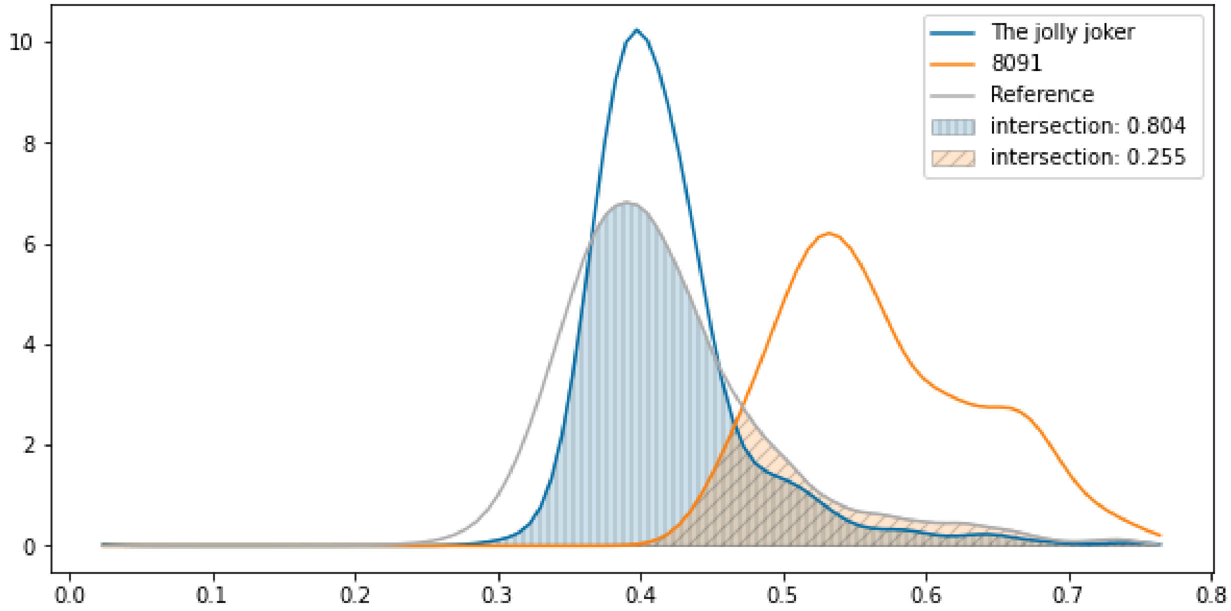

Image 3
Intersections between “The Jolly Joker”, folk-rnn (v2) tune No. 8091, and the reference set using the normalized Levenshtein distance to calculate the pairwise distances.

We can also compare $\mathbf{d}_{\mathcal{A}}$ and $\mathbf{d}_{\mathcal{A}x}$ using a MW-test or a KS-test. In the case of the two sequences above, and having as null hypothesis that the distributions of $\mathbf{d}_{\mathcal{A}}$ and $\mathbf{d}_{\mathcal{A}x}$ are the same, we reject the null hypothesis (using MW-test) for all distances in the case of tune No. 8091. For “The Jolly Joker”, the p-value is above the alpha risk of 5 percent (0.05) for all distances except for ncd_zlib, therefore we retain the null hypothesis.

Application and evaluation

We now evaluate the distance metrics above and the statistical tests for the collection of O’Neill’s 365 double jigs ($\mathcal{A}$), and a set of 365 tunes generated by folk-rnn (v2) ($\mathcal{B}$).

Pairwise distances within and between collections

We calculate $\mathbf{d}_{\mathcal{A}}$, $\mathbf{d}_{\mathcal{B}}$ and $\mathbf{d}_{\mathcal{AB}}$ for each of the distance measures. Image 4 shows the set of all normalized Levenshtein distances as a matrix partitioned by two overlaid lines. (The plot resulting using NCD appears similar.) Each element is a distance between two items, where the diagonal is the distance between an item and itself. The top-left matrix is all distances within $\mathcal{A}$, and the bottom-right is all distances within $\mathcal{B}$. The top-right (bottom-left) matrix is all distances between items in $\mathcal{A}$ with $\mathcal{B}$. Small distance is mapped to white, and large distance is mapped to black. Image 5 shows the same for the Cosine distances.

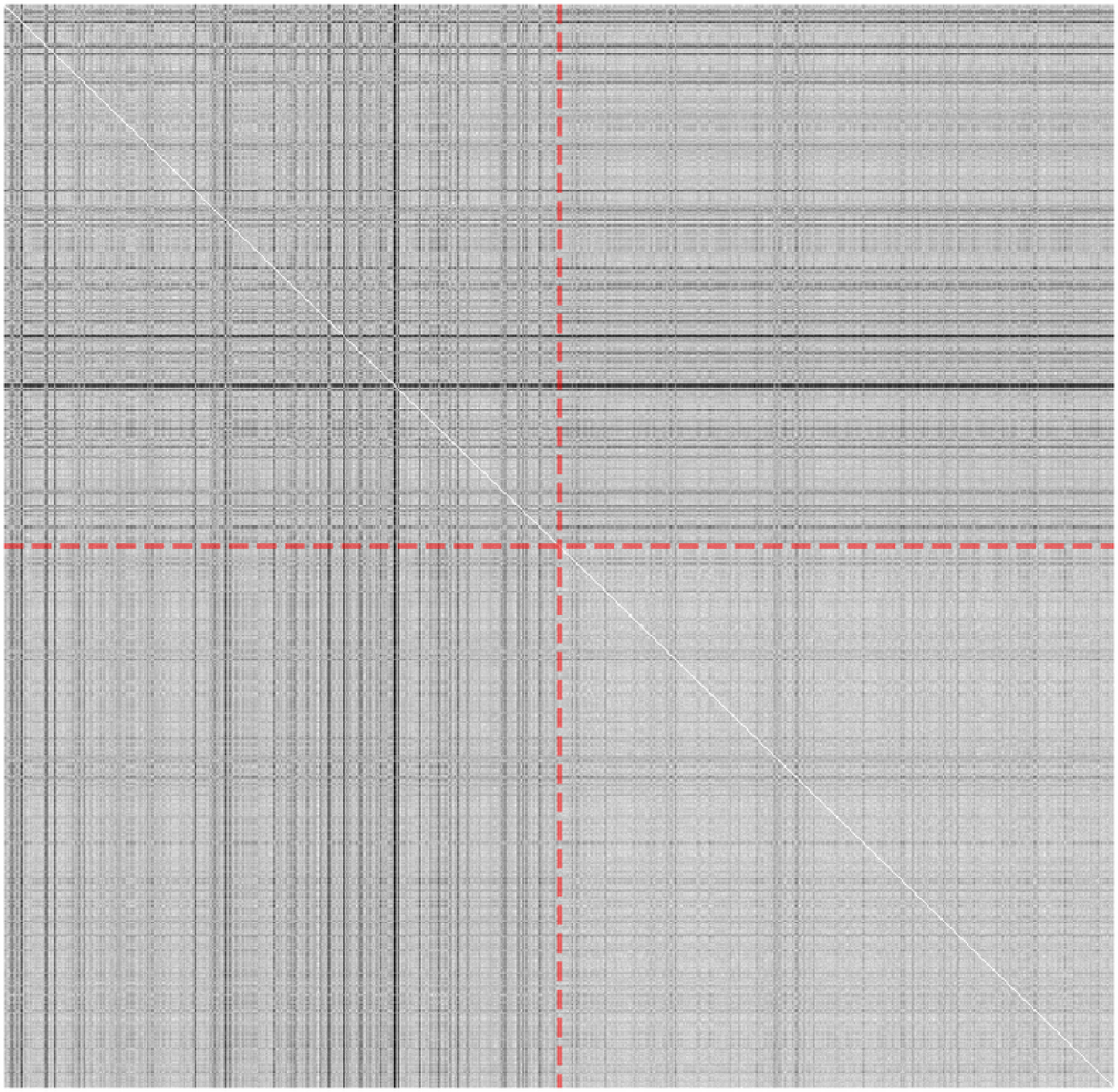

Image 4

Levenshtein distances of items within $\mathcal{A}$ (top-left), within $\mathcal{B}$ (bottom-right), and between $\mathcal{A}$ and $\mathcal{B}$ (top-right and bottom-left).

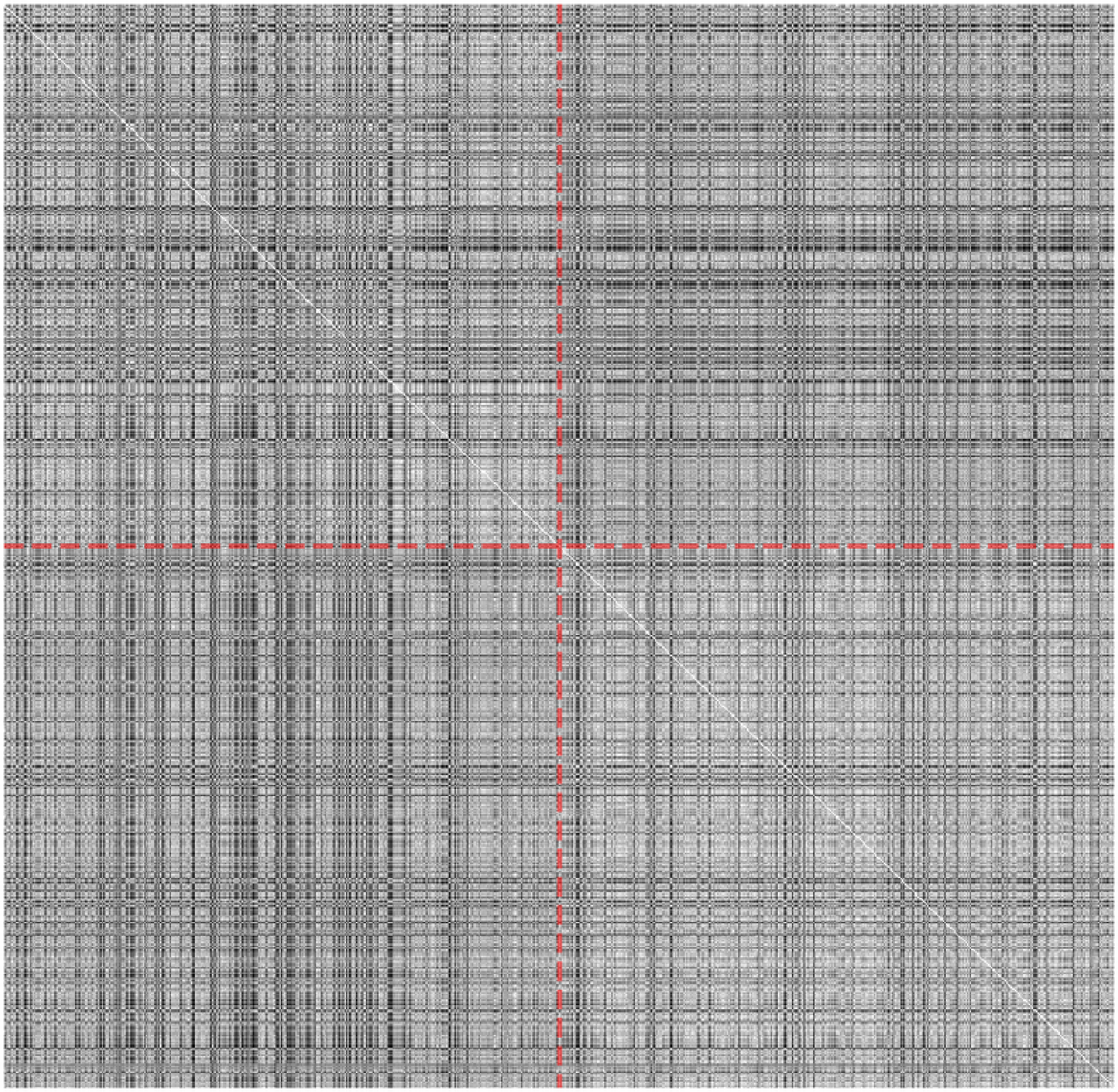

Image 5

Cosine distances of items within $\mathcal{A}$ (top-left), within $\mathcal{B}$ (bottom-right), and between $\mathcal{A}$ and $\mathcal{B}$ (top-right and bottom-left).

Immediately clear from these images is the differences within the collections. The items in $\mathcal{B}$ appear much more similar to each other than the items in $\mathcal{A}$. Some tunes in $\mathcal{A}$ are quite different to all the rest. The items in $\mathcal{A}$ that are furthest away from all others with respect to the normalized Levenshtein distance are those tunes that are very long, e.g., Nos. 257, 224 and 12. We also find very small normalized Levenshtein distance between tunes in $\mathcal{A}$ that are known duplicates, e.g., Nos. 16 and 358, and Nos. 59 and 156.

Comparing a collection to a reference collection

We now perform a variety of statistical tests on the distributions of the distances for a random 80% sampling of $\mathcal{A}$ and $\mathcal{B}$ (using IBM SPSS Statistics). After the sampling, we are left with $n = 292$ items from $\mathcal{A}$ and $m = 292$ items from $\mathcal{B}$, resulting in $n(n-1)/2 = 4286$ and $nm = 85264$ distances in $\mathbf{d}_{\mathcal{A}}$ and $\mathbf{d}_{\mathcal{AB}}$ respectively.

Tables 1 and 2 present descriptive statistics of the computed distances, giving a view of how the distance metrics differ. At a first glance, it seems like the less reliable distance would be the cosine distance, as a higher standard deviation indicates more variability and less consistency in the measurements, making it harder to predict.

Table 1

Distance	N	Mean	Std. Dev.	Minimum	Maximum
ncd_zlib	42486	0.674	0.062	0.148	0.905
ncd_lzma	42486	0.486	0.067	0.051	0.758
Levenshtein	42486	0.418	0.077	0.023	0.764
cosine	42486	0.202	0.126	<0.001	0.636

Table 1. Descriptive Statistics of $d_{\mathcal{A}}$.

Table 2

Distance	N	Mean	Std. Dev.	Minimum	Maximum
ncd_zlib	85264	0.664	0.060	0.467	0.904
ncd_lzma	85264	0.496	0.053	0.311	0.770
Levenshtein	85264	0.573	0.090	0.286	0.865
cosine	85264	0.205	0.119	0.006	0.643

Table 2. Descriptive Statistics of $d_{\mathcal{AB}}$.

For the KW-test and the KS-test we use a two-tailed alternative, meaning that the only thing that we test for is whether the means of the distributions underlying the samples are unequal. The results in Table 3 indicate that most pairwise distances are useful to differentiate between $d_{\mathcal{A}}$ and $d_{\mathcal{AB}}$, this suggest that the style represented in $\mathcal{A}$ is not the same as the one in $\mathcal{B}$.

However, when the alternative hypothesis is that the distribution underlying $d_{\mathcal{A}}$ is stochastically less than the distribution underlying $d_{\mathcal{AB}}$, the MW-test shows that for ncd_zlib the p-value is above the alpha risk of 5 percent (0.05), so the null hypothesis cannot be rejected. This would indicate that, even though ncd_zlib is useful to differentiate between collections (as indicated by the two-tailed tests results), the distance between items within collection $\mathcal{A}$ is stochastically bigger than the distance between items across collections.

Table 3

Distance	ncd_zlib	ncd_lzma	Levenshtein	cosine
Kruskal-Wallis statistic	819.170	1241.844	58043.036	96.965
Asymp. Sig. (p-value)	<0.001	0.115	<0.001	<0.001
Kolmogorov-Smirnov statistic	15.794	27.607	117.831	11.542
Asymp. Sig. (p-value)	<0.001	<0.001	<0.001	<0.001

Table 3. Results of the two-tailed Kruskal-Wallis test and the Kolmogorov-Smirnov test, comparing $d_{\mathcal{A}}$ and $d_{\mathcal{AB}}$.

If we look at the statistics, normalized Levenshtein has the highest value, which indicates that it is the distance metric that is the best at differentiating between $\mathcal{A}$ and $\mathcal{B}$.

Comparing an item to a reference collection

In order to compare an element to collections $\mathcal{A}$ and $\mathcal{B}$, we follow the procedure described previously. This gives us four mean-area-of-intersection values. We define $\mu_{\mathcal{AA}}$ as the mean of the area intersection between the distribution of $\mathbf{d}_{\mathcal{A}}$ and the distributions $\mathbf{d}_{\mathcal{A}a_i}$ for all $a_i \in \mathcal{A}$. Similarly, $\mu_{\mathcal{AB}}$ is the mean of the area intersection between the distribution of $\mathbf{d}_{\mathcal{A}}$ and the distributions $\mathbf{d}_{\mathcal{A}b_i}$ for all $b_i \in \mathcal{B}$. Then, $\mu_{\mathcal{BA}}$ is the mean of the area intersection between the distribution of $\mathbf{d}_{\mathcal{B}}$ and the distributions $\mathbf{d}_{\mathcal{B}a_i}$ for all $a_i \in \mathcal{A}$. Finally, $\mu_{\mathcal{BB}}$ is the mean of the area intersection between the distribution of $\mathbf{d}_{\mathcal{B}}$ and the distributions $\mathbf{d}_{\mathcal{B}b_i}$ for all $b_i \in \mathcal{B}$. The estimated means are the following:

Table 4

Distance	ncd_zlib	ncd_lzma	Levenshtein	cosine

$\mu_{\mathcal{A}\mathcal{A}}$	0.779	0.756	0.787	0.804
$\mu_{\mathcal{A}\mathcal{B}}$	0.692	0.669	0.626	0.735
$\mu_{\mathcal{B}\mathcal{A}}$	0.799	0.796	0.861	0.804
$\mu_{\mathcal{B}\mathcal{B}}$	0.765	0.768	0.770	0.794

Table 4. Mean-area-of-intersection for each distance, for all combinations of reference distributions and calculated distributions.

We can then classify unseen elements by labeling them depending on which mean area intersection is the estimated area of intersection it is closest to. We evaluate this procedure by calculating several metrics on the test set, including accuracy, precision, recall, and F1 score. The results are consistent with the statistical tests, as normalized Levenshtein outperforms the other distances.

Table 5				
Distance	ncd_zlib	ncd_lzma	Levenshtein	cosine
Acc	0.582	**0.596**	0.582	0.596
Prec	0.561	**0.571**	0.552	0.565
Recall	0.753	0.767	**0.877**	0.836
F1	0.643	0.655	**0.677**	0.674

Table 5. Evaluation of the classifier based on the mean-area-of-intersection on the test set.

Visualizing the collections

To visually compare the two collections of tunes, we use t-distributed Stochastic Neighbor Embedding (t-SNE) [29] and Uniform Manifold Approximation and Projection (UMAP) [30] algorithms. We can represent the items in $\mathcal{A}$ and $\mathcal{B}$ as four-dimensional data, where each dimension is one of the mean pairwise distances discussed previously. This is equivalent to saying that each dimension of the data is the mean of $\mathbf{d}_{\mathcal{A}}$ and $\mathbf{d}_{\mathcal{B}}$ calculated with one of the four distances studied in this paper. Once we have this representation, we make use of t-SNE and UMAP to convert the data to a 2-dimensional embedded space in order to visualize them.

We can use the visualizations to identify “typical” tunes and “outliers” for both $\mathcal{A}$ and $\mathcal{B}$, as well as which tunes are more similar. For example, if we look at Image 7, we can see the outlier “Morgan Rattler” (from $\mathcal{A}$), which is the longest jig in the collection with a length of 1456. The distance metrics that we are using are highly dependent on the length of the abc string. The tune no. 1923 (from $\mathcal{B}$),

M:6/8 K:Cmaj e f g e c c | e c g e c c | f a f e f d | e f g a g f | e f g e c c | e c g e c c | f e f d B d | c e c c 2 g | c' 2 g g a b | c' g g g f e | f 2 f a b f | f g a b a g | c' g g g f e | a f a g e c | e f g a g f | e f d c 2 a :|

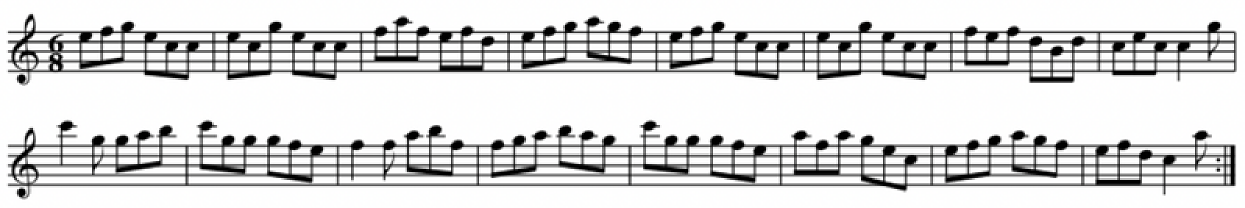

Image 6
folk-rnn (v2) tune No. 1923

is missing the repeat signs, which are characteristic of jigs. If we take a look at Image 8, it becomes more evident that tune no. 1923 is an outlier in $\mathcal{B}$ and "Morgan Rattler" is still an outlier in the main cluster of $\mathcal{A}$. We can also see that tunes Nos. 16 and 358 in $\mathcal{A}$ (known duplicates) can be found very close to each other in both visualizations.

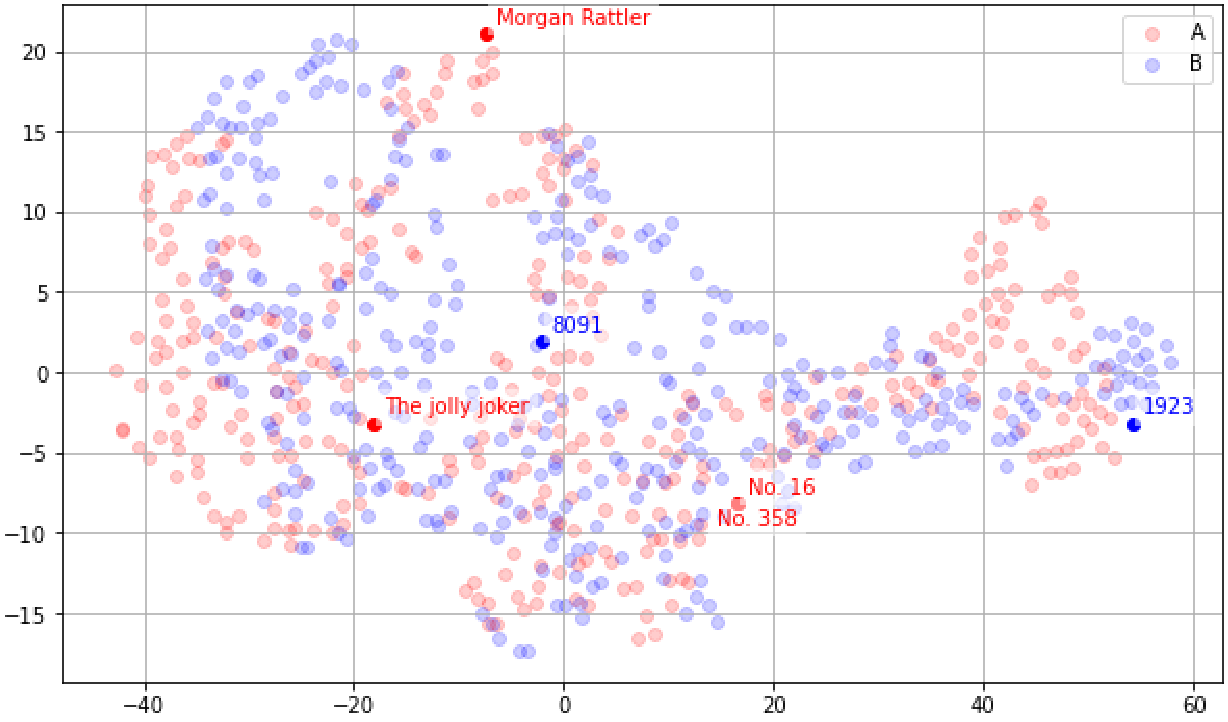

Image 7
Visualization of the tunes using t-SNE

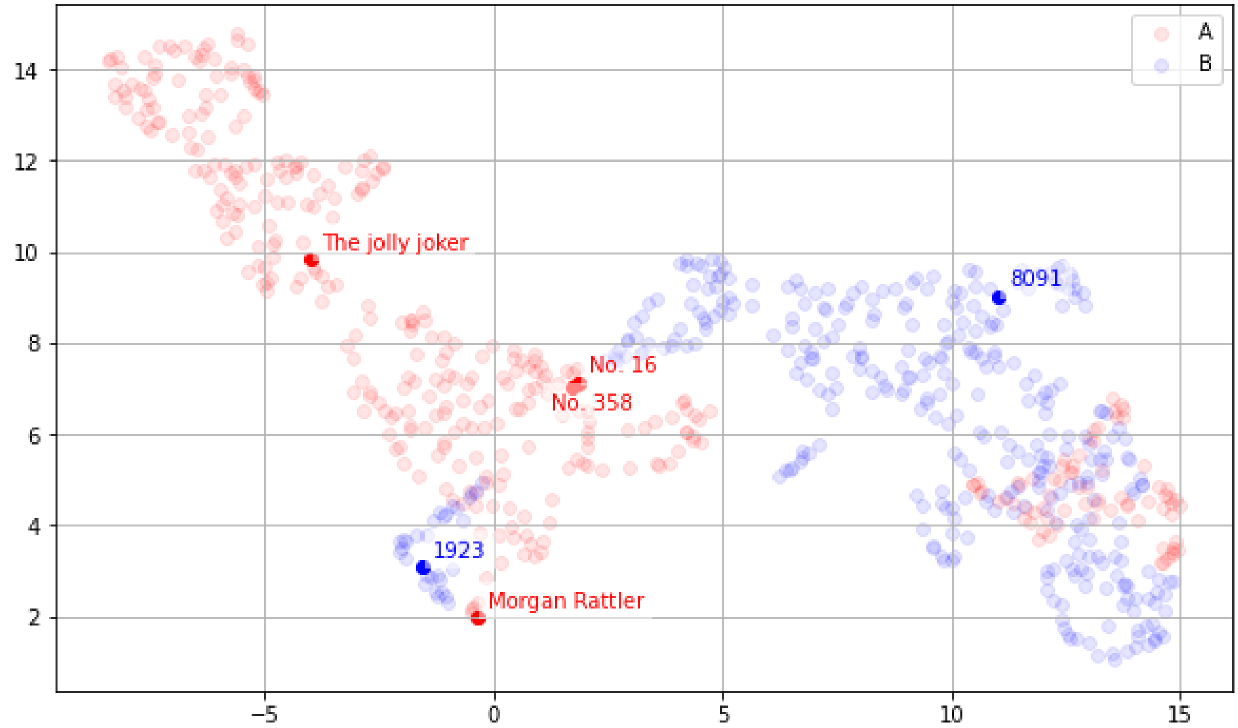

Image 8
Visualization of the tunes using UMAP

Much more work can be done in this direction, for instance, isolating portions of a collection that are overlapping with a reference corpus in order to select stylistically similar items.

Discussion and future work

Our paper examines various distance measures and statistical tests to compare items in corpora expressed with the abc notation format. These comparison methods combine elements of other methods, such as CAEMSI [17] and StyleRank [10], but we aim to do more than just compare collections. One objective is to provide a method for curation, filtering one collection in reference to another. Another is to identify “typical” elements and “outliers” within a collection. These approaches could be used in the machine learning pipeline as a proxy evaluation of a sequence model at points in its training. They could also augment musicological studies done by hand of tune collections, as done on O’Neill’s “1001” by Doherty [31] for instance. By using these studies as a benchmark, we can further evaluate the results obtained from the same collection in a more musically meaningful manner.

At the moment the distance measures we are using are focused only on sequences of tokens, or comparing statistics of vectors. One problem with the Levenshtein distance could be a sensitivity to musically irrelevant transformations, such as implicit repeat signs, first and second endings, and an anacrusis. In the future, we would like to explore domain-dependent distance measures, and considerations of the form of these kinds of tunes: that they have parts, and the relationship between the parts could provide information about the membership to a collection. Other future work includes exploring in-depth the impact of the choices made when comparing distributions, like the choice of kernel distribution estimation or the area of intersection. Furthermore, we intend to expand the scope of our study to include other music “styles”. We also plan to investigate BERT representations and develop a Fréchet Symbolic music distance. The goal would be to develop a robust method for comparing items in a collection to a reference corpus of real tunes in a symbolic format. Another possible extension is to apply principles of explainability to identify what it is about a tune that makes it an outlier. Finally, our research may be employed to enhance the effectiveness of machine learning algorithms that analyze and compare items in a collection to a corpus.

Acknowledgments

This work was supported by the grant ERC-2019-COG No. 864189 MUSAiC: Music at the Frontiers of Artificial Creativity and Criticism.

Footnotes

[1] For instance, see https://www.musicbusinessworldwide.com/tiktok-parent-bytedance-buys-ai-music-company-jukedeck/ ↩

[2] See https://abcnotation.com/wiki/abc:standard:v2.1 ↩

References

- Ariza, C. (2009). The Interrogator as Critic: The Turing Test and the Evaluation of Generative Music Systems. *Computer Music Journal*, *33*(2), 48–70. ↩
- Briot, J.-P., Hadjeres, G., & Pachet, F. (2019). *Deep Learning Techniques for Music Generation*. Springer. ↩
- Carvalho, N., Gonzalez-Gutierrez, S., Merchan Sanchez-Jara, J., Bernardes, G., & Navarro-Cáceres, M. (2021, July 28). Encoding, Analysing and Modeling I-Folk: A New Database of Iberian Folk Music. *8th International Conference on Digital Libraries for Musicology*. DLfM ’21: 8th International Conference on Digital Libraries for Musicology. https://doi.org/10.1145/3469013.3469023 ↩
- Cope, D. (1992). Computer modeling of musical intelligence in EMI. *Computer Music Journal*, *16*(2), 69–83. ↩
- Doherty, S. (2022). Melodic structures in the double jigs of o’neill’s the dance music of ireland: 1001 gems (1907). *J. Soc. Musicology in Ireland*, 19–45. ↩
- Dubnov, S., Assayag, G., Lartillot, O., & Bejerano, G. (2003). Using machine-learning methods for musical style modeling. *Computer*, *36*(10), 73–80. ↩
- Ebcioğlu, K. (1988). An expert system for harmonizing four-part chorales. *Computer Music Journal*, *12*(3), 43–51. ↩
- Ens, J., & Pasquier, P. (2018, June). A Cross-Domain Analytic Evaluation Methodology for Style Imitation. *Proc. Int. Conf. Computational Creativity*. ↩

- Ens, J., & Pasquier, P. (2018). CAEMSI: A cross-domain analytic evaluation methodology for style imitation. *ICCC*, 64–71. ↩
- Ens, J., & Pasquier, P. (2019). Quantifying Musical Style: Ranking Symbolic Music based on Similarity to a Style. *Proc. ISMIR*. ↩
- Ens, J., & Pasquier, P. (2020). Improved Listening Experiment Design for Generative Systems. *Proc. Joint Conf. AI Music Creativity*. ↩
- Ens, J., & Pasquier, P. (2020). MMM : Exploring Conditional Multi-Track Music Generation with the Transformer. *arXiv*. ↩
- Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Hawthorne, C., Dai, A. M., Hoffman, M. D., & Eck, D. (2018). Music Transformer: Generating Music with Long-Term Structure. *arXiv Preprint arXiv:1809.04281*. ↩
- Janssen, B., van Kranenburg, P., & Volk, A. (2017). Finding Occurrences of Melodic Segments in Folk Songs Employing Symbolic Similarity Measures. *Journal of New Music Research*, *46*(2), 118–134. https://doi.org/10.1080/09298215.2017.1316292 ↩
- Kraska-Miller, M. (2013). *Nonparametric statistics for social and behavioral sciences*. Crc Press. ↩
- Levenshtein, V. (1965). Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information Transmission*, *1*, 8–17. ↩
- Li, M., Chen, X., Li, X., Ma, B., & Vitányi, P. M. (2004). The similarity metric. *IEEE Transactions on Information Theory*, *50*(12), 3250–3264. ↩
- Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 50–60. ↩
- Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, *46*(253), 68–78. ↩
- McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv E-Prints 1802.03426*. ↩
- O’Neill, F. (1907). *The Dance Music of Ireland: O’Neill’s 1001*. Chicago. ↩
- Shier, R. (2004). Statistics: 2.3 the mann-whitney u test. *Mathematics Learning Support Centre. Last Accessed*, *15*, 2013. ↩
- Sturm, B. L. T. (2021). An Artificial Critic of Irish Double Jigs. *Proc. AI Music Creativity*. ↩
- Sturm, B. L. T., & Maruri-Aguilar, H. (2021). The Ai Music Generation Challenge 2020: Double Jigs in the Style of O’Neill’s “1001.” *Journal of Creative Music Systems*. ↩
- Sturm, B. L., & Ben-Tal, O. (2017). Taking the Models back to Music Practice: Evaluating Generative Transcription Models built using Deep Learning. *J. Creative Music Systems*, *2*(1). ↩
- Sturm, B. L., & Ben-Tal, O. (2017). Taking the Models back to Music Practice: Evaluating Generative Transcription Models built using Deep Learning. *J. Creative Music Systems*, *2*(1). ↩
- Sturm, B. L., Ben-Tal, O., Monaghan, Ú., Collins, N., Herremans, D., Chew, E., Hadjeres, G., Deruty, E., & Pachet, F. (2018). Machine Learning Research that Matters for Music Creation: A Case Study. *J. New Music*

Research, *48*(1), 36–55.↩

- Sturm, B. L., Santos, J. F., Ben-Tal, O., & Korshunova, I. (n.d.). Music transcription modelling and composition using deep learning. *1st Conf. Computer Simulation of Musical Creativity*. ↩
- Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(11). ↩
- Yang, L.-C., & Lerch, A. (2018). On the evaluation of generative models in music. *Neural Computing and Applications*. ↩
- Yin, Z., Reuben, F., Stepney, S., & Collins, T. (2021). “A good algorithm does not steal – it imitates”: The originality report as a means of measuring when a music generation algorithm copies too much. *Proc. EvoMUSART*. ↩