An Artificial Critic of Irish Double Jigs

Bob L. T. Sturm*

Tal, Musik och Hörsel, School of Electrical Engineering and Computer Science, KTH Royal Institute of Engineering, Stockholm Sweden bobs@kth.se

Abstract. This paper describes a component of the music generation system that produced an award-winning tune at The Ai Music Generation Challenge 2020. This challenge involved four Irish traditional music experts judging 35 tunes generated by seven systems in reference to a recognised collection of a specific kind of dance music. The winning system uses an "artificial critic" that accepts or rejects a generated tune based on a variety of criteria related to metric structure and intervalic content. Such an artificial critic can help one explore massive generated music collections, as well as synthesise new training music collections.

Keywords: Music generation, Irish traditional music

1 Introduction

The Ai Music Generation Challenges aim to improve the engineering of music generation systems by involving music practitioners in specific music traditions. The 2020 challenge (B. L. T. Sturm & Maruri-Aguilar, 2021) posed the following specific task to researchers: "Build an artificial system that generates the most plausible double jigs, as judged against the 365 published in F. O'Neill 'The Dance Music of Ireland: O'Neill's 1001"' (1907). This collection from the turn of the 20th century is recognised for its historical significance (Breathnach, 1971), and many tunes in the collection are still played today. A *double jig* has a rhythm similar to speaking the phrase, "DIddly DIddly", and consists of at least two repeated eight-measure parts. Each part is typically built from shorter phrases, and often the parts relate to one another. Figure 1 shows one double jig from O'Neill's "1001" possessing such characteristics. O'Neill's collection of 365 double jigs (abbreviated herein as "O'Ndj") is quite uniform in terms of structure and melodic and harmonic content, and so the implicit syntax of the collection should be within reach of machine learning.

Each participant of the 2020 challenge submitted a collection of 10,000 generated "tunes". Five tunes selected at random from each collection were independently evaluated by four Irish traditional music experts in reference to O'Ndj according to a variety of criteria. Each tune must pass at least four specific rejection criteria: it cannot be plagiarised; its rhythm is characteristic of a double jig;

* This paper is an outcome of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 864189).

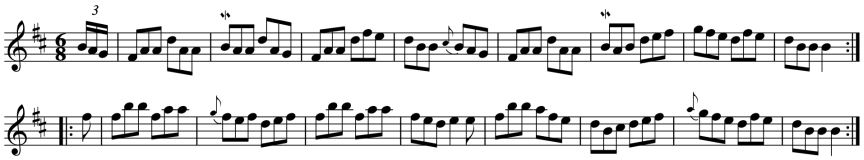

Fig. 1: The double jig *The Connachtman's Rambles* (#218) as it appears in O'Neill's "1001" (1907).

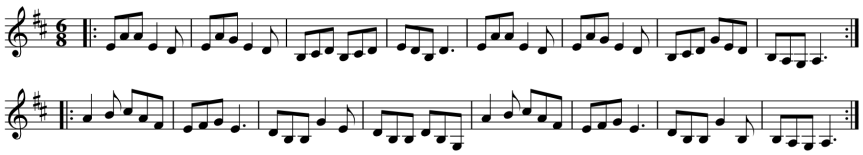

Fig. 2: First prize of *The Ai Music Generation Challenge 2020* was awarded to this double jig from a collection generated by *folk-rnn* (v2) using beam search sampling and assembled by an "artificial critic" with reference to the double jigs in O'Neill's "1001" (1907).

its pitch range is characteristic; and its mode and accidentals are characteristic. Tunes that pass these criteria are then considered more closely by the judge along five dimensions: melody, structure, playability, memorability, and interestingness. The judges met to discuss their evaluations, and singled out two of 35 tunes generated by seven systems. Second prize was awarded to a tune generated by the benchmark system, *folk-rnn* (v2) (B. L. Sturm & Ben-Tal, 2017). First prize was awarded to the tune notated in Fig. 2, which comes from a collection assembled by an "artificial critic" from tunes generated by a version of *folk-rnn* (v2) sampling pairs of tokens with beam search.

Among the 23,636 transcriptions used to train *folk-rnn* (v2) exists most of O'Ndj — along with thousands of tunes accompanying other dance styles, e.g., reels. Since the system already generates transcriptions of appreciable quality with respect to Irish traditional dance music (B. L. Sturm & Ben-Tal, 2017), it might be able to generate double jigs having the qualities of O'Ndj. Instead of attempting to fine tune the machine learning model on the very small O'Ndj, our strategy was to engineer a critic that picks tunes generated by the model that are most characteristic in reference to O'Ndj. In the following, we describe the engineering and development of this critic and demonstrate its application in the context of the 2020 challenge. We then discuss the extension and applicability of such critics to other ends.

2 An artificial critic for O'Neill's double jigs

Our critic employs four consecutive stages to iteratively select tunes in creating a collection, moving from coarse to finer musical considerations. These stages

are informed by the evaluation procedure of the challenge and rules inferred from O'Neill's "1001", and are ordered to reduce computational cost. The first stage rejects tunes with metric structures uncharacteristic of double jigs. The second stage rejects tunes with melodic structures uncharacteristic of O'Ndj. The third and most computationally expensive stage detects duplication, not only of material in O'Ndj, but also the training data of *folk-rnn* (v2) as well as generated tunes in the growing the collection. In the final stage, the critic attempts to transpose a plausible and original tune to a characteristic mode without exceeding melodic range constraints. These stages involves rejection criteria with specific parameters. We tune these parameters using a leave-one-out test with O'Ndj, where each tune is treated as a candidate and the reference collection is the remainder, and the aim is to not reject the tune.

The printed transcriptions in O'Neill's "1001" (1907) have been digitised in a textual notation format known as *abc notation* (Walshaw, 2021).[1] We make these transcriptions comparable with any generated by *folk-rnn* (v2) by transposing them to have a root of C, removing irrelevant fields and performance indications, and finally tokenising them using the vocabulary of *folk-rnn* (v2) (B. L. Sturm, Santos, Ben-Tal, & Korshunova, 2016). We use the music21 library (Cuthbert & Ariza, 2010) to process each abc transcription.

2.1 Stage 1: Plausibility of metric structure

The critic converts a tokenised tune into two sequences describing its metric structure. The *measure token sequence* of a transcription is the sequence of extracted measure tokens. For example, the measure token sequence of the transcription in Fig. 1 is `(|,|,|,|,|,|,|,|,:|,|:,|,|,|,|,|,|,|,:|)`. These are just the bar lines and repeat bars of the tokenised transcription. The *rhythm sequence* of a transcription is a representation of its measure note placement. This is created by replacing every pitch token with the symbol `s`, replacing triplet semiquavers and semiquaver pairs with quavers, removing broken rhythm symbols `>` and `<`, preserving all other duration tokens, and segmenting by measure lines. The rhythm sequence of the last two measures of the transcription in Fig. 1, is `(|,s,s,s,s,s,s,|,s,s,s,s2,|)`.

The critic compares the measure token sequence and rhythm sequence of a candidate tune to those extracted from O'Ndj. The critic deems that a candidate transcription has a *plausible metric structure* if: 1) it can be contiguously segmented into an integer number of whole eight-measure parts; 2) less than 11/16 of its rhythm sequence does not match the single jig patterns `(s2,s,s2,s)`, `(s2,s,s,s,s)`, `(s2,s,s3)`, `(s3,s2,s)`, and `(s3,s3)`; 3) it does not contain a note with duration longer than a dotted crotchet; and 4) its measure token sequence appears in O'Ndj. Each tune in O'Ndj meets the first three conditions, but 33 of the double jigs would be rejected by the fourth condition because they have a unique measure token sequence.

[1] One digitised collection is here: `http://www.oldmusicproject.com/oneils1.html`

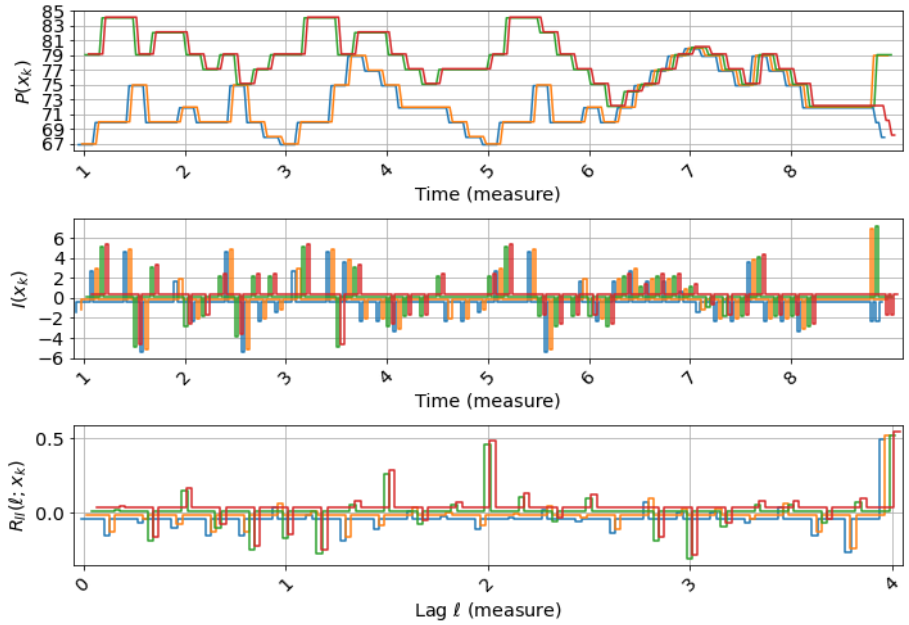

Fig. 3: For each 8-measure segment of the double jig in Fig. 1 starting after the anacrusis: the pitch-time series $P(x_k)$ (top), interval-time series $I(x_k)$ (middle), and interval periodicity $R_{II}(\ell; x_k)$ (bottom). Small offsets added for readability.

2.2 Stage 2: Plausibility of melodic self-similarity

For a tune passing stage 1, the critic extracts a descriptor of its intervalic content. It makes all repetitions explicit, partitions it into 8-measure contiguous segments coinciding with measure lines (accounting for any anacrusis), and transforms the sequences into a uniformly sampled series representing the melody, which we call the *pitch-time* series. To minimise redundancy, the sampling period of this series is one sixth of a quaver, which allows a whole number of samples to represent the shortest note duration in O'Ndj (a triplet semiquaver in 2 samples), as well as a semiquaver (3 samples). Each eight-measure segment is thus transformed into a series of length $6 \cdot 6 \cdot 8 = 288$ samples. The critic uses a sample-and-hold procedure where pitches are held until a different pitch occurs. The pitch-time series of the jig in Fig. 1 is shown at the top of Fig. 3.

The critic performs first-order differencing of the pitch-time series to create an *interval-time* series. Unless there is an anacrusis, the first interval of the first segment is zero; otherwise the first interval is the difference between the first pitch of the segment and last pitch of the anacrusis. The interval-time series of the two parts of the jig in Fig. 1 are shown in Fig. 3(middle). Finally, the critic computes the *interval periodicity* series of a transcription segment by performing a circular autocorrelation of the interval-time series, and normalising by the

value at zero lag. Since the circular autocorrelation is symmetric only half of it is kept (without the value at zero lag), resulting in a series of 144 samples. Figure 3(bottom) shows the intervalic periodicity series of the jig in Fig. 1. This shows how the two melodic parts have different self-similarities with respect to intervalic content: its two parts have high self-similarity at a lag of four measures, but its B part also has self-similarity at a lag of two measures.

The critic now computes a measure of similarity for a candidate tune in relation to O'Ndj by comparing interval periodicities. For each interval periodicity extracted from the candidate tune, the critic finds the Euclidean distance to the closest interval periodicity extracted from O'Ndj. If the largest of these distances exceeds some threshold then the critic rejects the candidate tune. The leave-one-out test with O'Ndj shows the mean distance is 0.525, and if the maximum distance is set to at least 1.1 then no tune from O'Ndj is rejected. Setting the threshold to 0.85 rejects only five tunes from O'Ndj.[2]

2.3 Stage 3: Duplicate intervals detection

For a tune that passes the second stage, the critic first measures the greatest amount of its intervalic content matching that of the 365 tunes in O'Ndj. The critic does this by counting the number of intervals matching in four whole-measure segments as the resolution of a quaver. Specifically, each interval-time series of eight measures is downsampled to quaver resolution and segmented using a window of four measures length and a hop of one whole measure (starting on the first measure, disregarding any anacrusis). This creates four interval-time sub-series. Those of the candidate tune are compared with others from O'Ndj, and the number of matching intervals is counted. If for a candidate tune the critic finds no duplication in O'Ndj, the critic then performs the same comparison to the 5,940 tunes in 6/8 meter in the training data of the model (v2). This is an expensive operation, involving more than 50,000 comparisons for each interval-time series of a candidate tune. To reduce computation when checking in v2 the critic compares only the pitch-time series to subset of of v2 found by k-means clustering. More specifically, the downsampled pitch-time series of v2 is preprocessed by k-means clustering with $k = 6$ and Euclidean distance, and the pitch-time series of the candidate tune is compared with those in the appropriate cluster. Setting the threshold for duplication to be more than 20 quavers, the leave-one-out test with O'Ndj finds 16 instances of duplication within the collection, which are confirmed by inspection.[3]

2.4 Stage 4: Editing

The critic now attempts to transpose the tune to a mode characteristic of O'Ndj while remaining in the pitch range G below middle C to E two octave above middle C. If the mode of the tokenised tune is C major, then it randomly transposes

[2] Double jigs (and distance) #8 (0.95), 76 (0.87), 101 (1.09), 136 (0.91), 200 (0.85).
[3] These duplications are (16,358), (26,113), (42,325), (59.156), (88,267), (134,296), (194,302), (261,334).

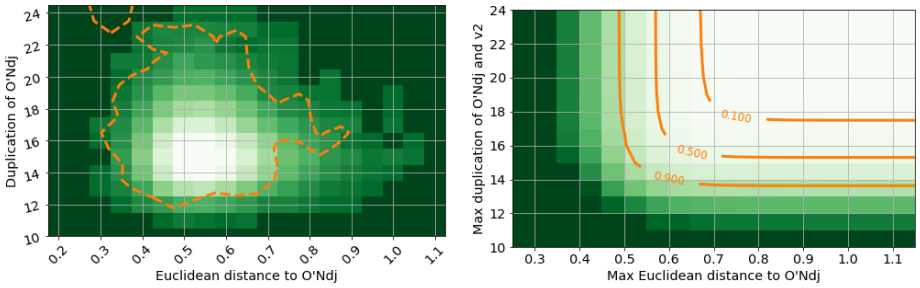

Fig. 4: Left: At Stage 3, the distribution of the number of tunes generated by *folk-rnn* (v2) possessing a given duplication of O'Ndj and Euclidean distance to O'Ndj, with the contour showing the smoothed region in which we find the tunes in O'Ndj from a leave-one-out test. Right: The rejection rate at stage 3 for tunes generated by *folk-rnn* (v2) for given distance and duplication thresholds.

it to D, G, or A with respective probabilities 113/248, 130/248, 5/248 (prior probabilities of modes for tunes in O'Ndj in those major keys). If the mode is dorian, then the critic randomly transposes it to A dorian, Edorian or Ddorian with respective probabilities 12/16, 2/16, and 2/16. If the mode is mixolydian, the critic randomly transposes it to A mixolydin or D mixolydian with respective probabilities 10/16 and 6/16. And finally, if the mode is minor, the critic randomly transposes it to B minor, E minor or A minor with respective probabilities 28/50, 20/50, and 2/50. The critic then determines whether the pitches are within the acceptable range, and if too low transposes up by an octave, or if too high transposes down by an octave. If the pitch range is still unacceptable, the critic repeats this procedure up to 200 attempts, and if the pitch range is still unacceptable the critic rejects the tune.

3 Application

We now apply this critic to a collection of 100,001 tunes generated by *folk-rnn* (v2) seeded with the 6/8 meter token and sampled using beam search on pairs of tokens.[4] In the first stage, the critic rejects 35,611 tunes having a measure token sequence that is not in O'Ndj; then it rejects 1,970 due to an inability to partition into an integer number of 8-measure segments, then 99 due to having a note duration longer than the dotted crotchet, and finally 3,375 based on a prevalence of the single jig pattern. This leaves 58,946 tunes passing into the second stage. At this point, if the critic uses the distance threshold 0.85 it rejects only 145 tunes. In the third stage, it rejects 118 tunes due to having in common more than 20 identical quavers in 24 in O'Ndj, and 692 more tunes for the same reason but in v2. Finally, of the remaining 57,991 tunes, the critic rejects 30 due to problems in finding a suitable music range. Out of the 100,001 tunes generated

[4] Beam search samples more than one token in a step. The search tree we use has 20 branches sprouting leaves pruned to those with a probability exceeding 0.01.

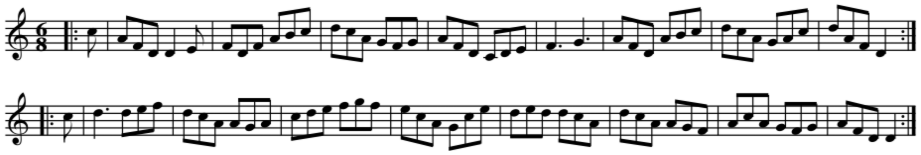

Fig. 5: Tune 2936 passes stage 1 of the critic, and sits near the mode of O'Ndj with Euclidean distance 0.549 and quaver interval duplication 17.

by *folk-rnn* (v2) then, 57,961 pass through all four stages of the artificial critic. If instead the critic selects tunes passing the first stage, but having a quaver interval duplication of at most 20, then 58,105 jigs result.[5]

Figure 4(left) shows the joint distributions of duplication and Euclidean distance for the *folk-rnn* tunes reaching the third stage, compared with the region in which most of O'Ndj is found using the leave-one-out-test. The modes of the two distributions match well: (0.55,17) for O'Ndj and (0.55,15) for *folk-rnn* (v2). Figure 4(right) shows the rejection rate of this stage for any choice of distance and duplication thresholds. For instance, the critic rejects 90% of the *folk-rnn* tunes when the maximum Euclidean distance is about 0.5, and the maximum permitted duplication of O'Ndj is larger than 16. Figure 5 shows a tune generated by *folk-rnn* (v2) that passes the first stage of the critic, and sits near the mode of the distribution of O'Ndj in Fig. 4(left).

4 Discussion

Our artificial critic aims to reject tunes that are not similar to the 365 double jigs in O'Neill's "1001" (O'Ndj). This is done with up to four stages of increasing specificity comparing the characteristics of the tune to those of O'Ndj. First, the critic examines the metric structure of a tune; then it looks at the intervalic periodicity of eight-measure segments; then it looks for excessive duplication of intervalic content from training material; and finally, the critic attempts to transpose the tune into an acceptable mode without violating pitch range constraints. Each stage is informed by and tailored to expert knowledge of the music style, an analysis of O'Ndj, the evaluation criteria of The Ai Music Generation Challenge 2020 (B. L. T. Sturm & Maruri-Aguilar, 2021), and testing on O'Ndj using a leave-one-out design. A version of this critic was used to create our submission to the 2020 Challenge (B. L. T. Sturm & Maruri-Aguilar, 2021),[6] but in the preparation of this manuscript, we discovered a variety of problems with the original implementation. For instance, it was not able to filter based on patterns more common to single jigs; and the original intervalic descriptors were implemented in such a way that unisons and repeated intervals were indistinguishable.

Other than reducing computational expense, there are no reasons why the specific steps are in the order described. The all-or-nothing thresholds could of

[5] This collection is here: `https://bit.ly/3vg6624`.

[6] This collection is here: `https://github.com/boblsturm/aimusic2020/blob/master/tunes_folkrnnv2wcritic.pdf`. Compare with that in footnote 5.

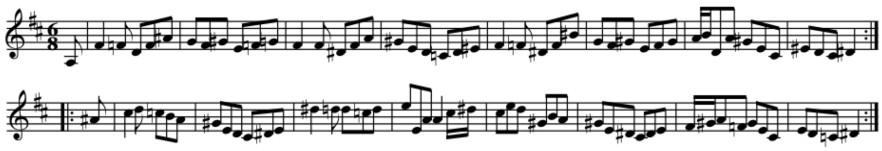

Fig. 6: This tune passes all four stages of the critic with a melodic self-similarity 0.516 and quaver interval duplication 13.

course be made softer, and the variety of characteristics could be considered together, perhaps combined into a score representing overall fitness. The setting of the thresholds in each stage is done using a leave-one-out test with O'Ndj, but it should also consider a collection of tunes that the critic should reject in its various stages. Then the parameters and stages should be tuned to admit all of O'Ndj while at the same time rejecting all of the unsuitable tunes. For instance, Fig. 6 shows a "nefarious" tune we created that passes the first stage, and has a melodic self-similarity acceptably close to O'Ndj. It is not a random collection of pitches, and has structure within each part, but it is far from the melodies in O'Ndj and should be rejected.

There are several ways in which to improve our critic. An improved critic will measure the similarity of a candidate melody to O'Ndj in a more complete way, and will consider the relationships between the parts of a tune. Tunes in O'Ndj have parts that relate in many ways through repetition and variation. Sometimes *folk-rnn* (v2) can generate tunes with such relationships between parts, but more often we see it move on to new ideas, leaving its melodic ideas half-baked. A variety of pattern-based approaches are applicable toward this end, e.g., Juhász (2006); Conklin and Anagnostopoulou (2011); Boot, Volk, and de Haas (2016); Janssen (2018); Yin, Reuben, Stepney, and Collins (2021). One might also include *folk-rnn* (v2) itself, to compute the likelihood of each segment of a given tune. Statistical approaches could also be used, e.g., Ens and Pasquier (2018); Yang and Lerch (2018). The latter could be used to compare the subsets of tunes created by the critic.

Finally, our work here is relatable to reinforcement learning (Jaques, Gu, Turner, & Eck, 2016), where our critic can provide a reward to an agent generating an entire tune. The critic can also be likened to a discriminative network in a generative adversarial network (Dong, Hsiao, Yang, & Yang, 2018), which is trying to determine whether a given observation is real or synthetic. It also moves in the direction of data augmentation (McFee, Humphrey, & Bello, 2015), where the tunes selected by the critic can be used to train new music generation models. In our present case, there is no training of the generative model or of the critic, but this can be a future direction of work once the critic is improved as outlined above. Nonetheless, a version of this critic will be applied to create a submission to *The Ai Music Generation Challenge 2021*,[7] where the style to be modelled is the Swedish slängpolksa.

[7] `https://github.com/boblsturm/aimusicgenerationchallenge2021`

References

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

Breathnach, B. (1971). *Folk music and dances of Ireland: A comprehensive study examining the basic elements of Irish folk music and dance traditions.* Ossian.

Conklin, D., & Anagnostopoulou, C. (2011). Comparative pattern analysis of Cretan folk songs. *J. New Music Research*.

Cuthbert, M., & Ariza, C. (2010). music21: A toolkit for computer-aided musicology and symbolic music data. In *Proc. int. symp. music info. retrieval* (p. 637-641).

Dong, H.-W., Hsiao, W.-Y., Yang, L.-C., & Yang, Y.-H. (2018). MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proc. aaai conf. ai.*

Ens, J., & Pasquier, P. (2018, June). A cross-domain analytic evaluation methodology for style imitation. In *Proc. int. conf. computational creativity.* Salamanca, Spain.

Janssen, B. (2018). *Retained or lost in transmission? analyzing and predicting stability in Dutch folk songs* (Unpublished doctoral dissertation). University of Amsterdam.

Jaques, N., Gu, S., Turner, R. E., & Eck, D. (2016). Generating music by fine-tuning recurrent neural networks with reinforcement learning. In *Deep reinforcement learning workshop, nips.*

Juhász, Z. (2006, June). A systematic comparison of different European folk music traditions using self-organizing maps. *Journal of New Music Research*, *35*(2), 95–112.

McFee, B., Humphrey, E. J., & Bello, J. P. (2015). A software framework for musical data augmentation. In *Proc. ismir.*

O'Neill, F. (1907). *The Dance Music of Ireland: O'Neill's 1001.* Chicago.

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., Santos, J. F., Ben-Tal, O., & Korshunova, I. (2016). Music transcription modelling and composition using deep learning. In *Proc. conf. computer simulation of musical creativity.* Huddersfield, UK.

Sturm, B. L. T., & Maruri-Aguilar, H. (2021). The Ai Music Generation Challenge 2020: Double jigs in the style of O'Neill's "1001". *Applied Sciences (submitted).*

Walshaw, C. (2021, Jan). *The abc standard.* Retrieved March 11 2021, from `http://abcnotation.com/wiki/abc:standard`

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. In *Proc. evomusart.*