C. Goddard, M. Barthet, and G. A. Wiggins,
"Assessing Musical Similarity for Computational Musical Creativity,"
J. Audio Eng. Soc., vol. 66, no. 4, pp. 267–276, (2018 April.).
DOI: https://doi.org/10.17743/jaes.2018.0012

Assessing Musical Similarity for Computational Musical Creativity

CALLUM GODDARD[1], MATHIEU BARTHET[1], AND GERAINT A. WIGGINS[2, 1]

[1] *Queen Mary University of London, London, UK*
[2] *Vrije Universiteit Brussel, Brussels, Belgium*

Computationally creative systems require semantic information when reflecting or self reasoning on their output. In this paper we outline the design of a computationally creative musical performance system aimed at producing virtuosic interpretations of musical pieces and provide an overview of its implementation. The case-based reasoning part of the system relies on a measure of musical similarity based on the FANTASTIC and SynPy toolkits that provide melodic and syncopated rhythmic features, respectively. We conducted a listening test based on pair-wise comparison to assess to what extent the machine-based similarity models match human perception. We found the machine-based models to differ significantly to human responses due to differences in participants' responses. The best performing model relied on features from the FANTASTIC toolkit obtaining a rank match rate with human response of 63%, while features from the SynPy toolkit only obtained a ranking match rate of 46%. While more work is needed on a stronger model of similarity, we do not believe these results prevent FANTASTIC features being used as a measure of similarity within creative systems.

1 INTRODUCTION

Automation of musically creative tasks, as the field of Musical Metacreation (MuMe) [1] seeks to investigate, generally requires elements of semantic information related to the specific task being automated. Such information is rational, meaningful information related to both the task and its context. The work presented here is specific to the creative musical task of musical performance by computer systems and the creativity and creative behaviors that these systems may display.

From a Computational Creativity perspective, a system displaying creative behavior must be capable of reflection. This is the ability for an agent (or in the context of this paper, a computational system) to evaluate or reason about its creative output and in light of this evaluation adapt or alter its behavior. This capability to reflect is crucial to creative systems [2, 3] and semantic information can be used to aid in the evaluation and reasoning that guides the system's reflection process. However, because it is more common for previous work developing musical performance systems, such as Computer Systems for Expressive Musical Performance (CSEMP) to not employ a full reflection loop or self-reasoning, we proposed in [4] to use the Creative Systems Framework (CSF) by Wiggins [5, 6] as a design tool to frame and describe both new or even existing CSEMPs as creative systems (should their authors wish to turn them into creative systems).

We relied on the CSF to design a new computationally creative music performance system that uses case-based reasoning to produce virtuosic musical performances with a physical model of a bass guitar (selected due to author expertise and experience with the instrument, and to allow for nuanced control of the performance rendering) [4]. Here, we refine the system implementation and focus on the measure of musical similarity used within our case-based reasoning system. We can see potential interest in this measure of musical similarity within semantic audio applications, particularly in the areas of online music education. For example, to train recommendation systems that can suggest new pieces of music for someone to learn (see, e.g., [7]) that are musically similar to what they already know but that might require more advanced playing ability, or vice versa, such as pieces that are musically dissimilar but require the same or similar playing ability as those that are currently playable by an individual. The musical similarity information could also be used within the vast on-line transcription resources that are available to both guitar and bass guitar players to aid navigation and again recommendation.

2 CASE-BASED REASONING WITHIN MUSICAL PERFORMANCE SYSTEMS

2.1 Case-Based Reasoning

All performing musicians use their own previous experiences, knowledge, and ability to develop a musical

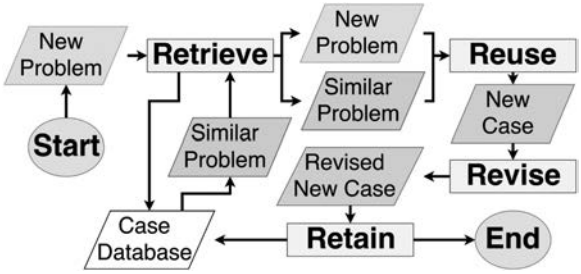

Fig. 1. The basic outline of a general case-based reasoning system.

performance. This is analogous to case-based reasoning (CBR), which is an approach that uses solutions to previous problems to solve new ones. These solutions (which initially need to be collected or created) are made available to the CBR system by being stored in a case database.

There are four steps in case-based reasoning—retrieval, reuse, revise, and retain [8]. Starting with a new problem, first a solution to a similar problem is retrieved. This solution is then reused as a solution to the new problem. Because the two problems may not be an exact match, the solution needs to be checked, which is done in the revise step. If the solution is not found to solve the problem in a satisfactory way it is then modified so that it does. Once the solution to the new problem is finalized it is retained as a new case within the case database so that it may be used to solve future problems. The basic outline of a CBR system is shown in Fig. 1.

If the problems that the system solves have many different solutions, reflection can be added to the revise step of a CBR system. For this an additional method of evaluating the derived solution, beyond checking the problem was solved, is needed. This will include additional heuristic information related to the problem. There is also the additional requirement for mechanisms to be in place that can further modify or change the derived solution in ways that will improve its evaluation.

2.2 Case-Based Reasoning in Musical Performance Systems

As defined in [9], musical interpretation in music is "the act of performance with the implication that in this act the performer's judgment and personality necessarily have their share." When applying case-based reasoning (CBR) as an approach to generating musical performances, a performance for a (normally) new, previously unseen musical piece is produced by considering the previous performances of similar musical pieces. CBR has been used to great effect in CSEMPs, such as SaxEx [10, 11] and DISTALL [12–14].

SaxEx produces expressive saxophone performances of jazz standards. Performances are produced using Spectral Modeling Synthesis to manipulate un-expressive saxophone recordings into expressive ones. The similarity measure used is based upon Narmour's implication realization model and Lerdahl and Jackendoff's generative theory of tonal music (GTTM) [10, 11]. DISTALL learns and applies expressive rules for piano performance using first order logic and linked clauses to represent its musical piece [12–14]. Similarity is measured by the distances between the solutions to the maximal flow minimal weight problem [12–14]. Tempo and dynamic curves are transfered from similar musical pieces to produce expressive piano performances. CBR has also been used in MuzaCazUza [15] to generate melodies by matching suitable melodic excerpts determined by a formula based upon Schöenberg's chart of regions.

3 VIRTUOSITY AND MUSICAL PERFORMANCE

The related work highlighted in Sec. 2.2 has a focus on expression. Our interest moves beyond expressive musical performances to that of performances that display virtuosity.

3.1 Defining Virtuosity in Musical Performance

Virtuosity is an ill defined term that relates to a musical performance and has both positive and negative connotations. Virtuosity demands both the highest levels of musicianship and the highest levels of technical proficiency. Performances recognized as demonstrating virtuosity exceed the normal expectations and standards for the performance. For virtuosity to be considered, the performance requires a performer who is capable of reflection and self-reasoning, the conveying of meaningful expression or symbolic significance, and an appreciative audience [16]. It is the audience members and listeners themselves who ultimately decide if a performance is virtuosic or not, and they make this judgment based upon their understanding of the domain the music and performance is in, the performer, as well as their own individual expertise, knowledge, and sensibilities. This judgment is summarized by Howard [16, p. 47] as "a judgment of merit over results achieved by the combination of exceptional musicianship and technical proficiency" and act like a seal of approval given by the critical community. Judgments can, and likely do, differ between receivers and anyone, including naive audience members, can potentially make a judgment on the performance. However experts, academics, and music critics are likely to possess a better understanding of each of the factors and thus, their judgments can be given more significance.

3.2 Musical Performance

We formalize the production of a musical performance as the result of a process in which a set of musical instrument techniques, $\{t_1, t_2, \ldots.\}$, are applied to a sequence of musical notes, $\langle n_1, n_2, \ldots\rangle$, by the player of the instrument. We represent the actions of the *Player* of the musical instrument as a function that applies a set of techniques to a sequence of notes. This is summarized by Eq. (1).

$$Performance = Player(\{t_1, t_2, ...\}, \langle n_1, n_2, ...\rangle) \qquad (1)$$

The selection of techniques and how they are applied to each note is to be left implicit within the *Player* function as it can be achieved through many different methods. For example using rules such as the KTH rules [17] or more

Table 1. Creative System Framework Symbols

Symbol	Definition
$\mathcal{U}$	A *Universe* of all possible concepts (artifacts) that can be both partial and complete, real or abstract, and also includes the notion of an empty concept ($\top$).
$\mathcal{C}$	Conceptual Spaces that are non-strict subsets of $\mathcal{U}$ which include c and $\top$.
c_x	Concepts, where $\forall c_1, c_2 \in \mathcal{U}.c_1 \neq c_2$
$\top$	An empty concept.
$\mathcal{R}$	Set of rules that constrain a single $\mathcal{C}$ from $\mathcal{U}$.
$\mathcal{T}$	A set of rules for traversing a $\mathcal{U}$, this includes search heuristics.
$\mathcal{E}$	A set of evaluation rules to evaluate or assign value to any concept in $\mathcal{U}$.
$\mathcal{L}$	A language, which contains an alphabet that is used to express concepts (c_x), and the rule sets: $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{E}$. Where $\mathcal{R} \in \mathcal{L}$, $\mathcal{T} \in \mathcal{L}$, $\mathcal{E} \in \mathcal{L}$. $\mathcal{L}$ is required to be sufficiently expressive to allow for metalevel modification of $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{E}$.
[[.]]	A function generator that maps a subset of $\mathcal{L}$ to a function that associates concepts in $\mathcal{U}$ with real numbers [0,1].
$\langle\langle ., ., . \rangle\rangle$	A further function generator that maps three subsets of $\mathcal{L}$ to a function that generates a new sequence of concepts, from an existing one.

Please refer to Goddard et al. [4] for further explanation of the CSF symbols and the components they refer to.

sophisticated machine learning methods, e.g., [18], as well as using case-based reasoning as outlined by the work in Sec. 2.2.

4 THE CREATIVE SYSTEMS FRAMEWORK

We take Computational Creativity to be: "The philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative" [19, p. 21].

Creative behavior can be differentiated into exploratory creativity and transformational creativity. The Creative Systems Framework (CSF) [5, 6] is a formalized abstract representation of exploratory creative systems, based on the philosophy of Boden [20]. Its purpose is to help to analyze, describe, and design creative systems.

The CSF is based upon the following septuple:

$$\langle \mathcal{U}, \mathcal{L}, [\![.]\!], \langle\langle ., ., . \rangle\rangle, \mathcal{R}, \mathcal{T}, \mathcal{E} \rangle \tag{2}$$

whose symbols are as shown in Table 1.

Eq. (3) is a formalization of Boden's notions of conceptual space by Wiggins and Forth [21] that allows for a conceptual space for a given $\mathcal{R}$ to be created. This will be a conceptual space containing all artifacts that are defined by the ruleset $\mathcal{R}$.

$$\{c \mid c \in \mathcal{U} \wedge [\![\mathcal{R}]\!](c) \geq 0.5\} \tag{3}$$

Concepts are deemed part of a conceptual space if the results of applying functions generated by the function generator [[.]] to $\mathcal{R}$, when compared to $\mathcal{U}$, is greater than a real valued comparator. In Eq. (3) this is a value of 0.5.

Exploratory creativity is achieved through the exploration of a conceptual space. Within the CSF, the ruleset $\mathcal{T}$ determines how the space is traversed. Traversal of the conceptual space is summarized by Eq. (4) where $\mathcal{T}$ is interpreted by the function $\langle\langle ., ., . \rangle\rangle$ that acts upon a sequence of known concepts/artifacts, c_{in}, to produce a sequence of new concepts, c_{out}. $\mathcal{R}$ and $\mathcal{E}$ are included in the interpretation function to allow for reasoning over the type and value of artifacts that are being traversed by $\mathcal{T}$. However, they are not a requirement of $\langle\langle ., ., . \rangle\rangle$. By removing $\mathcal{R}$ from the interpretation it is possible to generate artifacts not bound by the rules of $\mathcal{R}$, and removing $\mathcal{E}$ allows for the generation of artifacts that is not guided by any evaluation.

$$c_{out} = \langle\langle \mathcal{R}, \mathcal{T}, \mathcal{E} \rangle\rangle(c_{in}) \tag{4}$$

The value of an artifact/concept is defined by ruleset $\mathcal{E}$ and determined from a set of functions generated by interpreting $\mathcal{E}$ with [[]]. We take value to be "a relation between an artifact, its creator and its observers and the context in which creation and observation takes place" [22, p.2]. For further explanation of the CSF and its components, please see [4].

5 THE SYSTEM

Previously in [4] we produced a formalized system design in terms of the CSF for a performance system that uses case-based reasoning to produce virtuosic interpretations of a piece of music. We also briefly discussed one potential implementation. We provide a summary of the formal system design here, before expanding upon a refined system implementation that we are currently building.

5.1 Interpreting a Musical Piece

Referring back to Eq. (1), where we formalized a performance to be the application of a set of performance techniques to a sequence of notes (the musical piece), and in line with the definition in [9], we consider the interpretation of a piece as being the result of a decision making process related to what instrument or musical performance techniques should be applied to each note in a musical piece. We consider the application of expressive, contextual, situational, and historic intentions and conventions to fall under the category performance techniques.

We call the process of assigning a performance technique to a musical note, *adorning* the note. Notes may be adorned with multiple different techniques, assuming they do not pose a contradiction in the way they should be performed. All adornments (performance techniques) have an explicit, singular fixed interpretation.

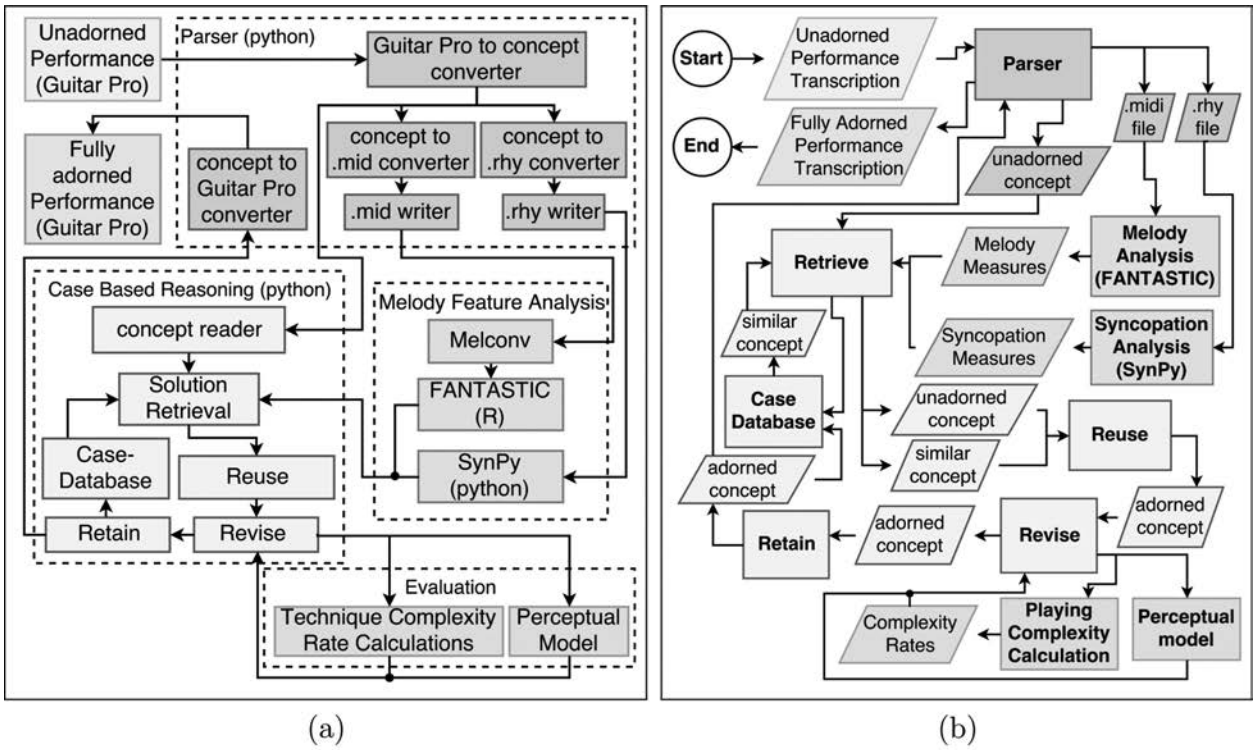

Fig. 2. (a) Code Block Diagram; Modules are indicated with dashed lines. (b) Data and Process Flow Diagram

We draw a distinction between the interpretation of a sequence of musical notes, and the interpretation of the adornments, with the latter not being considered within our system design. Instead the design decision was taken to have all adornments used within our system be capable of describing expressive performance intentions and have precise definitions of how any expressive intention is to be performed. This is so that our system design does not exclude any expressive or other such interpretations that are traditionally made through a realization of a performer's intention and personal interpretations of how adornments are performed.

This design choice allows for a decoupling between the interpretation of a musical piece and its technical execution. Having this distinction allows for any rendering or synthesis method to be used to produce a performance of the interpretation, assuming a suitable mapping or encoding is available from adornments to rendering/synthesis parameters.

5.2 Formal Description of the System

We produced a formal definition of our system using the CSF framework, outlined in Backus-Naur Form [4]. The specification of the language $\mathcal{L}$, the functions that are generated by $[[.]]$ and $\langle\langle ., ., .\rangle\rangle$, as well as an outline of the syntax for $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{E}$ are all presented in [4].

The $\mathcal{U}$ in which our system operates is a universe of all musical pieces. We restricted the operation of our system to a subset of $\mathcal{U}$ that contains musical note sequences of length greater than zero and that are not adorned with contradictory playing techniques. These restrictions are formally outlined by $\mathcal{R}$, and this subset of musical pieces forms the conceptual space $\mathcal{C}$ that the system operates in. $\mathcal{T}$ formalizes the functions needed for CBR and $\mathcal{E}$ the evaluation rules for determining virtuosity.

5.3 Implementation

Figs. 2(a) and 2(b) show a Code Block diagram and data and process flow diagrams of the current implementation of the system, as outlined in Sec. 5.2. This implementation takes Guitar Pro Files, which are a specialized digital notation program for electric guitar and bass notation (see: http://www.guitar-pro.com) as input. The musical information contained in the file is converted into $\langle concept \rangle$ representation. The musical notes within the $\langle concept \rangle$ are then adorned to produce a virtuosic interpretation before being converted back into a readable Guitar Pro File. Guitar Pro can then be used to render a performance of the interpretation. PyGuitarPro (http://pyguitarpro.readthedocs.io) is being used to read, edit, and convert Guitar Pro files into the $\langle concept \rangle$ representation, which is defined by $\mathcal{R}$. This $\langle concept \rangle$ can then be processed by the CBR module, which is the implementation of $\mathcal{T}$, and $\langle\langle ., ., .\rangle\rangle$.

First a $\langle concept \rangle$ that has the most similar melodic and rhythmic features is retrieved from the case database. The adornments from this $\langle concept \rangle$ are then compared and applied to a sequence of notes within the input $\langle concept \rangle$, forming a new interpretation of the input piece. Once the adornments have been applied, they are checked to ensure that the $\langle concept \rangle$ is performable according to $\mathcal{R}$. The ruleset $\mathcal{E}$ is to be implemented within the evaluation module and used to allow for the system to reflect and reason upon

the interpretations being produced during the revise stage of the CBR. This is to use a, as yet to be determined, playing complexity rate calculation and a perceptual model of bass guitar performance to evaluate the interpretation. At this stage, the interpretation can be converted back into a Guitar Pro file ready for performance and its ⟨*concept*⟩ stored in the case database for future use.

We have chosen to break down musical pieces into single bar segments and apply the CBR process to each bar individually. Once all bars of a musical piece have been processed they are then re-combined to form a complete interpretation of the musical piece. A continuity check between bars will be performed to ensure that $\mathcal{R}$ is conformed to, and an evaluation of the whole piece carried out, in addition to the individual evaluation of each bar contained within the the piece.

6 MUSICAL SIMILARITY

Much, if not all, evaluation and reasoning performed within and in relation to our system is dependent on perceptual information. Fundamental to the production of an interpretation of a musical piece is the functionality to identify similar pieces of music. We have chosen to use computable melodic, and rhythmic features as a basis for assessing musical similarity between ⟨*concepts*⟩. For this we have chosen to use melodic features from FANTASTIC [23] and syncopation features from the SynPy [24] toolkits.

6.1 FANTASTIC

FANTASTIC stands for Feature ANalysis Technology Assessing STatistics (In a Corpus) [23]. It is a program written in R (see http://www.r-project.org) that analyzes symbolic representations of monophonic melodies by computing features that can be used to characterize a melody or melodic phrase with a set of numerical or categorical values. These values represent different aspects of musical structure, making use of concepts from descriptive statistics, music theory, and music cognition [23]. FANTASTIC also has the option to compute corpus-level features with respect to a corpus of melodies, however we currently are not using this functionality of the toolkit.

FANTASTIC provides its own similarity function, which can compute the similarity between two or more melodies, based upon one or more computed features. When using only numeric features, the Euclidean distance, shown in Eq. (5), where d is the distance between two multi-dimensional points, p and q, is used to compute the similarity between melodies. Euclidean distance has been proposed as suitable similarity measure by Gärdenfors [25].

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + ... + (p_n - q_n)^2} \tag{5}$$

FANTASTIC's similarity function also applies a so-called z - *standardization* [23] to the feature values before the Euclidean distance is calculated. This is the subtraction of the feature mean (μ) and division by feature variance (σ). This standardization ensures an equal weighting is applied to all features when calculating the similarity.

To be able to use Euclidean distance as a measure of similarity, only the numeric features from FANTASTIC could be used. This excluded the use of the two contour and melodic mode categorical features. In testing with our dataset there were also issues computing features dependent on polynomial contour calculations so these features were also excluded from use. The remaining features relating to absolute pitch or notes, pitch intervals, note durations, global lengths, melodic step, and interpolation contours were all selected to be used. This gave a total of 26 features being utilized from the FANTASTIC toolkit.

FANTASTIC accepts Music-CSV (MCSV) files [26] as input. These MSCV files contain the symbolic representations of monophonic melodies. We are using the MELCONV software by Klaus Frieler to convert monophonic MIDI files to the MCSV format. Thus the presence of the MELCONV code block within Fig. 2.

6.2 SynPy

FANTASTIC does not handle rests within music; all note duration times are represented as inter-onset intervals (IOIs) and the features computed are only related to the IOI duration values. As we are primarily working with bass lines, the melodic content can be quite static and instead rhythmic variation are required to differentiate different pieces. We have selected to use features computed from SynPy, a python toolkit for syncopation modeling [24], as a way to include additional rhythmic features within our measure of similarity.

In SynPy seven different models for syncopation are implemented. Each model computes a numerical measure that relates to the syncopated-ness of the rhythm that is being analyzed. We combine the mean syncopation measures for each model with the melodic features of FANTASTIC and use both to compute the musical similarity between ⟨*concepts*⟩ in our case database.

SynPy can compute syncopation measures from MIDI or from its own rhythm (.RHY) file format. In testing, the .RHY files were more stable when being analyzed and thus we have implemented a ⟨*concept*⟩ to .RHY file converter.

7 MUSICAL SIMILARITY STUDY

Musical similarity is a very subjective judgment. To ensure that musically suitable ⟨*concepts*⟩ are being retrieved, we conducted a study to compare the values of musical similarity computed using FANTASTIC and SynPy features, with judgments of musical similarity by musicians and bass guitar players.

7.1 Study Design

Our study design is based on the method outlined by Allan et al. [27] where participants are presented with a triadic comparison of audio tracks and asked to specify which two audio tracks are most musically similar. Allan et al. [27] advocate presenting full permutations for every triadic comparison to account for presentation order bias. They propose using a Balanced Complete Block

Partitioning (BCBP) setup to allow for full permutations to be presented, while controlling the combinatorial explosion issues that occur when testing more than a couple of audio tracks. This approach involves partitioning every possible permutation of stimuli into smaller groups and presenting each of these groups to a different subject.

We chose to follow the Balanced Complete Block Partitioning (BCBP) setup for five tracks partitioned between six participants [27]. This setup splits every permutation combination into six groups (blocks) of 10. While no permutation is repeated over all these permutation blocks there is repetition of triads both between and within participants. This allows for within subject and between subject response consistency to be assessed.

We randomly selected five bars of monophonic music from an initial bass guitar transcription dataset containing 30 bars of notated music within which span pop, rock, funk, and jazz genres. These were then labeled A, B, C, D, E and separate files made for each bar's notated transcription. The tempo of the selected bars were: 125, 104, 92, 121, and 95 beats per minute for tracks A, B, C, D and E respectively. To allow a fair comparison between the computed values of similarity that only account for musical content, all playing techniques, dynamics, and expressive notations were removed from the notated transcriptions files. The audio for each of the separate bars was then rendered from their notated transcriptions files using Guitar Pro 6. This resulted in five separate audio files with durations varying between two and three seconds, with a mean of 2.2 and variance of 0.2 seconds. No other segmentation or processing preceded the final formulation of the audio content.

7.2 Demographics

There were 12 participants (11 males, 1 female). Eleven identified themselves as being musicians; five participants played bass guitar and three were music teachers. Ages ranged from 22 to 54 years old, with an average age of 34. Musical instrument playing experience, with the exception of the non-musician ranged from 10 to 50 years with an average playing experience being 24 years.

7.3 Results

To allow for 12 participants, each of the 6 permutation blocks were presented twice to different participants. An analysis of the complete set of participants' responses was conducted followed by a partitioned analysis where participants were clustered based upon the similarity of their responses.

7.3.1 Complete Set of Participants' Analysis

The complete set of all participants' responses were matched to three computed measures of similarity using FANTASTIC's similarity function. The first measure used an aggregated similarity measure that utilized all the FANTASTIC and SynPy features outlined in Sec. 6 (totaling 33 features); the second measure used only the FANTASTIC features (totaling 26 features); and the third used only the SynPy features (totaling 7 features) to compute similarity. Table 2 shows the voted similarity of tracks, for each combination along with the computed measures of similarity using features from FANTASTIC + SynPy, FANTASTIC only, and SynPy only. The number of matches between the computed similarity and reported similarity by participants is shown in Table 3 as well as the rank matches and rank match rate. The highest ranking match was 63%, achieved when only using FANTASTIC features; the lowest was 46% when using SynPy features; both combined yielded a match of 60%.

A Kruskal-Wallis rank sum test was performed between all participants' answers for every presented permutation, and each of the three computed similarity values. We wished to see if the responses from the participants and the computer similarity values could have been derived from the same populations (null hypothesis). A p-value greater than 0.05 would indicate both sets come from the same population (null hypothesis can't be rejected), whereas a p-value smaller than 0.05 would indicate the similarity judgments come from significantly different populations. The results are shown in Table 4, where all p-values are smaller than 0.05 indicating a significant difference in similarity rating between participants and all computer similarity values.

The consistency of each participant's individual rating and the consistency between each pair of participants were calculated using Fleiss' κ (Kappa). Individual consistency was calculated by treating each rating instance from a participant as a separate rater for each repeated combination of tracks. The κ values are shown in Table 5. The κ values calculated between pairs of participants presented with the same block partition are shown in Table 6. Fleiss' κ (Kappa) was calculated for all responses to each of the 10 possible combinations of tracks. The results are shown in Table 7. κ values less than 0 indicate poor agreement, values between 0.21 to 0.40 indicate fair agreement, values between 0.61 to 0.80 indicate strong agreement. p values less than 0.05 indicate that the agreement between participants' responses are not due to random chance.

7.3.2 Partitioned Sets of Participant Analysis

Participants were clustered based upon a dissimilarity matrix formed from the Fleiss' κ between each pair of participants' responses, not just pairs that were presented with the same block. It should be noted the p-value for every κ between participant pairs, with the exception of when there was only one response that could be compared, was less than 0.05. This indicated that the agreement in responses between each pair of participants where two or more responses could be compared is unlikely to be due to chance. However, as some pairs of participants could only be compared based upon one response, we chose to cluster based upon k-medoids, due to the method being more robust to outliers.

The R function `pamk`, Partitioning Around Medoids (PAM) [28] with estimation of number of clusters, was used to partition participants. The optimal number of clusters was identified by the function to be four.

Table 2. Full Results

Track Combination	Track Pairs	Times Voted Most Similar	Computed Similarity: FANTASTIC + SynPy	Computed Similarity: FANTASTIC	Computed Similarity: SynPy
ABC	AB	0	0.7632	0.7095	0.3985
	AC	10	0.7879	0.8440	0.3944
	BC	2	0.8346	0.7997	0.8249
ABD	AB	3	0.7632	0.7888	0.3985
	AD	4	0.7761	0.7417	0.6416
	BD	5	0.7366	0.7095	0.5106
ABE	AB	2	0.7632	0.7888	0.3985
	AE	4	0.7812	0.8058	0.4300
	BE	6	0.8102	0.7706	0.8048
ACD	AC	10	0.7879	0.8440	0.3944
	AD	0	0.7761	0.7417	0.6416
	CD	2	0.7731	0.7647	0.5004
ACE	AC	5	0.7879	0.8440	0.3944
	AE	6	0.7812	0.8058	0.4300
	CE	1	0.8337	0.7967	0.8613
ADE	AD	1	0.7761	0.7417	0.6416
	AE	10	0.7812	0.8058	0.4300
	DE	1	0.7212	0.6876	0.5154
BCD	BC	3	0.8346	0.7997	0.8249
	BD	6	0.7366	0.7095	0.5106
	CD	3	0.7731	0.7647	0.5004
BCE	BC	3	0.8346	0.7997	0.8249
	BE	9	0.8102	0.7706	0.8048
	CE	0	0.8337	0.7967	0.8613
BDE	BD	4	0.7366	0.7095	0.5106
	BE	8	0.8102	0.7706	0.8048
	DE	0	0.7212	0.6876	0.5154
CDE	CD	2	0.7731	0.7647	0.5004
	CE	9	0.8337	0.7967	0.8613
	DE	1	0.7212	0.6876	0.5154

Computed similarity values were calculated using Euclidean distance between all features present in each similarity measure. Values range between 0–1. Times voted most similar contains the votes for most similar pairs per combination. Total votes for each of the combination is 12.

Table 3. Percentage match between computed musical similarity and participants' responses

Feature Set	Matches	Match Rate (%)	Rank Match	Rank Match Rate (%)
FANTASTIC + SynPy	57 / 120	47.500 %	18 / 30	60.000 %
FANTASTIC	65 / 120	54.167 %	19 / 30	63.333 %
SynPy	35 / 120	29.167 %	14 / 30	46.000 %

Rank match comparisons counts the number of times the relationships between the most (first) similarly voted pair of tracks with the second, the second with the third, and first with the third match the relationships of the computed values of similarity.

Table 4. Kruskal-Wallis rank sum test between computed similarity values and human responses

Feature Set	Kruskal-Wallis: χ^2	Kruskal-Wallis: p
FANTASTIC + SynPy	29.012	2.307e-05
FANTASTIC	23.993	0.0002178
SynPy	37.869	3.013e-08

Table 5. Fleiss' κ for each participant's responses.

Participant: Pair	Participant: id	Repeated Permutation (κ): 1	Repeated Permutation (κ): 2	Repeated Permutation (κ): 3
1	1a	−8.33e-17	−8.33e-17	NaN
	1b	NaN	−8.33e-17	0
2	2a	NaN	−8.33e-17	NaN
	2b	0	−8.33e-17	NaN
3	3a	−8.33e-17	NaN	NaN
	3b	−8.33e-17	−8.33e-17	NaN
4	4a	−8.33e-17	NaN	NaN
	4b	−8.33e-17	NaN	NaN
5	5a	−8.33e-17	−8.33e-17	NaN
	5b	NaN	NaN	NaN
6	6a	0	NaN	NaN
	6b	−8.33e-17	NaN	NaN

A NaN value indicated complete consistency in an individual's response. A value of –8.33e-17 indicates that one response differed from the other two. A value of 0 indicates all responses differ.

There were participants who identified as bass players in clusters two (2), three (2), and four (1). The one participant who identified as a non-musician was placed in cluster one. One participant in each of clusters one, two,

Table 6. Fleiss's κ rating for pairs of participants presented with the same block partition.

Pair	κ	Z	p
1	0.219	1.1	0.271
2	0.375	2.2	0.0278
3	0.286	1.53	0.126
4	0.615	3.12	0.00168
5	0.318	1.62	0.105
6	0.241	1.33	0.184

Table 7. Fleiss' κ rating for combined participant responses for each possible track combination.

Combination	κ	Z	p
ABD	−0.0909	−1.03	0.302
ACD	−0.0909	−0.739	0.46
ADE	−0.0909	−0.96	0.337
BCD	−0.0909	−1.02	0.306
CDE	−0.0909	−0.942	0.346
ABC	−0.0909	−0.739	0.46
ACE	−0.0909	−0.896	0.37
ABE	−0.0909	−0.903	0.366
BCE	−0.0909	−0.739	0.439
BDE	−0.0909	−0.739	0.46

and three identified as a music teacher. The mean musical experience of participants in each cluster was: 26.67, 21.5, 21.33, and 15 years for clusters one, two, three, and four respectively. The number of matches and rank match rates between the three computed measures of similarity were calculated for each cluster. The results are shown in Table 8.

7.4 Discussion

The significant difference in the Kruskall-Wallis rank sum test results indicate that the computed values of similarity come from a different population to that of the participants' responses. Variations can be seen between participants' responses within each combination of tracks, whereas the computed values of similarity do not have any variation. This may explain why there were not high match rates between any of the computed measures of similarity and the full set of participant responses.

Given the subjective nature of musical similarity this variation in responses could be due to differences between participants' own subjective interpretations of musical similarity. The formation of four clusters from the partitioning process indicates that different participants may share similar interpretations of musical similarity. Clusters that had a higher level of musical experience (cluster one) and a higher number of bass players (cluster two) did see improvements in match rate, while the cluster with the least musically experienced participants (cluster four) saw a worsening in match rates compared to the full set of participants' responses. This would appear to indicate that the computed measures of similarity are a closer match to more expert listeners (those with more musical experience and specialization with the instrument). However, it should be noted that the cluster sizes are small, and some κ measures between participants could only be calculated based upon one response from each; thus, there is still a possibility of random chance effecting this partitioning. A comparison of our results to a larger study with more expert listeners would be interesting and could help confirm what is being indicated here.

From all our ranking match rate analysis it appears overall SynPy features are not as effective an indicator of musical similarity as FANTASTIC. However this difference in feature performance suggests that participants were relying on different features more heavily than was accounted for in the computed measures. This follows Allan et al.'s [27] work that states that when using aggregated measures of similarity, the contributing features require weighting adjustments to be made to better match people's own understanding of musical similarity. The relative weighting of features used within our computed similarity measures could be adjusted to better align with the clustered participants' responses using Eq. (6). This could be done by adjusting weight factors w for the features (D) so that the error ε is reduced and the similarity calculation matches the study responses. However, such optimizations would be best done with results from a larger study.

$$D_{study} = \epsilon + \sum_{k=1}^{n} (w_k . D_k) \tag{6}$$

More work is needed on a stronger similarity model, but meanwhile we do not believe these results prevent using FANTASTIC features as a measure of similarity within a creative system. There may be an effect on how the output from a creative system is valued due to the significant disparity between observers and the system's "understanding" of measures of musical similarity. When considering our own system that is to produce virtuosic performances, this mismatch might produce serendipitous performances and also alter people's expectations of how the music can be performed. However, these measures of similarity are likely a poor metric to use within a recommender system without any weighting optimization.

8 SUMMARY

We have described a creative system that utilizes semantic audio information to produce virtuosic interpretations of musical pieces. The tools and framework used to design such a system have been summarized and a refined implementation outlined. A method for matching musically similar bars of music has been described and a study conducted to evaluate how perceptually valid the computed similarity values are. While the study was small in scope, it highlights the subjective nature of musical similarity and the careful considerations required when using computable features as an indication of measures of similarity within creative systems.

Table 8. Percentage match rates between computed musical similarity and clustered participant responses

		FANTASTIC + SynPy		FANTASTIC		SynPy	
Cluster	Participant id	Match (%)	Rank Match (%)	Match (%)	Rank Match (%)	Match (%)	Rank Match (%)
1	1a, 1b, 2a	50.000 %	42.857 %	53.333 %	66.667 %	50.000 %	23.810 %
2	2b, 4a	60.000 %	70.833 %	65.000 %	70.833 %	60.000 %	37.500 %
3	3a, 3b, 4b	63.333 %	80.952 %	56.667 %	47.619 %	40.000 %	52.381 %
4	5a, 5b, 6a, 6b	20.000 %	38.095 %	47.500 %	47.619 %	20.000 %	33.333 %

9 ACKNOWLEDGMENT

This work was funded by the Engineering and Physical Sciences Research Council (EPSRC) through the Media and Arts Technology Programme, a Research Councils UK Centre for Doctoral Training (EP/G03723X/1).

REFERENCES

[1] P. Pasquier, A. Eigenfeldt, O. Bown, and S. Dubnov, "An Introduction to Musical Metacreation," *Comput. Entertain.*, vol. 14, no. 2, pp. 2:1–2:14 (2017 Jan.), https://doi.org/10.1145/2930672.

[2] A. Bundy, "What Is the Difference between Real Creativity and Mere Novelty?" *Behavioural and Brain Sciences*, vol. 17, no. 3, pp. 533–534 (1994), https://doi.org/10.1017/S0140525X0003572X.

[3] K. Agres, J. Forth, and G. A. Wiggins, "Evaluation of Musical Creativity and Musical Metacreation Systems," *Comput. Entertain.*, vol. 14, no. 3, pp. 3:1–3:33 (2016 Dec.), https://doi.org/10.1145/2967506.

[4] C. Goddard, M. Barthet, and G. A. Wiggins, "Designing Computationally Creative Music Performance Systems," *Proceedings of AM'17* (2017 August 23–26), https://doi.org/10.1145/3123514.3123541.

[5] G. A. Wiggins, "A Preliminary Framework for Description, Analysis and Comparison of Creative Systems," *Knowledge-Based Systems*, vol. 19, pp. 449–458 (2006), https://doi.org/10.1016/j.knosys.2006.04.009.

[6] G. A. Wiggins, "Searching for Computational Creativity," *New Generation Computing*, vol. 24, no. 3, pp. 209–222 (2006), https://doi.org/10.1007/BF03037332.

[7] M. Barthet, A. Anglade, G. Fazekas, S. Kolozali, and R. Macrae, "Music Recommendation for Music Learning: Hotttabs, a Multimedia Guitar Tutor," *Proc. of the 2nd Worskshop on Music Recommendation and Discovery (WOMRAD) collocated with ACM-RecSys* (2011).

[8] A. Aamodt and E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *Artificial Intelligence Comm.*, vol. 7, pp. 39–59 (1994).

[9] P. A. Scholes, *The Oxford Companion to Music, 2nd ed.* (Oxford University Press, Oxford, 1960).

[10] J. L. Arcos, R. L. Mantras, and X. Serra, "SaxEx: A Case-Based Reasoning System for Generating Expressive Musical Performances," *J. New Musical Res.*, vol. 27, no. 3, pp. 194–210 (1998), https://doi.org/10.1080/09298219808570746.

[11] R. L. de Mantras and J. L. Arcos, "AI and Music: From Composition to Expressive Performance," *AI Magazine*, vol. 23, pp. 43–57 (2002), https://doi.org/10.1609/aimag.v23i3.1656.

[12] A. Tobudic and G. Widmer, "Relational IBL in Music with a New Structural Similarity Measure," *Proceedings of the International Conference on Inductive Logic Programming*, pp. 365–382 (2003), https://doi.org/10.1007/978-3-540-39917-9_24.

[13] A. Tobudic and G. Widmer, "Case-Based Relational Learning of Expressive Phrasing in Classical Music," in P. Funk, P. González Calero (Eds.), *Advances in Case-Based Reasoning*, pp. 419–433 (Springer Berlin Heidelberg, 2004), https://doi.org/10.1007/978-3-540-28631-8_31.

[14] A. Tobudic and G. Widmer, "Relational IBL in Classical Music," *Machine Learning*, vol. 64, no. 1, pp. 5–24 (2006 Sep.), https://doi.org/10.1007/s10994-006-8260-4.

[15] P. Ribeiro, F. Pereira, M. Ferrand, and A. Cardoso, "Case-Based Melody Generation with MuzaCazUza," *AISB'01* (2001).

[16] V. A. Howard, "Virtuosity as a Performance Concept: A Philosophical Analysis," *Philosophy of Music Educ. Rev.*, vol. 5, no. 1, pp. pp. 42–54 (1997).

[17] A. Friberg, R. Bresin, and J. Sundberg, "Overview of the KTH Rule System for Musical Performance," *Advances in Cognitive Psych.*, vol. 2, no. 2-3, pp. 145–161 (2006), https://doi.org/10.2478/v10053-008-0052-x.

[18] G. Widmer and W. Goebl, "Computational Models of Expressive Music Performance: The State of the Art," *J. New Music Res.*, vol. 33, no. 3, pp. 203–216 (2004), https://doi.org/10.1080/0929821042000317804.

[19] S. Colton and G. A. Wiggins, "Computational Creativity: The Final Frontier?" *Proceedings of the 20th European Conference on Artificial Intelligence*, ECAI'12, pp. 21–26 (2012), https://doi.org/10.3233/978-1-61499-098-7-21.

[20] M. A. Boden, *The Creative Mind: Myths and Mechanisms, 2nd ed.* (Routledge, Londo, 2004), https://doi.org/10.4324/9780203508527.

[21] G. A. Wiggins and J. Forth, *Computational Creativity and Live Algorithms* (Oxford University Press, Oxford, UK, 2016).

[22] G. A. Wiggins, P. Tyack, C. Scarf, and M. Rohrmeier, "The Evolutionary Roots of Creativity: Mechanisms and Motivations," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 370, no. 1664 (2015), https://doi.org/10.1098/rstb.2014.0099.

[23] D. Müllensiefen, "FANTASTIC: Feature ANalysis Technology Accessing STatistics (In a Corpus): Tech. report v1.5," Tech. rep., Goldsmith's University London (2009).

[24] C. Song, M. Pearce, and C. Harte, "SynPy: A Python Toolkit for Syncopation Modelling," presented at the *12th Sound and Music Computing Conference* (2015).

[25] P. Grdenfors, "Conceptual Spaces as a Framework for Knowledge Representation," *Behavioral and Brain Sciences*, vol. 27, no. 3, pp. 403–403 (2004), https://doi.org/10.1017/S0140525X04280098.

[26] K. Frieler, "Melody - CSV File Format (MCSV)," Tech. report (2005).

[27] H. Allan, D. Müllensiefen, and G. Wiggins, "Methodological Considerations In Studies of Musical Similarity," *ISMIR* (2007).

[28] A. P. Reynolds, G. Richards, B. de la Iglesias, and V. J. Rayward-Smith, "Clustering Rules: A Comparison of Partitioning and Hierarchical Clustering Algorithms," *J. Math. Models. Algor.*, vol. 5, no. 4, pp. 475–504 (2006 Dec.), https://doi.org/10.1007/s10852-005-9022-1.

THE AUTHORS

Callum Goddard

Mathieu Barthet

Geraint A. Wiggins

Callum Goddard studied electronic engineering and music technology systems at the University of York, UK. From 2011–2013 he worked as research assistant developing new musical interfaces at Aalto University School of Arts, Design and Architecture in the Sound and Physical Interaction (SOPI) Research Group. He is currently a Ph.D. candidate of the Media and Arts Technology Ph.D. program at Queen Mary University of London.

•

Mathieu Barthet Ph.D. is a lecturer in digital media and technical director of the Media and Arts Technology Studios at Queen Mary University of London. His research lies at the intersections of music information retrieval, perception, and human computer interaction. He currently investigates new interfaces for musical expression, listening, and participatory performance and also conducts fundamental research on musical timbre and emotions. He is Principal Investigator of the EU Internet of Musical Things project (iomut.eu) and Co-Investigator of the EU Audio Commons (audiocommons.org) project at QMUL.

•

Geraint A. Wiggins studied mathematics and computer science at Corpus Christi College, Cambridge, and holds Ph.D.s in artificial intelligence and musical composition from the University of Edinburgh. He has worked in musical AI research since 1987. From 2000–2004 he chaired the Society for the Study of Artificial Intelligence and the Simulation of Behaviour. From 2007–2014 he was the founding chair of the international Association for Computational Creativity. He is now Professor of AI at the Vrije Universiteit Brussel and Professor of Computational Creativity at Queen Mary University of London.