Finding Measures of Creativity from Music with Bisociations

Juho Kallio

Seminar report

UNIVERSITY OF HELSINKI
Department of Computer Science

Helsinki, June 22, 2016

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty

Faculty of Science

Laitos — Institution — Department

Department of Computer Science

Tekijä — Författare — Author

Juho Kallio

Työn nimi — Arbetets titel — Title

Finding Measures of Creativity from Music with Bisociations

Oppiaine — Läroämne — Subject

Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Seminar report	June 22, 2016	16

Tiivistelmä — Referat — Abstract

In this paper Koestler's idea of bisociation is used and extended to explain self references in music as a form of creativity. Using bisociations as the core framework and known methods for extracting rare connection from a graph structure, a method is presented for finding measures from music that correlate with creativity. The results are preliminary but in line with the claims.

Avainsanat — Nyckelord — Keywords

bisociation, music, artificial intelligence

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1 Introduction

In 1964 Arthur Koestler wrote a book about creativity [7]. The key particle in his theory for any kind of creativity was the *bisociation*, a concept for an event where two normally unconnected things are combined in a surprising way. The ideas have been used with some success in computer science[1], but not necessarily on the principled home field of computational creativity.

We first look at Koestler's bisociations as a framework to model creativity. Especially we look at abstract classical music, as this represents some challenges for the framework. We introduce a new type of bisociation to model self references, as according to Douglas Hofstadter [4] this seems to be key concept at least in J. S. Bach canon music.

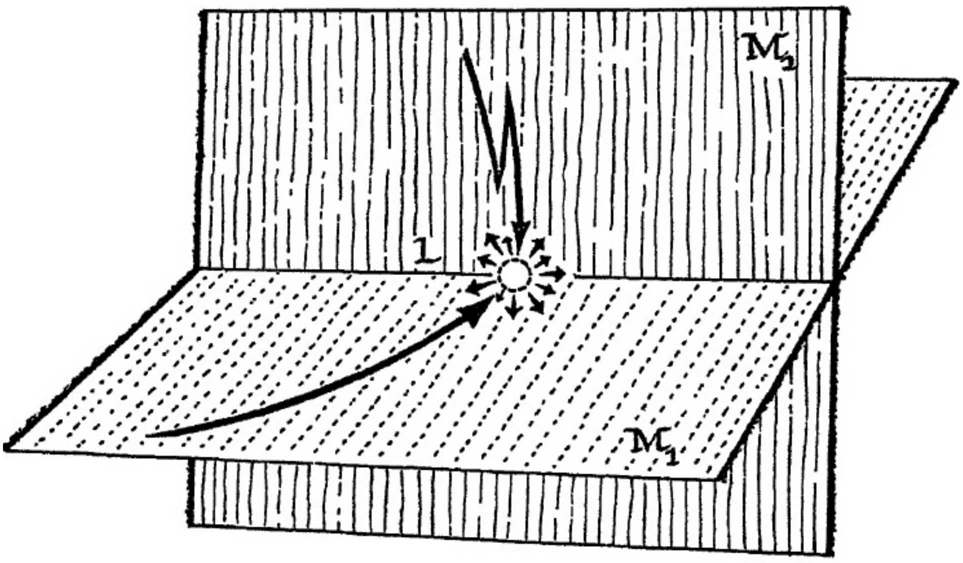
For computational methods, we look at The BISON project. It was created for finding more powerful insights from data using bisociation motivated techniques. The target was to find an unsupervised search for ideas where the type of the results is not strictly defined beforehand [3]. These methods work on graph based data [2], which is flexible restriction enough to make them usable for our purposes.

The computational methods are used to find different kinds of measures of creativity in single melody music. The creativity measured here is of certain type which we claim to be present at least in art that doesn't represent external concept. These are tested with music from Bach, popular lullaby Twinkle, Twinkle, Little Star and computer created music. The hypothesis is, that Bach should contain more of this creativity than the rest of the material. The results are in line with the hypothesis, but are too limited to draw conclusions.

2 Bisociation

A bisociation is a connection of two normally unconnected things. According to Koestler's ideas [7], it is the essence of creativity. In jokes we find a surprising way to connect two normally unconnected things, and in art we get fascinated of the unusual connection that is visible. In Koestler's language, we have different planes of contexts, which we refer here as *idea* *planes*, and when we manage to connect ideas from two planes, we have a bisociation. The nature of the connections defines the nature of creativity. With humor we have a class, see figure 1, and with scientific discovery the planes are perfectly aligned.

Figure 1: Koestler's bisociation planes with humor like creative clash. M_1 and M_2 are two separate idea planes that happen to collide here with an event of joke. Source: [7].



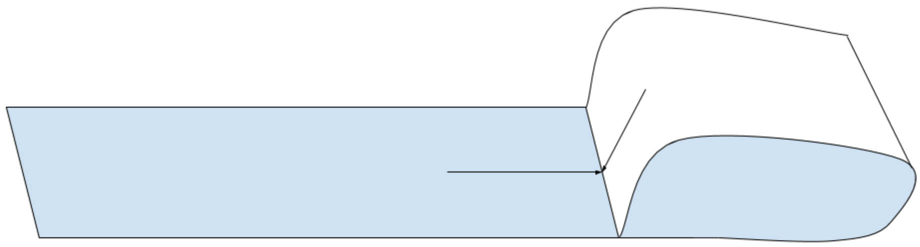
With art as the form of creativity, according to Koestler there is no clash but the planes are in juxtaposition, see figure 2. When we see a painting, we see something else than just the material, color and shapes. Even by just noticing the shapes, the painting has started representing for us something else than just the color pigments and canvas. I think that with music it is the same. If we heard only the sounds as what they really are, the sounds would not cause any particular feelings or thoughts in ourselves. It's easy to connect well made painting to it's subject, and not hard to see how a music piece with singing can connect ourselves to different time and place through the words, but how come something as simple as one tune piano melody be seen as connection to another idea plane? Some melodies are sad and some happy, but there is also music that I don't consider to represent any external concept. To me, that is a huge challenge for Koesler's theory. Even more, it is an opportunity.

Figure 2: Idea planes in juxtaposition, which represents the Koestler's idea of art.

Canon is a musical form where a song is created only by variating the original melody, it is introduced for example by Hofstadter [4]. There may be small changes in it, it can be speed up or slowed down, or it even can be reversed. This kind of technique definitely connects the music internally, but it is hard to see a clear connection to a separate idea plane. Still, according to Koestler's idea of creativity, there should be one. Douglas Hofstadter argues [4], that this kind of self referencing creates *strange loops*, which is a concept that he defines for paradox like events where after some progress we end at the same place where we left from. The basic idea is in essence, that we self referencing is a special event which is in itself already meaningful for humans, unlike simple sounds. For the purpose of our framework, inner references become the idea plane that the sequences of sounds are now representing.

I believe the concept of strange loop can also be seen through the Koestler's bisociation framework. Self reference happen with only single idea plane. If the reference is an obvious or common one, it is kind of just part of the structure of the idea plane. When it is a rare one, it creates a loop like structure in the idea plane, see figure 3.

Figure 3: The new kind of idea plane with a self referencing strange loop.



To give you a feeling of what lengths Hofstadter goes with his idea, I will simply mention that later he attempted to define consciousness in his later book I Am a Strange Loop. For the purpose of this paper, we now have the two ideas, bisociation and strange loop, which together have the power to explain abstract classical music as a form of creativity. In addition to canons, general composing techniques such as introducing a theme by repetition in beginning and getting back to it in more subtle ways later [11] don't seem too different from these ideas.

3 Discovery of Subgraph Bisociations

U. Nagel and co [10] describe a method for finding bisociations from graph structure. In their context, bisociation stands for a rare connection, that is, a clear and rather unique connection between two separate ideas. The original motivation of the work was to have a datamining method for interesting and potentially valuable connections in a data set such a scientific research papers. The research papers would be nodes in a graph, and clusters of nodes would form domains of research. The ideas to be connected would be the domains. Using the framework of Koestler's bisociations, I believe this can be viewed in two ways:

1. The domains are the idea planes and this graph allows finding direct connections between them. This is probably the original idea, and perfectly reasonable, but not particularly useful for us here.

2. The formed is not just a map but also a presentation of an idea plane of the academic research papers. From this point of view, the links are self references for the complete idea of *academic research papers*. Within clusters of papers, the references define connected domains but the more unique ones can in fact be strange loop like bisociations.

I believe the first model of viewing the method can always be seen as well through the second model, as it is basically just wrapping the first model's ideology into another abstraction. Changing the views the other way around may not be possible, as when we only have one bisociative idea plane, we can represent that with a graph with the idea of the second model, but it may be impossible or at least impractical to divide our only idea plane into smaller planes. For our purposes, we are interested precisely of the second model, finding bisociations from a single idea plane.

3.1 Finding Clusters

The graph is divided into clusters. A cluster δ is a set of vertices that represents something tightly connected, like a theme or just a sequence of notes in music. The optimal exact method for clustering is likely to be domain and graph specific, since in other cases it should be preferred to find clusters based on *activation similarity*, which means that their neighborhood is similar [13], and in other cases based on how highly the nodes are connected. For the purpose of finding self references within a single idea plane, the choise of clustering method could be viewed being part of the artists vision and individual taste.

A promising method of cluster selection based on activation similarity is presented in [10]. I ended up using for clustering another method from [14], mainly to save time as the source code was available and usable with minimal modifications. As we are seeking all kinds of self references within one idea plane, we can choose the clustering method freely. We could even ditch the graph structure for clustering, find features of the nodes and use some kind of machine learning, e.g. k-means [8].

3.2 Discovering Bisociations

First we need to define few concepts. A *bridging node* $\texttt{bn}(\delta_1, \delta_2)$ is a node the connects the two clusters δ_1 and δ_2. We can have multiple of these for any two clusters, and if we have none, the clusters are not connected. Also, a bridging node can be part of one of the clusters, which means that the clusters are direct neighbors.

A *bisociation candidate* is defined as a subgraph induced by two clusters and all their neighbor nodes, though in practice it may be more useful to think of it as two clusters and their set of bridging nodes. The bisociation candidates are valued based on the *b-score*,

$$\texttt{b-score}(\delta_1, \delta_2) = \frac{2}{\sum_{v \in \texttt{bn}(\delta_1, \delta_2)} d(v)} \cdot \frac{\texttt{min}(|\delta_1|, |\delta_2|)}{\texttt{max}(|\delta_1|, |\delta_2|)} \cdot (|\delta_1| + |\delta_2|), \qquad (1)$$

where the bridging node's degree $d(v), v \in \texttt{bn}(\delta_1, \delta_2)$ is the number of neighbors of the bridging node, and the size of a cluster $|\delta|$ is the number of nodes the cluster contains. The motivation for the equation 1 is three-fold: the first part $\frac{2}{\sum_{v \in \texttt{bn}(\delta_1, \delta_2)} d(v)}$ favors connections that are rare, as is essential for bisociation to be a bisociation. The second part $\frac{\texttt{min}(|\delta_1|, |\delta_2|)}{\texttt{max}(|\delta_1|, |\delta_2|)}$ aims to find subgraphs that are about the same size. The third part $(|\delta_1| + |\delta_2|)$ favors bigger subgraphs, as we value higher more general bisociations. The need for this becomes clearer when we realize that the bigger subgraphs are also likelier to be more connected to everything, which works against the first element.

The goal is to find the bisociation candidates in the graph and order them by their b-score. This allows us to investigate closer the best bisociations and, when used within single idea plane, have some measures for self references. The bisociation candidate set consists of every cluster pair. The minimal cluster contains one vertex of the original graph, and the same vertex can be used in only one cluster. The worst case when we have only one node clusters, we have to check $\binom{n}{2}$ pairs. Furthermore, we need to find the bisociation candidate's bridging nodes and their degree. The exact implementation of this depends a bit of the form the graph is modeled, but basically we check all the neighbors of cluster δ_1 and if they either are part of cluster δ_2 or its neighbors, they are also bridging nodes. This operation takes calculation count related to the bigger sum of the two clusters' nodes' degrees, $\mathcal{O}(\texttt{max}_\delta \in \{\delta_1, \delta_2\} \sum_{v \in \delta} d(v))$, as is shown in [10].

4 Music and Bisociations

We use here the Koestler's bisociation framework, and specifically the idea that art is about two idea planes in a juxtaposition. For some music this works well, as we can connect to it emotionally and observe the idea of e.g. sorrow indirectly from the music. Here we focus however to such music that doesn't seem to connect to anything external and which seem to contradict the framework.

I'm limiting the scope of investigated music to piano or similar music and only the melody part of it. I will use the works of J. S. Bach as examples of music of high creativity. I will make the assumption, that in the Bach samples the idea plane we are trying to connect to is created mainly from the self references in the music. One example of such self referencing is the canon form, where whole piece is constructed of different kinds of variations of the theme. I assume that similar self references are present in basically any kind of music and that the amount and quality of them should correlate with the level of this specific kind of creativity in the music. This should be motivated by the edge cases of complete repetition and randomness. The former has everything connected in many ways, but with any kind of idea of connection, it is really hard to make claim of rare connection. In the latter example then, randomness provides only rarely anything we could imagine as a theme or other similar concept, which could then be referenced elsewhere.

In addition to the reference music by Bach, I'm using music created by my program AiMusic [5]. It's program that automatically composes music with a genetic algorithm [9]. For such algorithm, the fitness function is one of the most interesting pieces. It defines, what is good or bad music. I've tried using there the algorithm presented here, a machine learning model that tries to learn what is good and the combination of the two.

4.1 Music as Graph

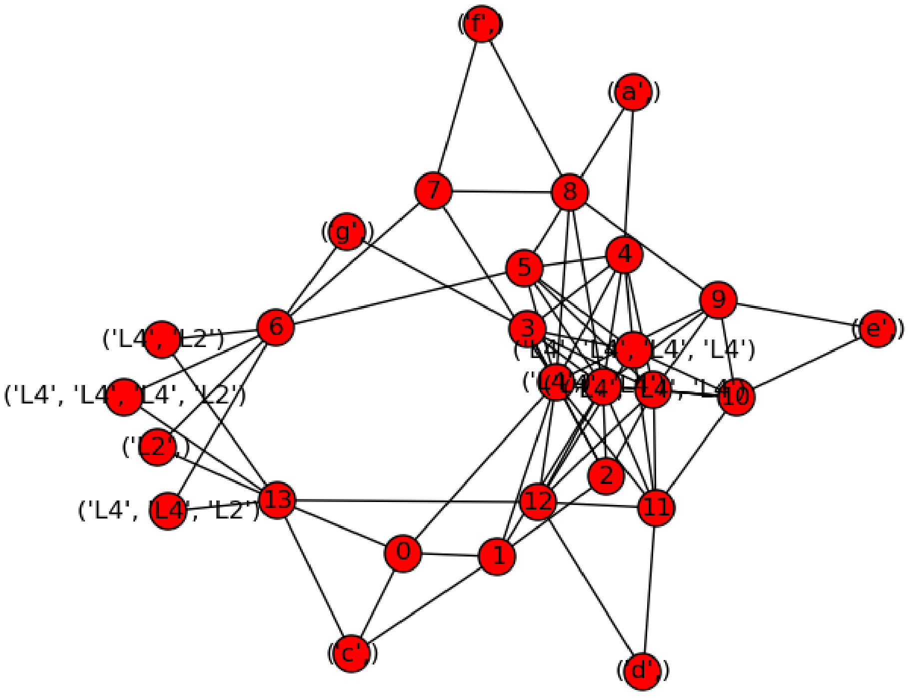
Music is not easily presentable as graph. There is research on ways to represent music [12], but especially since the time and scope of this paper is limited, I'm suggesting a more direct approach. We can make the assumption, that the abstract high quality classical music we focus on, contains bisociations, and also that the music generated by my simple program does not. This gives us data for a simple brute force approach, where we will try different kinds of graph structures and choose the one that gives us clearest difference with bisociations between music by Bach and music that people normally don't consider music.

A graph structure for music means here a model derived out of single music piece where we present musical events like notes as vertices and their connection as edges. It's a structure that can be stored as a two dimensional matrix.

The vertexes in the graph could be either placements of the notes, sequence of pitches or sequence of note lengths. Even more complex vertexes could be interesting, but here the scope is limited to the most essential.

The song Twinkle twinkle little star starts from quarter length c. We mark down the vertex "note index 0" to represent the first node, but won't yet add anything else, since we avoid adding nodes that connect to nothing. As the next note is another quarter note c, we now have a musical vertex that connects note indexes, and add the vertexes "note c", "length quarter" and "note index 1". We also add edges between all our vertexes except "length quarter" and "note c", as these are connected indirectly via the note indexes. We add to the memory the possible connection, the sequence "note c, note c", and add it later with related edges when encountered again. Continuing like this, we get a graph of Twinkle twinkle little star, see figure 4.

Figure 4: First line of Twinkle twinkle little star as a graph. The nodes with numbers only represent the notes, nodes with character sequences are pitch sequences and nodes with lengths, e.g. ('L4', 'L2'), are note length sequences.



The computer created music with bisociation fitness function outscored Bach with this simple model. This kind of model is very simple and uses very few forms of self reference, and thus the program will find some optimal solution and still not be more creative than Bach. To investigate the behaviour further, I added a new feature: pitch change sequences. A note sequence of c, d, e would return to pitch change sequence of 1, 1. This is also a highly dependent feature with the pitch sequences. Extending the graphs outside the single song, like to related music from the same genre, would be interesting as well, but it is not relevant for the core concept.

4.2 Using Subgraph Bisociation Discovery

The motivation with the music graph is to be able to make comparisons between the different pieces of music. In order to do this, we need to extract reasonable subgraphs and find the bisociation candidates between them. Here the subgraphs are limited to the indexes of the notes only. With a single piece of music this makes sense, since we are really trying to find connections between parts of the piece. A note in the middle of the piece might be connected to sequences (d, c, d, e), (c, d, e), (d, e), (e), (L4, L8) and (L8). Now this note would be highly connected with another note that connected to the pitch sequence (c, d, e), as this would indicate connections to (d, e) and (e) as well. If (c, d, e) and (d, e) are rare in the piece and the notes both belong to balanced and sizeable clusters, we would have a good b-score.

For comparing and analyzing the graphs, I used the sum of 10 best bisociation candidates as the main criteria. Choosing exactly 10 is completely arbitrary, but we need some fixed subset as using only the best doesn't tell enough of the whole picture and using all candidates might favor bigger graphs and also, only the best candidates should be considered real bisociations. It might be better to have some threshold and sum all the bisociations with higher score than that, but the challenge is that the exact threshold would depend on the graph structure. One idea could be to choose the threshold from the b-scores of the candidates, e.g. use the best b-score of the worst graph.

5 Results

I compared the found bisociation scores between Twinkle twinkle little star, J.S. Bach simple one tune conversions of the works BWV 772 inventio 1 and BWV 784 inventio 13 and music produced with AiMusic using first

Table 1: b-scores with full length pieces and only pitch and length sequences as features

Music	Length	Best	Sum of top 10	Sum of all
Twinkle twinkle little star	42	0.5	2.5	23.4
AiMusic No. 56, machine learning fitness	151	0.05	0.43	32.6
AiMusic No. 63, b-score fitness	124	4	20	101.8
BWV 772 inventio 1	256	2	12.54	245.12
BWV 784 inventio 13	330	2.67	19.33	331.22

machine learning based and then bisociation based fitness functions. As the music lengths differ and longer piece gives more space for more significant bisociations, I did two things. First I selected only the shortest piece length of notes from each music piece, and second I used the whole pieces. Both methods are unfair in their own ways, but hopefully the combination leaves room for some conclusions.

The first experiment was comparing the music with only the lengths and pitches of notes as the features for the constructed graph, see table 1. The music from Bach yields significantly higher scored bisociation candidates compared to the rest of the field, except the computer made music that was explicitly optimized for finding the perfect structure for b-score.

Changing the sample length mainly seem to impact the sum of all b-scores, see table 2. Note that the length can impact the b-scores both ways, as more music can make the existing connections more or less common, and create new connections, influencing also the subgraph selection.

With the added features, Twinkle twinkle little star suddenly has the best bisociations. My conclusion is, that the connection space becomes too

Table 2: b-scores with only 42 first notes of each piece, using pitch and length sequences as features

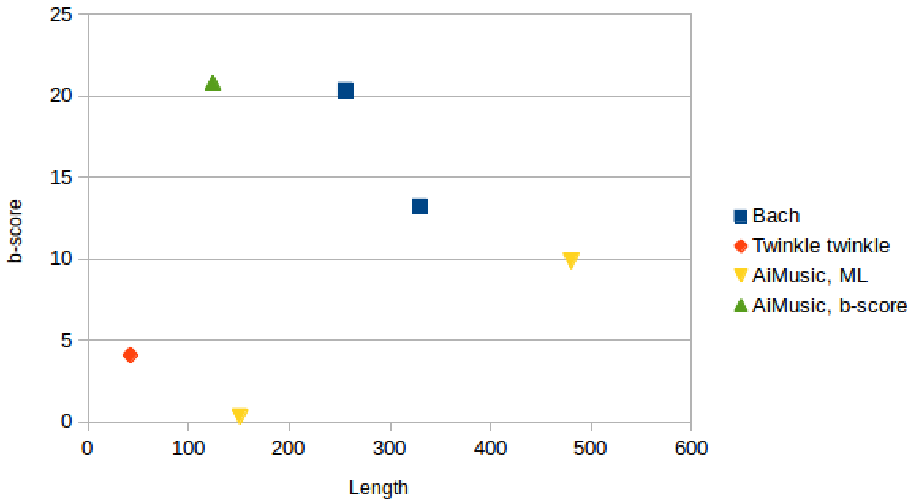
Music	Best	Sum of top 10	Sum of all
Twinkle twinkle little star	0.5	2.5	23.4
AiMusic No. 56, machine learning fitness	0.08	0.79	9.47
AiMusic No. 63, b-score fitness	4	12.33	22.28
BWV 772 inventio 1	2.67	13.17	32.45
BWV 784 inventio 13	3	20.33	45.41

Table 3: b-scores with full length pieces and pitch, pitch change and length sequences as features, one length sequences neglected

Music	Length	Best	Sum of top 10	Sum of all
Twinkle twinkle little star	42	0.80	4.10	28.05
AiMusic No. 56, machine learning fitness	151	0.05	0.33	21.80
AiMusic No. 63, b-score fitness	124	4	20.80	78.36
BWV 772 inventio 1	256	4.00	20.33	244.60
BWV 784 inventio 13	330	1.60	13.23	345.73
AiMusic No. 10-17 concatenated, machine learning fitness	480	1.14	9.87	554.0

crowded and all connections common. Twinkle twinkle is the shortest piece, which also supports this hypothesis. Furthermore, the bisociation scores seem to strongly relate to the length of the song. This makes sense, since the shorter the sample is, the less there will be connections connecting everything to everything.

Figure 5: Top 10 b-scores with length, pitch, and pitch change sequences as features, one length sequences being neglected



It is mostly the one length connections that make everything connected, especially as all the connections are equally valued. When all one length connections where removed from the graphs, Bach scored again well, see table 3 and figure 5. Here bisociation optimized computer created music was the best, and when machine learning based computer music pieces where concatenated together, the end result scored surprisingly well, only little below the other Bach piece. This might indicate the algorithm favoring longer pieces, as the concatenated computer created music performed poorly alone and got close to 0 b-score.

I also tried using b-scores as the only fitness function for music generating genetic algorithm. It was much slower than having e.g. random forest classifiers as fitness function but still usable. Furthermore, the results [6] were somewhat promising. Bisociation scoring seemed to really find connections inside the music, and at the same time avoid outright repetition. Obvious downside of using only this kind of fitness function is the absolute lack of harmony, and at times either too slow or fast rhythm.

6 Conclusions

We have represented self contained creativity in music in terms of self references, which we have argued to be in fact a new kind of bisociation. This representation has allowed us to use the subgraph bisociation discovery method to investigate self reference based creativity in music. The results are inconclusive due to the limited test sizes, and another limitation is that self referencing might take almost any kind of form. Here we explored only few possibilities. The methods are however promising and preliminary results seem to find higher scored bisociations from Bach than from the Christmas song Twinkle, Twinkle, Little Star or computer generated music.

The models used are simplistic, and it would be interesting to experiment for example with weighted graphs. This would change the algorithm somewhat but could be very useful, as not all the connections are equal in music. A simple approach could be to value the sequences so that the longer the sequence, the stronger the connection. One possible way to model this would be to have

$$\texttt{weight} = 1 - \frac{1}{\texttt{sequence length}},$$

which would remove the one length sequences, that seemed to be harmful. It is also an open question whether the number of certain kinds of connections should affect their weights. It could be debated, that if we only have half the amount of different length sequences compared to pitch sequences, maybe connections with length sequences should be weaker as they are common.

The cluster selection is another aspect that could be improved. The chosen approach of placements of the notes as the only allowed subgraph is not the only sensible one, and different algorithms for the selection could be used, as it is not clear, between which elements the connections in music should exactly be made. Especially with longer pieces, it might make sense to try to find larger and maybe more abstract clusters.

The method described here is tested only with music with one consecutive voice. With more general form of music, the algorithm and graph presentation is likely to need some adjustments. Ideas here should be applicable, but it is not obvious that there won't be any challenges. The method could also be useful in adjusting the micro timings of music. Midi music doesn't sounds the same as human played, because all the timings and volumes are generally equal. It would be quite natural to model these within a graph, and this would give a sensible target which the computer could attempt to optimize.

References

[1] Berthold, Michael R. (editor). Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, ISBN 978-3-642-31829-0.

[2] Berthold, Michael R.: *Towards bisociative knowledge discovery.* In Berthold, Michael R. (editor): *Bisociative Knowledge Discovery*, volume 7250 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, ISBN 978-3-642-31830-6.

[3] Faloutsos, Christos: *Foreword.* In Berthold, Michael R. (editor): *Bisociative Knowledge Discovery*, volume 7250 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, ISBN 978-3-642-31830-6.

[4] Hofstadter, Douglas R.: *Godel, Escher, Bach: An Eternal Golden Braid.* Basic Books, Inc., New York, NY, USA, 1979, ISBN 0465026850.

[5] Kallio, Juho: *Aimusic.* `https://github.com/juhokallio/AiMusic`.

[6] Kallio, Juho: *Pure bisociations song.* `https://soundcloud.com/user-508904177/pure-bisociations`.

[7] Koestler, Arthur: *The act of creation.* Hutchinson, 1964.

[8] MacQueen, J.: *Some methods for classification and analysis of multivariate observations.* In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press. `http://projecteuclid.org/euclid.bsmsp/1200512992`.

[9] Mitchell, Melanie: *An Introduction to Genetic Algorithms.* MIT Press, Cambridge, MA, USA, 1998, ISBN 0262631857.

[10] Nagel, Uwe, Thiel, Kilian, Kötter, Tobias, Piątek, Dawid, and Berthold, MichaelR.: *Towards discovery of subgraph bisociations.* In Berthold, Michael R. (editor): *Bisociative Knowledge Discovery*, volume 7250 of *Lecture Notes in Computer Science*, pages 263–284. Springer Berlin Heidelberg, 2012, ISBN 978-3-642-31830-6. `http://dx.doi.org/10.1007/978-3-642-31830-6_18`.

[11] Randel, Don Michael: *Harvard Dictionary of Music*, chapter exposition, page 302. Belknap Press of Harvard University Press, Cambridge, MA, 2003, ISBN 0-674–01163-5.

[12] Soriano, A., Paulovich, F., Nonato, L.G., and Oliveira, M.C.F.: *Visualization of music collections based on structural content similarity.* In *Graphics, Patterns and Images (SIBGRAPI), 2014 27th SIBGRAPI Conference on*, pages 25–32, Aug 2014.

[13] Thiel, Kilian and Berthold, MichaelR.: *Node similarities from spreading activation.* In Berthold, Michael R. (editor): *Bisociative Knowledge Discovery*, volume 7250 of *Lecture Notes in Computer Science*, pages 246–262. Springer Berlin Heidelberg, 2012, ISBN 978-3-642-31829-0. `http://dx.doi.org/10.1007/978-3-642-31830-6_17`.

[14] Weeks, Margaret R., Clair, Scott, Borgatti, Stephen P., Radda, Kim, and Schensul, Jean J.: *Social networks of drug users in High-Risk sites: Finding the connections.* AIDS and Behavior, 6(2):193–206, June 2002. `http://dx.doi.org/10.1023/A:1015457400897`.