ChordScanner: Browsing Chord Progressions based on Musical Typicality and Intra-Composer Consistency

Hiromi Nakamura   Tomoyasu Nakano   Satoru Fukayama   Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST)
{hiromi.nakamura, t.nakano, s.fukayama, m.goto}@aist.go.jp

ABSTRACT

We propose ChordScanner, a visual interface enabling users to find chord progressions that they typically use and that many composers also typically use. ChordScanner provides a time-oriented view of chord progressions and highlights two kinds of chord progressions: chord progression the user frequently used and chord progression typical of many composers. To develop this system, we proposed a method for calculating intra-composer consistency based on the user's pieces and musical typicality based on many composer's past pieces by using a probabilistic model. The system helps users notice their repeatedly used chords. It also presents chord progression rankings based on the consistency and typicality. We investigated the overlap between rankings obtained with both consistency and typically, by calculating Japanese consumer generated songs, and also interviewed a professional composer.

1. INTRODUCTION

Not only amateur composers but even professional composers are sometimes faced with the problem that they tend to use several chord progressions repeatedly. Chord progression distributions seem to follow Zipf's law in that a few progressions are used in many songs even though most others are rarely used [1]. Composers may consciously or unconsciously select which chord progressions they like or find easy to use in their compositions. Even if they notice them, they often do not know how to go beyond those chord progressions. It also difficult for them to determine whether or not they should try to go beyond them or instead maintain their style.

On the other hand, clients who want to order new pieces have a different problem: sometimes it is hard for them to convey their wishes correctly in musical terms. They have some image of the piece they want to order and may even know a large amount of music as listeners, but they do not always have musical knowledge. According to a professional composer who had experience with pieces ordered by musical producers, *'They have some images of the pieces they want, not only specific images like existing pieces but also broad outlines based on typicality, but they tend to describe them in vague terms. So sometimes it is hard for me to understand their wishes correctly.'* This is a problem that occurs not only with music producers but also with people who want background music for movies and companies that want corporate songs. They too may also find it hard to choose composers who may good for composing the pieces they want.

Chord progressions have higher reusability than melodies. Understanding chord progressions helps composers establish and maintain their style, and visualizing chord progressions and their typicality helps clients understand a composer's characteristics and a genre's characteristics, making it easier for them to communicate with composers.

In this paper, we propose ChordScanner, a visual interface that enhances user recognition of chord progressions (Fig. 1). It is based on the composers' past composed pieces and those of many composers. ChordScanner provides two benefits to the users. One is that they can comprehensively grasp where they tend to use characteristic chord progressions, and the other is that they can distinguish whether these chord progressions are often used by many composers or are characteristic of the user. This interface enhances the users' analysis and creative practice, it also helps them characterize their pieces and decide what chord progressions to use next. It also facilitates communication between composers and clients because it helps clients break their vague impressions down into chord progressions and share musical information with composers.

To provide this system, we define consistency and typicality. Consistency is a value based on one composer's past pieces, and it is high if the composer uses the same chord progressions repeatedly. It relates to style and is a characteristic feature of the composer. Typicality, on the other hand, is calculated on the basis of many composer's past pieces.

2. RELATED WORK

Many creativity support tools to assist composers in composing chord progressions have been proposed. These systems help composers find chords harmonizing with a given melody and enable the user to adjust chords according to 'mood' preference [2,3]. ChordRipple also helps novice composers use chord progressions that make their music both diverse and acceptable and to find chords that are appropriate but rarely used in a particular musical context [1]. ChordScanner does not suggest chords, as ChordRipple and MySong [3] do, but the user can use it to help decide whether to accept or reject suggested chords because our system helps analyze the user's style and the styles of many composers' past work.

A horizontal bar graph is useful for showing a lot of information in a small area [4]. Colorscore shows an overview of structures of classical music by using a horizontal

Copyright: © 2018 Hiromi Nakamura et al. This is an open-access article dis- tributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

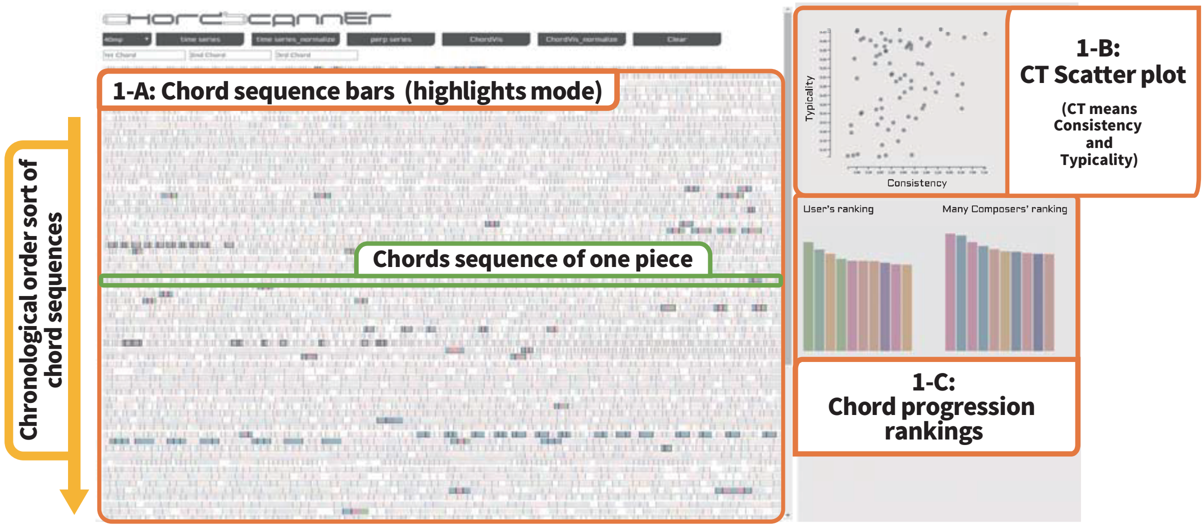


Figure 1. In this figure, 1-A: Horizontal chord sequence bars of each piece, 1-B: Scatter plot of whole-music consistency and typicality, 1-C: User's ranking and many composer's ranking based on generative probabilistic model.

bar graph so that novices can quickly understand them [5]. ColorScore visualizes each instrument's part as a color bar time sequence, and it compresses the views of these bars into one bar. ColorScore focuses on visualizing many instrument tracks, but our ChordScanner focuses on chord sequence. And because we want the user to get a comprehensive view of whole pieces, we treat one bar as one song and put it in order as a horizontal graph.

Analyzing and visualizing several elements of music, Uehara et al. proposed a visualization tool that presents two different graphs analyzing one dataset in different ways [6]. They visualized acoustic feature values as a tune scatterplot and visualized meta information (artists, chord progressions, etc.) as a meta information scatterplot. ChordScanner also has three kinds of views (Fig. 1-A, B, and C) that help the user understand chord progressions. In addition, the consistency and typicality of datasets can be visualized as a novel function.

3. CONSISTENCY AND TYPICALITY

In this section, we describe consistency and typicality. We also explain the method of calculating these values. ChordScanner calculates consistency and typicality using the same procedure but different datasets. Consistency is calculated uses pieces composed by only one composer (the user), and typicality is calculated using several composers' pieces (Fig. 2). The user can select the datasets used to calculate musical typicality if the user wants to compare genres, for example, or pieces composed at certain times.

3.1 Methods

For calculating consistency and typicality, we used Nakano et al.'s method [7,8], which can compute the typicality of a song as the sum of the probabilities of the songs that share the type of the given song. To train a probabilistic model that can compute the consistency and the typicality, a variable-order Pitman-Yor language model (VPYLM) [9,10] can be used to calculate the probability and avoid the zero-frequency problem by enabling a hierarchical smoothing. As in the previous method [7], given the type (i.e., the topic distribution) of each song, we compute the typicality of a song as compared to a set of other songs. When the above definition is applied to an *n*-gram (VPYLM) [10], we use a Bayesian topic n-gram model (Hierarchical Pitman-Yor topic model: HPYTM) [11] to calculate a topic distribution for each song.

This method also allows us to calculate the probability of the user's high-ranked chord progressions from many composer's datasets (Fig. 3). The ranking of chord progressions having high probability based on song sets by the users (i.e., consistency) is based on the counts of chord progressions in the datasets consisting of the users' songs, but the users can find out how often chord progressions they use frequently are also used by other composers (i.e., typicality).

3.2 Interface

ChordScanner has three views: chord sequences visualization for checking the chords of songs, a scatterplot of consistency/typicality for understanding the consistency and typicality of those songs, and generative probability ranking charts of users'/many composers' pieces for understanding chord progressions that are repeatedly used. In this part, we explain each function of the view.

3.2.1 Chord sequences visualization by horizontal bars

The area shown in Fig. 1-A visualizes chord sequence of pieces. This view illustrates each song's chord sequence as one horizontal color bar whose total width and each chords width are normalized. Chords are displayed in a color based on the root note. We mapped colors and chords as indicated in Fig. 4. By default, the horizontal color bars visualizing each piece's chords sequence are ordered chronologically, and ChordScanner allows sorting by the consistency and typicality of each song. It allows users to understand how their preference is changed following a creative phase and to find out which chord progressions are typically used in high- or low-consistency pieces.

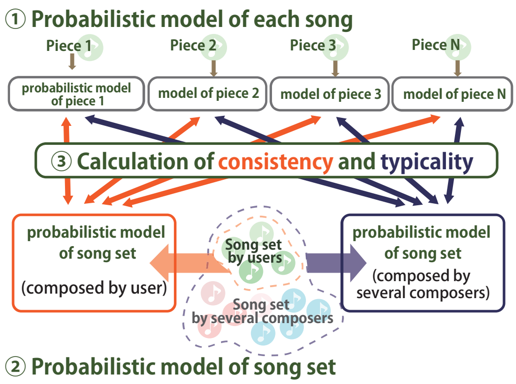
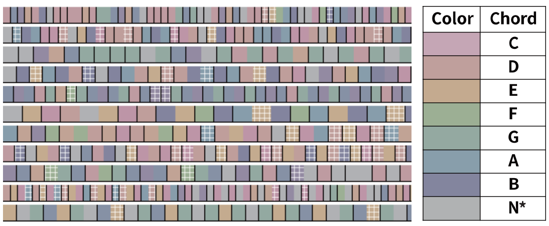


Figure 2. Method of calculating consistency and typicality based on different datasets.

The user's ranking (e.g,. 40mp's ranking)	Probability of the user's chord progressions by many composer's dataset (e.g,. 7 vocaloid composers' ranking)
Fm7 | A#m7 | D#m7 | G#	0.372
G/C | Cadd9 | Dsus4 | Em7	0.513
Cm9 | F6 | A#M7 | Gm7	0.402
E6 | Nadd9 | F# | G#m7	0.416
Nm | D | G6 | A	0.391
Gm7 | D#m | F | Gm7	0.388
Em9 | A6 | DM7 | Nm7	0.060

Figure 3. Chart of probability of the user's chord progressions in many composers' dataset.



*N = Non-detectable chords (by music structure analyzing system, Songle)

Figure 4. The enlarged image of horizontal chord sequence bars (Fig. 1-A.) and color scheme of those bars.

3.2.2 Scatterplot of consistency and typicality

Consistency and typicality of each song are plotted in the area shown in Fig. 1-B. This part allows users to control the number of songs visualized in the area shown in Fig. 1-A. Our system allows users to select songs they want to pick up by dragging the pointer to a selected area (Fig. 5).

3.2.3 Generative probability ranking chart of users'/many composers' pieces

The area shown in Fig. 1-C represents generative probability ranks calculated from datasets. ChordScanner outputs rankings based on two datasets, one consisting of the user's songs and the other consisting of many composers' songs. Users can see where these chord progressions are used in songs by selecting the chord. Chord progressions chosen by the user can seem to float to the surfaces because the transparency of other parts decreases (Fig. 6).

3.3 Scenarios

In this section, we show two examples of scenarios of our proposed system. First, we explain an example of a scenario by composers. Composers can set their pieces and some pieces that they want to compare with regard to typicality. ChordScanner visualizes chord sequence at the area shown in Fig. 1-A, where the user can get a bird's-eye view of a whole song's chord progressions. The composer can infer the overall tendency of their compositions from the colors and overlaid patterns, but they may not recognize which are typical chord progressions for the composer and many composers. So the composer can check the ranking of chord progressions in the area shown in Fig. 1-C. The composer can find out which chord progressions are frequently used in the composer's pieces and which are frequently used in the pieces of many composers.

ChordScanner allows the composer to find where the chord progressions are used in pieces by clicking the chord in the ranking. After the user selects chords, the chord sequences in the area shown in Fig. 1-A change their transparency: selected chord progressions become less transparent and the others become more transparent. The user also finds chords by typing these into input forms. This change helps users to figure out which songs use the selected chords and where those songs use them. The user also adjusts the number of songs visualized at the area shown in Fig, 1-A by using the area shown in Fig. 1-B, the scatterplot of consistency and typicality (Fig. 6).

In addition, our system allows the user to adjust the number of songs the user selected by dragging the area; for example, the user can refine the list for checking high-consistency and low-typicality pieces. This view helps the user finds some characteristic chord progressions while the user is conscious of the songs' consistency/typicality. For example, if the user highlights chord progressions that include many composers' chord progression rankings after the user chooses high-consistency and low-typicality pieces, the user can find out how often a typical chord progression is used in the user's high-consistency pieces.

ChordScanner can also be used to visualize probability values of user's ranking chords based on many composers' datasets, so a user can understand how often the chord which is included into user's ranking is used by many composers.

The main procedure for clients is the same as that for composers. Clients also utilize this interface for finding typical chords of past pieces by setting these songs as datasets of many composers. Of course, they can use this interface to understand a composer's chord progressions style when they are deciding with whom to place their orders. In addition, it may help communication among clients and composers when clients convey an idea because they have chord progressions that they want to explain to composers.

4. EXPERIMENT

We calculate the probability $p(x, n)$ (x: chord, n: order) to achieve "stochastic phrases" [9] of chord progressions by the composer and many composers. However, the dataset of the user may consist of not only the user's characteristic chord progressions but also musically typical chords.

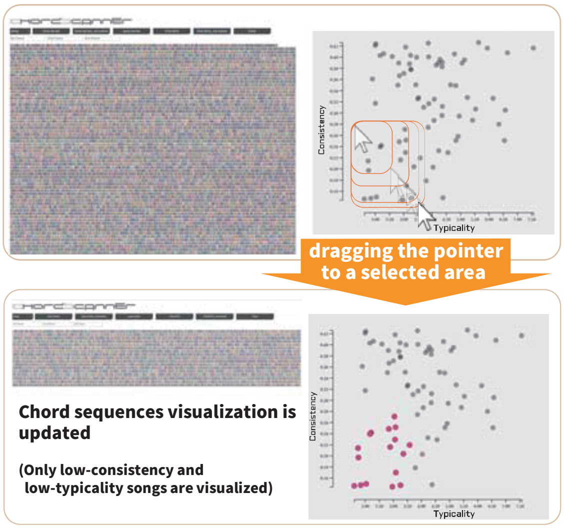


Figure 5. Targeting pieces based on CT scatterplot. The user can refine his search by dragging the pointer to a selected area (Fig. 1-B).

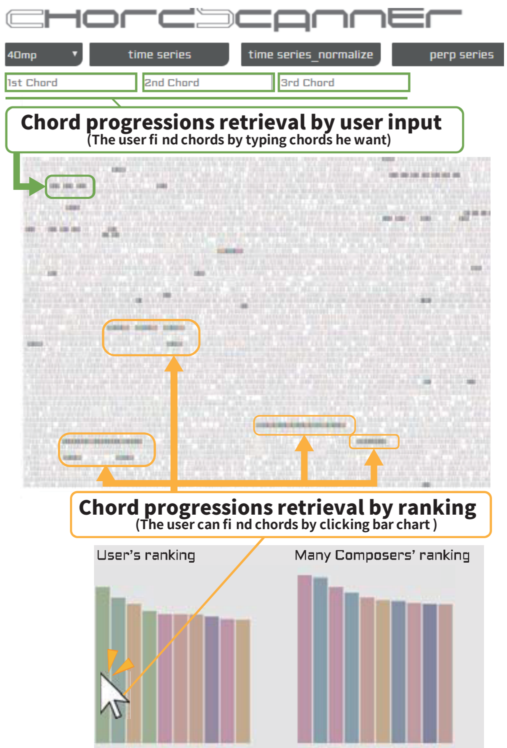


Figure 6. Procedures of highlighting chords based on two kinds of user input. The user can find where the ranking chord appeared in their pieces by clicking ranking bar (Fig. 1-C).

We assume that chord progressions that are included in both rankings are very typical chords and thus cannot represent the character of a composer. Hence, we considered modifying the user's dataset ranking by eliminating from the user's ranking chords that appear in many composer's datasets (Fig. 7).

However, it is still unclear how a number of chord progressions will be overlapped. We, therefore, examined how numbers of chord progressions will be overlapped with these rankings, based on two kinds of dataset groups. We also interviewed a professional composer to get feedback about the rankings and interfaces.

4.1 Experimental procedure

We collected two different kinds of datasets for calculating the generative probability of chord progressions. First one, we targeted 7 amateur and semi-professional composers and chord progressions of their 457 songs analyzed by Songle [12] as Vocaloid [13] and a desktop music software database group (hereinafter called Vocaloid group). We chose these composers and pieces based on the amounts of pieces they composed. The average number of songs was 64.29 (max=74, min=54). In addition, we used 177 songs of 6 professional J-POP artists and collected chord sequences from the (published) scores of their songs and modified manually, as J-POPS songs database group (hereinafter called "J-POPS group"). The average number of songs was 29.5 (max=31, min=26).

4.1.1 Comparing generative probability ranking of one composers' with the ranking of many composers'

We got a musical key sequence of these songs by transposing them to C major because the choice of chords depends on the musical key. The model parameters of the HPYTM were trained by using the Gibbs sampler (the switching token-based sampler [11]) with 500 iterations、and the model parameter of the VPYLM were trained by using the Gibbs sampler with 500 iterations. ChordScanner is assumed to be used by users who input their composed pieces. In this experiment, we treated one composer's pieces as data for calculating consistency.

4.1.2 Interview about chord progression ranking and UI

We asked the subject, a professional composer, to listen to two composer's pieces before we started this interview. We chose those two composers from the Vocaloid group. In the interview we first showed the subject Table 1 without the column headings and then asked which rankings were calculated with which datasets. We also got feedback on our interface by using composer 1(40mp)'s dataset. After we explained how to use our interface, the subject used it.

4.2 Results and feedback from the interviewee

We obtained rankings of typicality and consistency of chord sequences in two datasets: Vocaloid group dataset and J-POP group dataset. The ranking of each dataset is shown in Table 1 and Table 2, respectively. The leftmost column in the table shows the ranking sorted based on the typicality values which are calculated with all songs in the dataset. The other columns show the ranking based on the consistency values which are calculated among songs of a particular composer for each column; composer "40mp" (middle) and composer "Woodchuck" (rightmost) for Table 1, and composer "Porno Graffiti" (middle) and composer "Superfly" (rightmost) for Table 2.

4.2.1 Result: Comparing generative probability ranking of one composers with the ranking of many composers'

As shown in Table 1, the chord progressions among the rankings based on consistency did not match the top 10 chord progressions in the typicality ranking.

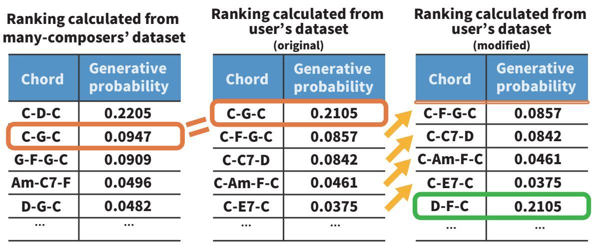


Figure 7. Modification of the user's generative probability ranking.

On the other hand, as shown in Table 2, the top-ranked chord progressions of the consistency ranking for composer "Porno Graffiti" (Cm7 B Cm7 B) did match to the 2nd ranked chord sequence regarding typicality. In addition, the top-ranked and the 2nd ranked chord progressions of consistency ranking for composer "Superfly" (Eb Bb F C, and Ab C Bb F Ab) also did match to the top and 6th ranked chord sequence regarding typicality.

4.2.2 Feedbacks about chord progression ranking and UI

We interviewed with a professional composer. The interviewee could tell correctly from the ranking which dataset was used for obtaining the ranks. We also got comments from the interviewee: *"Regarding the composer 1, there are frequently used tension chords and there exist chord progressions which represent somewhat a character or a style of the composer 1, which made it easier for me to tell which one is by the composer. On the other hand, the other composer (composer 2) does not have that kind of chords (add9), which also gave me a clue."*

This comment indicated that the ranking calculated with our method represented the style and the characteristics of chord progressions created by a composer and provided us with a sufficient clue to distinguish between styles.

The interviewee also commented that this system could support the user to understand about chord progressions. The composer also suggested us to include visualization of a music structure in the system. Since the SongleAPI provides structural information such as locations of chorus and verse sections, we plan to visualize musical structure in our system by using them.

Furthermore, the interviewee commented as: *"It is interesting to see which chord is repeatedly used in a song, and that is useful to me. I would like to know, if possible, the rare chord progressions; such as chord progressions which are only frequently used by a particular composer or a chord sequence which nobody has ever used."*

Although our system currently focuses on chord progressions which users tend to use often, we plan to cope with rare chord progressions in our system as future work.

4.3 Discussions

The initial objective of this investigation was to see whether we could adjust the consistency ranking by removing chord sequences which were highly ranked in the typicality ranking. However, as shown in the results, the overlap among rankings did not always exist. There was no overlap among rankings with Vocaloid songs whereas there was an overlap with J-POP songs. Although removing the overlapped chord sequences was our initial strategy to improve the accuracy of the consistency rankings, this strategy may be valid only for datasets with certain properties. Therefore, we plan to take another approach such as calculating typicality among chord sequences which is highly ranked in the consistency ranking, as discussed in Section 3.

We also want to consider the issue that similarity measure between chords may differ among composers. Some composer may think the inversion chords are the same as the chord in the original form, but others may not. We plan to implement a function to control what are the same chords based on the composer's preferences.

Furthermore, we are planning to include a function which recommends chord sequences by letting the user to select several songs. This function could be useful to create a list of chord sequences which the composer wants or does not want to use in the future. It could also be useful in gaining an understanding of existing music. Visualizing the ratio of usage of chord sequences included in both rankings is also part of our future work. This visualization may allow the user to create novel music which is still acceptable to the user.

5. CONCLUSIONS

We proposed ChordScanner, which enhances a user's understanding of chord progressions he or she tends to use frequently. This system visualizes the time-sequence of chord progressions and highlights their typicality with user's interaction. We also proposed a method using a probabilistic model to calculate consistency and typicality. In addition, we investigated the overlap between rankings obtained with both consistency and typically.

We used only Japanese music in the datasets, and the experimentation was limited. In a future study we will try another kind of music genre (e.g., techno, traditional, or classical) and compare several country's styles using the same genre; for example, J-POP and K-POP. We also will interview various kinds of clients—musical producers, movie makers, and corporate communications officers—to get feedback about what kinds of visualization are useful for users who do not have musical knowledge. Furthermore, we will conduct experimentation in which participants make a new piece by referring to our system.

Acknowledgments

This work was supported in part by JST ACCEL Grant Number JPMJAC1602, Japan.

6. REFERENCES

[1] C. Z. A. Huang, D. Duvenaud, and K. Z. Gajos, "ChordRipple: Recommending chords to help novice composers go beyond the ordinary," *Proc. IUI 2016*, pp. 241–250, 2016.

[2] D. Morris, I. Simon, and S. Basu, "Exposing parameters of a trained dynamic model for interactive music creation," *Proc. AAAI2008*, pp. 784–791, 2008.

Rank	7 composers' ranking	Composer 1's ranking (40mp)	Composer 2's ranking (Woodchuck)
1	C | Dm | F/C | C	Fm7 | A#m7 | D#m7 | G#	G#m | C# | F#
2	G | Cm | Fm | G	G/C | Cadd9 | Dsus4 | Em7	A#/F | C | Am | Dm
3	C | Dm | F/C | C | Dm | F/C	Cm9 | F6 | A#M7 | Gm7	Cm | A# | Cm
4	G/D | C/D | G/D | C/D	E6 | Nadd9 | F# | G#m7	Dm | Gm | Dm | Gm
5	Dm | Gm | D# | F	Nm | D | G6 | A	A | Bm | Em
6	E | D6 | A | N | E | D6	Gm7 | D#m | F | Gm7	F#7 | B
7	G#M7 | G | G7 | Cm7	Em9 | A6 | DM7 | Nm7	G | A | F#m | Bm
8	C# | A#/D# | D#m	Cadd9 | Dsus4 | Em7 | G/C	Am | F#m7 | B
9	A | F#m | D | E	F# | G#m7 | E6 | F#	Am | D | G | Em
10	Dm7 | C/D | Dm | C/D	F# | Nm | D | GM7	C#m7 | F#6 | D#m

Table 1. Chord progressions rankings of Vocaloid group datasets: the key is normalized in C.

Rank	6 composers' ranking	Composer A's ranking (Porno Graffiti)	Composer B's ranking (Superfly)
1	<u>Eb | Bb | F | C</u>	**Cm7 | B | Cm7 | B**	<u>Eb | Bb | F | C</u>
2	**Cm7 | B | Cm7 | B**	Fm7 | E#7alt5 | D/A9	<u>Ab | C | Bb | F | Ab</u>
3	Ab | C | Bb | F	C | B | C | B | C | B	Abm7 | C
4	F | C | F | C | F	E9 | F	F/C | C/G | F/C
5	F | E | F | E | F | E	G | Dm7 | G | Dm7	Gm | F
6	<u>Ab | C | Bb | F | Ab</u>	D9 | Cm7 |Fm7 | E#7alt5	F | C | Eb | Bb
7	E/A9 | A9 | E/A9 | A9	Fm7 | E#7alt5 | D/A9 | Cm7	E | F/Cadd11 | Cm7 | E
8	Ab | Gb | Ab | Gb | Ab	Cm7 | B | Cm7	F/C9 | Ab7 | Bb6add9
9	Ab | Bdim | Ab	G7sus4 | G7sus4 | G7sus4 | G7sus4	G | Eb | C | Eb
10	C | Abm7 | C	F7alt5 | Csus4	Esus4add7 | E/B | Am

Table 2. Chord progressions rankings of J-POP group datasets: the key is normalized in C.

[3] I. Simon, D. Morris, and S. Basu, "MySong: automatic accompaniment generation for vocal melodies," *Proc. CHI 2008*, pp. 725–734, 2008.

[4] J. Heer, N. Kong, and M. Agrawala, "Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations," *Proc. CHI 2009*, pp. 1303–1312, 2009.

[5] A. Hayashi, T. Itoh, and M. Matsubara, Colorscore: Visualization and condensation of structure of classical music. in *Knowledge Visualization Currents*, Springer London. pp. 113–128, 2013.

[6] M. Uehara and T. Itoh, "Pop music visualization based on acoustic features and chord progression patterns applying dual scatterplots," *Proc. SMC 2015*, pp. 43–48. 2015.

[7] T. Nakano, D. Mochihashi, K. Yoshii, and M. Goto, "Musical typicality: How many similar songs exist?" *Proc. ISMIR2016*, pp. 695–701, 2016.

[8] T. Nakano, K. Yoshii, and M. Goto, "Musical similarity and commonness estimation based on probabilistic generative models of musical elements," *International Journal of Semantic Computing*, vol. 10, no. 1, pp. 27–52, 2016.

[9] D. Mochihashi and E. Sumita, "The infinite Markov model," *Proc. NIPS 2008*, pp. 1017–1024. 2008.

[10] K. Yoshii and M. Goto, "A vocabulary-free infinity-gram model for nonparametric Bayesian chord progression analysis," *Proc. ISMIR 2011*, pp. 645–650, 2011.

[11] H. Noji, D. Mochihashi, and Y. Miyao, "Improvements to the Bayesian topic n-gram models," *Proc. EMNLP 2013,* pp. 1180–1190, 2013.

[12] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A web service for active music listening improved by user contributions," *Proc. ISMIR 2011*, pp. 311–316, 2011.

[13] H. Kenmochi, and H, Ohshita, "Vocaloid-commercial singing synthesizer based on sample concatenation", *Proc. INTERSPEECH*, pp. 4009–4010, 2007.