Naturalness of Mandarin Cloned Songs

Yundi Xu

Master Thesis: General Linguistics

Supervised by Paul Boersma
University of Amsterdam
June, 2025

Abstract

This study investigates the acoustic cues that might distinguish unnatural cloned singing voices from human singing voices. The examined acoustic cues include pitch, intensity, jitter, shimmer, Cepstral Peak Prominence and Harmonics-to-Noise Ratio. For comparison, three groups of vocal sources were created: original versions, cover versions and cloned versions. The cloned versions were synthesized using a voice model created from the Opencpop corpus. Pitch contour graphs were used to describe patterns, while the other cues were compared through linear mixed models. The results showed that the acoustic features jitter and shimmer are distinguishing factors for cloned versus human singing voices, since cloned songs always exhibited lower jitter and shimmer. These low values might result from over-regularization during the singing voice cloning process, which led to a perception of emotionlessness. Intensity is another cue that explains why cloned songs are perceived as less natural than human singing songs, but this study could not draw conclusions on spectral tilt, CPP or HNR.

Contents

1. Introduction

1.1. Voice Synthesis and Music Clone

Voice Synthesis has grown into a mature field in recent years. It has evolved from formant synthesis to technology with algorithms, machine-learning and perception-based modelling (Malisz et al., 2019). The development enhances its intelligibility and coherence, and expands its scope of application beyond simply generating synthetic voices. That is where voice cloning comes into play.

Voice cloning is a subfield of voice synthesis that replicates the voice of a specific individual to replace the content that they never actually recorded. In other words, it endows the given synthetic audio with another identifiable voice identity (Rosi et al., 2025). The technology builds on Text-to-Speech (TTS) systems and is enhanced by formant synthesis and neural networks (Zhang, 2024).

Voice cloning first requires the creation of an artificial voice that resembles the target human voice. During this process, high-quality audio recordings of the target speaker are needed for acoustic feature extraction. These extracted features are successively used by neural networks and voice decoders to train a voice model. Then, given texts are inputted into the trained model to generate new speech. Likewise, to clone new songs, given vocal tracks are inputted to generate new vocal tracks.

There are at least two approaches to cloning a voice. The first is speaker adaptation, which fine-tunes multi-speaker generative models to match the voice nuances of a specific person in accordance with their features. It demands less data from that person. The other is speaker encoding, which estimates and employs the voice of an unseen speaker to directly create a model (Arik et al., 2018). Regardless of their difference in generating the final models, they both undergo the process of feature extraction and reconstruction, where mistakes are likely to occur and naturalness might be compromised.

In fact, a couple of studies have suggested that synthesized voices are lower in naturalness compared to real human voices. Synthesized voices are found to be less intelligible, pleasant, likable and natural than human voices (Kühne et al., 2020; Zhang, 2024). The deficiency leads to human ability to distinguish synthesized voices from real ones with around 70% precision (Mai et al., 2023; Müller et al., 2022; Warren et al., 2024). The existing literature mostly employs perception tasks, with a lack of acoustic feature investigation. In recent years, more researchers have noticed this absence, Zhang (2024) attributed the lower pleasantness than human voices to physical voice qualities, such as pitch and tone. Nussbaum et al. (2025) suggested unnatural voices may deviate from human voices in pitch contour, temporal structure, or spectral composition.

These days have seen voice cloning expanding from speech to music, where the original vocal tracks are replaced by those of cloned voices. However,

unlike voice cloning in speech, cloned songs have been rarely evaluated in terms of naturalness (Nussbaum et al., 2025). As a blend of musical and linguistic elements, cloned songs are expected to demonstrate unique features in addition to those from cloned speech.

1.2. Naturalness

Naturalness is a highly influential factor in distinguishing machine-generated from real human voices. Originally derived from biology, it referred to an adaptive form, with its opposite being deviation. When introduced into phonetics, this concept is mostly adopted as the equivalent of human-likeness but remains conceptually undefined (Cooper et al., 2024). Nussbaum et al. (2025) proposed two types of naturalness: the first is the deviation-based naturalness. It assumes a reference that represents maximum naturalness. The second is the human-likeness-based naturalness, which emphasizes the similarity to human voice.

The reason for this undefined situation is the vagueness of human-likeness. Human-likeness is always considered as a subjective judgment. Seebauer et al. (2023) connected features like “fluttering”, “strange”, “irritating”, “metallic” with human-unlikeness. Concerning the topic of cloned songs, the focus is to distinguish human vocal songs from machine deepfake ones. Thus, naturalness is defined as the opposite of unnaturalness, which refers to mechanical and robot-like features here.

The evaluation of naturalness is typically conducted through two approaches.

The first is a subjective approach through perception tasks like listening tasks. Participants are presented with audio-clips of human and synthetic voices, and rate them on scales (Kühne et al., 2020). But these tasks are influenced by individual bias and may lack consistent standards, which lowers reproducibility and may lead to unstable conclusions (Mayo et al., 2011; Xiong et al., 2023).

The other approach is more objective, eliminating the random bias of individual variation. Software engineers and researchers need to assess the feasibility of their newly developed models. They evaluate metrics like jitter, shimmer, spectral slope, cepstral peak prominence(CPP), fundamental frequency(F0) and Mel-cepstral distance(MCD) by programs like openSmile package (Eyben et al., 2010) in Python. Compared with perception tests, this approach is adopted more by machines as the discriminating way.

From a linguistic perspective, these acoustic metrics require more careful interpretation under the difference. For example, linguistics provides an explanation for the jitter difference between cloned and human voices, pointing out why one is higher than the other. This perspective compensates for the simplified interpretation of acoustic cues in computer science. Therefore, this study intends to measure these objective metrics with the awareness of linguistics.

1.3. Acoustic Cues

Humans depend on linguistic features such as prosody, accent and fluency to make judgments about the sounds they hear. The appreciation for music further demands harmony in timbre, dynamics and genre.

For machines, measurable acoustic cues are their objects for assessment. To date, few linguistic studies have explored the acoustic cues of synthetic songs. Given its blend of speech and music, a relatively comprehensive and valid evaluation of the naturalness of synthetic songs requires attention to at least two dimensions: vocals and melody. Since cloned songs retain the original melody, the emphasis should be laid onto vocals.

Jitter, shimmer, spectral flux, dynamics, cepstral peak prominence (CPP) and the fundamental frequency (F0) explain voice quality (Hinterleitner et al., 2015; Seebauer et al., 2023). Despite the lack of evaluation in cloned songs naturalness, some existing literature has investigated another branch of synthetic music, which is AI-generated music. Composed by artificial intelligence, those works are new in both vocal and backing tracks. That is to say, from the melody and lyrics to the vocal and singing techniques, they do not use other works as reference. Compared to standard Text-to-Sound (TTS) synthesis, it considers a wider range of contextual factors when it comes to metrics like dynamics and pitch (Nishimura et al., 2016). The acoustic cues such as spectral features, excitation parameters, waveform and duration are always measured (Nishimura et al., 2016), which can be referred to.

In Mandarin Chinese, tone is one of the most critical features, since pitch contour carries lexical meanings. Thus, unnatural transitions in pitch trajectory may affect perceived naturalness.

In addition to pitch, a comprehensive investigation of voice quality is also needed. Spectral tilt, the decrease in intensity of higher harmonics in the sound spectrum, can manifest voice quality and is estimated to exhibit a difference between synthetic and human voices (Garellek, 2022). Breathy voices show a higher spectral tilt than normal voices, which in turn are higher than creaky voices, often associated with deficient synthesized voices.

Cepstral Peak Prominence (CPP), the difference between the cepstral peak and the corresponding line, is also highly correlated with breathiness. Higher CPP indicates smoother sound quality, while lower values are always more creaky. It is always combined with Harmonics-to-Noise Ratio (HNR), the hoarseness measurement by calculating the ratio between periodic part and noise.

Jitter, the variation in frequency, quantifies the perturbation resulting from unstable vibration of vocal folds. Higher values of jitter often contribute to a voice sounding rough and hoarse. But even in healthy voices, it is expected to observe a small amount of jitter because of the complexity of vocal fold control. Usually, individuals will differ in jitter which helps to distinguish one from another.

Shimmer, the variation in amplitude, is also used to evaluate voice quality

combined with jitter. The human production system usually generates unstable fluctuations in amplitude. Higher values of shimmer often suggests a hoarse, breathy and rough voice. Its range of values also distinguishes speakers.

As previously stated, cloned songs are different from cloned speech since they are sung instead of spoken. Even if the melody dimension is not the focus, this study still aims to provide another perspective for assessing cloned music, which is a musicological perspective.

Voice range profile (VRP) is first expected to be this measurement. It is commonly used in musicology to evaluate a singer's vocal capacity through the maximum range of pitches and dynamics of a vowel. However, since VRP is usually measured under a controlled environment where singers are instructed to sing the same phoneme from low to high pitch, it is not plausible to measure it using these songs as materials. Thus, this study uses intensity as a proxy. Although intensity also demands identical recording environment, it can be normalized and compared. Intensity measures the energy in the amplitude waveform. It reflects how the dynamics flow within one sentence or one segment.

1.4. Research Questions

Unnaturalness of cloned songs is the opposite of human-like features, and it is expected that acoustic features of cloned songs are significantly different from those of original or human cover songs in perception and objective cues.

Thus, this study aims to investigate what the difference in acoustic cues is. The focused research question is

Can pitch, intensity and voice quality acoustic metrics serve as distinguishing factors between cloned and real human singing voices?

To address this question, a set of acoustic features were selected and compared between cloned songs, their original versions and the counterparts covered by the singer from the corpus Opencpop. The features include

1. Pitch
2. Intensity
3. Spectral Tilt
4. Cepstral Peak Prominence (CPP)
5. Harmonics-to-Noise Ratio (HNR)
6. Jitter
7. Shimmer

The results are assumed to show different features of pitch contour between cloned and human singing voices. Meanwhile, statistically significant differences are expected to be found between the average of human and cloned singing voices in some or all of other calculable acoustic cues. Higher values in cloned songs are expected in terms of jitter and shimmer, while higher values in human singing songs are expected for CPP and HNR.

2. Methodology

This study examines disparities in acoustic features between Mandarin pop songs sung by humans and their cloned counterparts to identify objective correlates of naturalness.

2.1. Materials

Three versions of vocal data are categorized into Group A, B and C. Each group comprises 5 songs and a total of 15 stimuli were analyzed (Appendices).

2.1.1. Group A: Cover Versions

Audio sources were extracted from Opencpop, a Mandarin singing corpus (Wang et al., 2022). It consists of 100 Mandarin pop songs (5.2 hours) performed by a professional female singer under a controlled condition. The recording files were recorded at 44.1 kHz. The corpus also includes annotated TextGrid files (Graph 1) marking sentences, characters, syllables, notes, duration, segments and melismas. No singer demographic data is available. Access to this corpus has been permitted.

The 5 target stimuli were randomly selected from the inventory.

2.1.2. Group B: Original Versions

Original versions consist of the five same songs performed by their original artists. All selected songs were performed by different female singers.

2.1.3. Group C: Cloned Versions

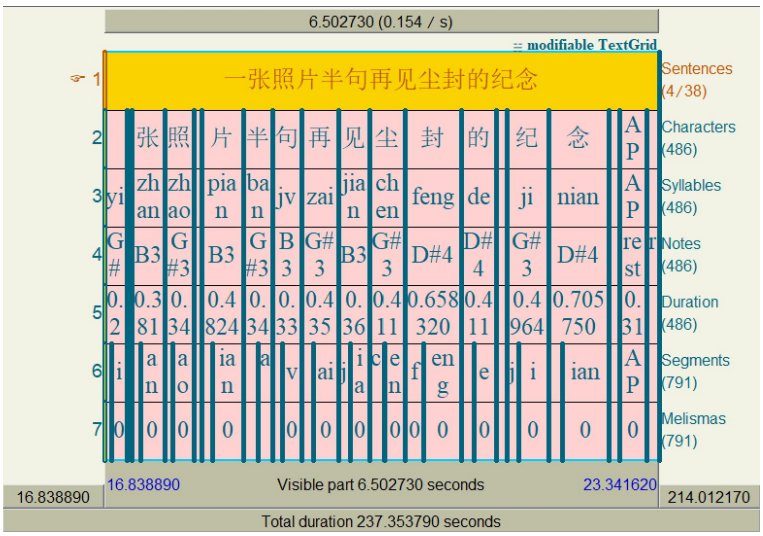
Cloned songs were implemented through Ultimate Vocal Remover 5 (UVR5) and Retrieval-based-Voice-Conversion-WebUI (RVC).

The source for the voice model was from Opencpop. The voice model used 5 songs in its list, other than the 5 songs used as cloning stems. These songs were randomly selected. The recordings in Opencpop were unaccompanied singing, so no further removal of backing tracks was needed. The blank parts were cut. The acoustic features in these isolated vocals were extracted and used to train the target voice model.

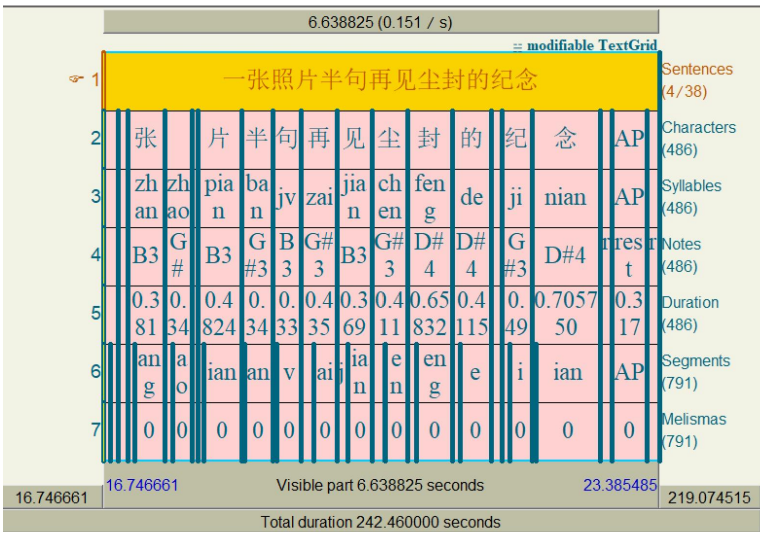
The vocal stems were the clean versions of Group B, whose instrumental tracks had been removed by UVR5. Target cloned versions were finally synthesized by RVC, depending on the trained voice model and vocal stems, yielding cloned vocals without backing tracks. The detailed cloning process will be described in the section “Cloning Process”.

2.1.4. Textgrid Annotation

The primary annotation is in Praat Textgrid format from the Opencpop corpus. Each textgrid file comprises 7 tiers (Graph 1, Graph 2): (1) sentences, (2) characters, (3) syllables, (4) notes, (5) duration, (6) segments, (7) melismas.



Graph 1 TextGrid File for Sound File 2004 Cover.wav (One in Group A)

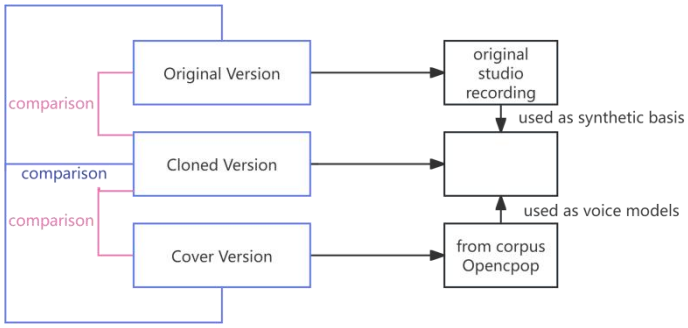


Graph 2 TextGrid File for Sound File 2004 Original.wav / Clone.wav (One in Group B and C)

For the segments tier, Mandarin employs *pinyin*, a syllabic transcription system where consonantal onsets and syllable finals compose syllables. This means finals in Mandarin are not only simple vowels but also composites containing a nucleus (a vowel or vowels) and often a nasal final, such as /ua/, /ian/ and /ing/. Therefore, in this tier, consonantal onsets and syllable finals, rather than the standard IPA, are observed.

The Textgrid files from Opencpop corpus are suitable for Group A, but not for Group B and C. While original and cover versions may be perceptually similar in temporal distributions, they diverge from cloned songs in time. To ensure accurate comparisons, the TextGrid annotations from the Opencpop corpus were manually verified for the Group A, and the TextGrid files for Groups B and C were adjusted based on them. During adjustment, the researcher manually dragged the boundaries in the segments tier to align with the actual performance, based on visual inspection of the waveform and

auditory perception. The sentence boundary changed along with the adjustment of segment boundaries. Function “rubber banding” in TextGrid Window was used, so that boundaries within a selection could be dragged together with the Option and Command keys pressed.



Graph 3 The Comparisons between Groups

2.2. Cloning Process

The cloning process was performed by RVC WebUI. The first step was to extract features and create a voice model. It required audio data whose duration was between 10 to 50 minutes. In this study, the total duration of 5 used sound files was 19 minutes 46 seconds, with the empty part omitted. These truncated files were uploaded to RVC and automatically traversed by the system. The processed audio files were then used to extract features through the “rmvpe” algorithm, which yielded the best result as its instructions said. The subsequent model training had a total of 500 epochs, and a batch size of 4 was employed per GPU. The trained model timbre was saved in a .pth file format, and the efficient features were saved in an .index file.

The second step was to infer the new cloned songs, utilizing the .pth file and .index file. Because the features were extracted from a female singing voice and were going to be implanted in female singing voices, no modified tone was applied. The “rmvpe” algorithm was selected and the contribution of retrieved features was 50%. After setting all these parameters, the cloned songs were automatically conversed.

2.3. Acoustic Measurement

The chosen acoustic measurements were grouped into sentence-level and segment-level analyses. The feature extraction was performed by Praat (Boersma & Weenink, 2025). Three scripts (Appendices) were created for batch processing.

2.3.1. Sentence-level Cues

2.3.1.1.Pitch

Fundamental frequency was extracted using the “To Pitch (raw cross-correlation)” function in Praat on a sentence basis. The pitch contour graphs were the final outputs.

2.3.1.2.Intensity

Because some intervals involved a silent segment or were totally silent, not all intervals were included. In fact, all intervals that contained a minimum intensity lower than 0 were excluded, since 0 dB typically represents the threshold of human hearing. The range per sentence was measured as the

difference between the 90th percentiles and the 10th percentiles, to mitigate the influence of extreme values.

2.3.1.3.Spectral Tilt

Spectral tilt was measured per sentence to observe the difference between the high energy zone and the low energy zone within the same sentences.

2.3.2. Segment-level Cues

2.3.2.1.Jitter

Jitter is usually measured on sustained vowels because they provide easier observations of micro-instabilities. Since the production was within songs and no deliberate extension of vowels was allowed, this study extracted all segments including vowels in Mandarin (Appendices). The difference was calculated in milliseconds(ms). It was measured in Praat through the function “To PointProcess Periodic- Get jitter”.

2.3.2.2.Shimmer

For the same reason, shimmer was measured on all segments including vowel sounds. The measure was expressed in decibels (dB) via the function “To PointProcess Periodic- Get shimmer” in Praat.

2.3.2.3.Cepstral Peak Prominence (CPP)

Segments involving vowels were extracted from the phoneme tier in the TextGrid. The corresponding parts in .wav files were used to create CPP by the function “To PowerCepstrogram”.

2.3.2.4.Harmonics-to-Noise Ratio (HNR)

It was also applied to segments involving vowels. HNR was obtained through the function “Analyze Spectrum - To Harmonicity(cc)”. The minimum pitch value was assumed to be 60 Hz.

2.3.2.5.Deviation from Medium Intensity

The deviation was measured on all segments instead of only those which include vowels. The intensity variation within one segment is much lower than that in a sentence. In this sense, it is implausible to get comparable values through difference between the 90^{th} percentiles and the 10^{th} percentiles. Therefore, instead of selecting a range, this study measured the difference value between the average intensity of a segment and the median intensity of this file. To prevent the extreme values, those segments with a negative intensity were excluded.

2.3.2.6.Spectral Tilt

Spectral tilt was also measured on the segment basis to see whether a difference exists.

2.4. Statistics

The data was processed through R studio (R Core Team, 2024)

To investigate which acoustic features characterize the unnaturalness of cloned songs, this study set a ternary contrast. It was between cover (Group A), original (Group B) and cloned versions (Group C). In the following exploratory research, two pairs of binary contrasts were set. The first one was between cover

(Group A) and cloned versions (Group C), while the second was between original (Group B) and cloned versions (Group C).

The statistical comparisons were made across intensity, spectral tilt, jitter, shimmer, CPP and HNR. Linear mixed models were employed to assess the acoustic differences across sources. The lmer() function from the lmerTest package in R was used with restricted maximum likelihood (REML) estimation. The dependent variables were respectively each observed acoustic cue. The between-participant variable was the source type, where cover (Group A), original (Group B) and cloned (Group C) were the three levels. The random variables included source and id. Id stood for segments located at the same place across three versions of songs. For example, 2004_01 would occur three times in cloned, original and cover versions respectively. The effect of source might vary randomly across different songs.

Pitch contour was compared graphically.

3. Results and Discussions

3.1. Pitch Contour

The number of pitch contour graphs at the sentence level was in total 666. The analysis skipped all blank graphs where no lyrics existed. To provide a clear description of pitch contour patterns, the analysis selected graphs labeled as tens or ten multiples (10, 20, 30...) across the 15 song samples for illustration. Although a limited number of pitch contours were illustrated here, their features

were representative and observed across the graphs. Since the cover versions, recorded by another singer instead of synthesized from the original version, was unaligned with the original and clone version, it could not be precisely compared to the original and clone graphs. In this case, most of the comparisons below were conducted between original stems and their cloned derivatives.

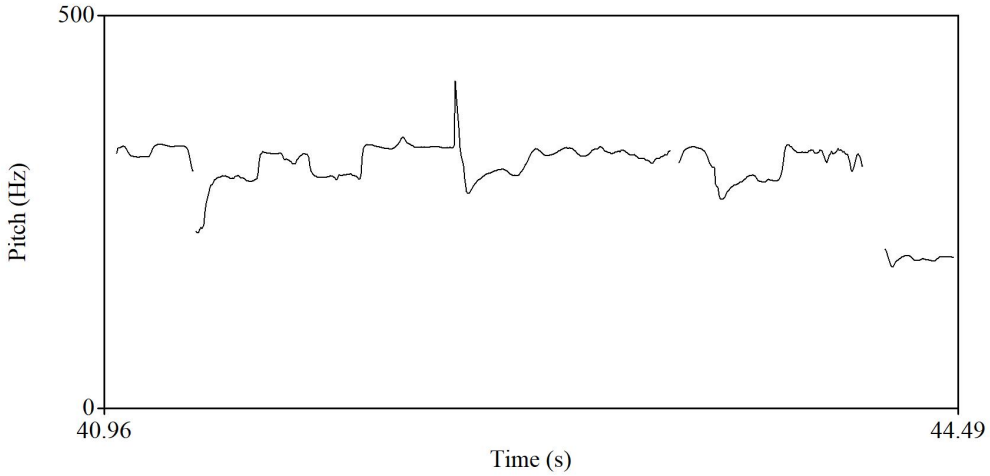
Between original and clone versions, similar to what had been observed in deepfake speech, the F0 sequence of human and deepfake (cloned) voices was similar but not identical in the same sentences (Warren et al., 2025). Their differences were mainly represented through the following aspects:

3.1.1. Continuity

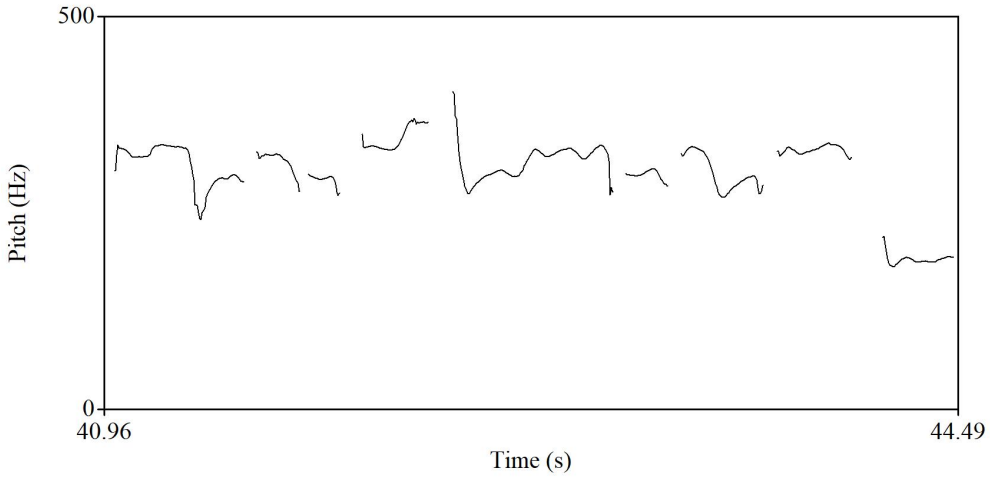
Across most comparisons, graphs of original versions exhibited the most continuous pitch contour (Graph 4) , while that of clone versions showed more fragmentation in the pitch curves (Graph 5). The gaps in cloned songs showed that some syllables connected in human singing had been broken during the cloning process. This discontinuity suggested that cloned songs might ignore connections between syllables, which always exist in human singing. The absence of syllable-to-syllable connections impacted song fluency at the microscope level, with absent holistic chunks of words sounding like breathing.

Breathing is a subtle factor that influences human perception of naturalness in speech subconsciously (Layton et al., 2024). Frequent breaths lead to inconsistent perception of songs, with each break disrupting the listeners' perception. In singing performance, proper breathing plays a huge part in the

naturalness and aesthetics of sounds (Wang, 2024). Although a certain number of breathing gaps enhance the naturalness of perception, the imperceptible mixed breathing is what singers usually adopt, which requires prolonged exhalation between words, different from the short breaks between syllables in the cloned songs. In addition, the normal pattern for breath in speech shows most breaths exist at the phrase and sentence boundaries, rather than the syllable boundaries.

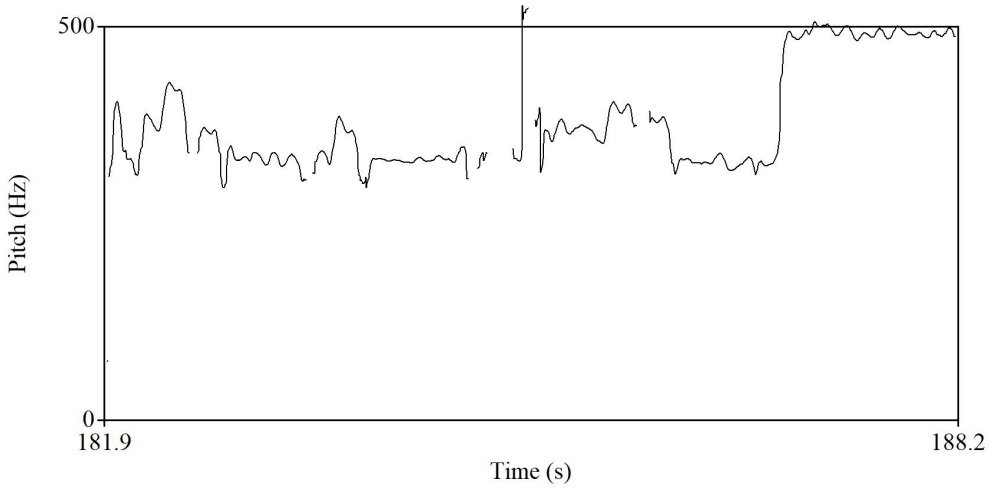


Graph 4 (2013Original-sentence 10)

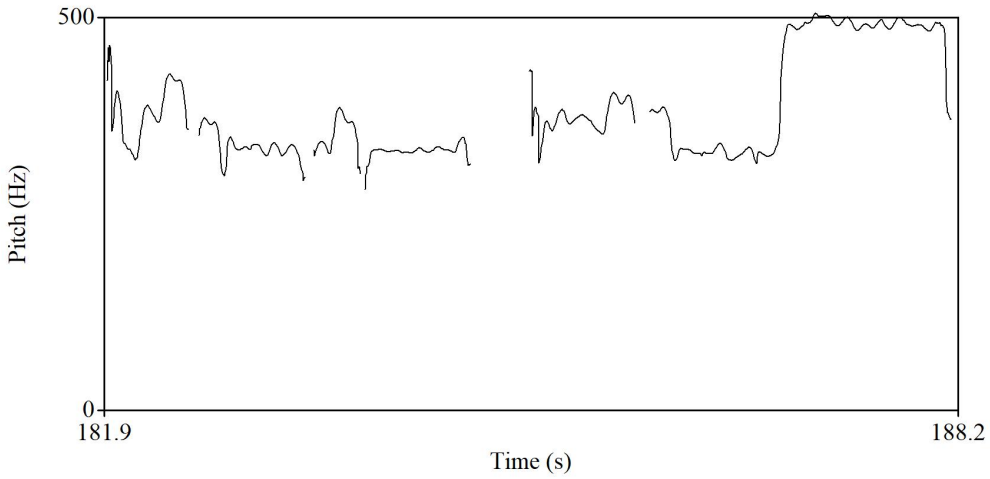


Graph 5 (2013Clone-sentence 10)

However, in a few times, original versions (Graph 6) might display more gaps than cloned versions (Graph 7). A drastic pitch leap always co-occurred with this deviation. In the following example, at around 185 seconds, a sudden jump to higher than 500 Hz in the original version was observed. In contrast, the graph from the clone song maintained a continuous pitch contour here.



Graph 6 (2004 Original-sentence 30)

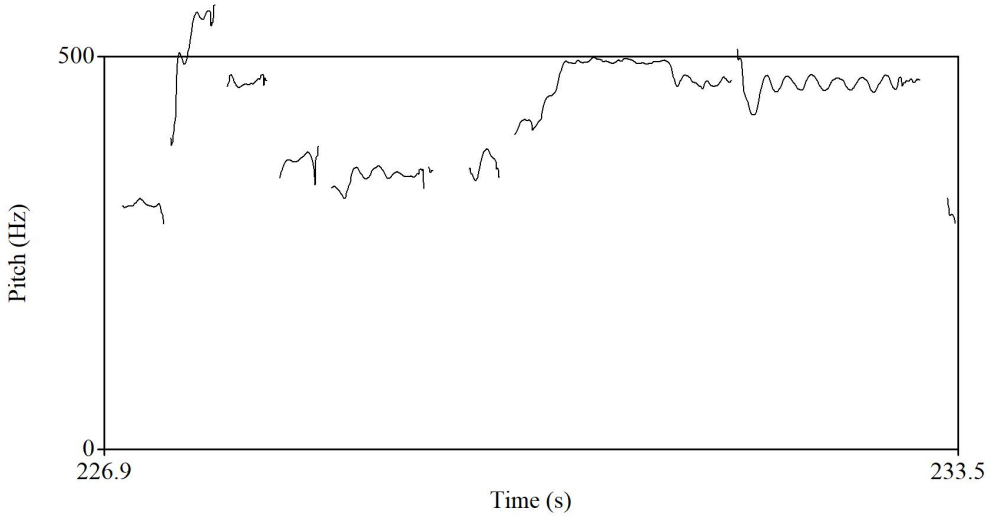


Graph 7 (2004 Clone-sentence 30)

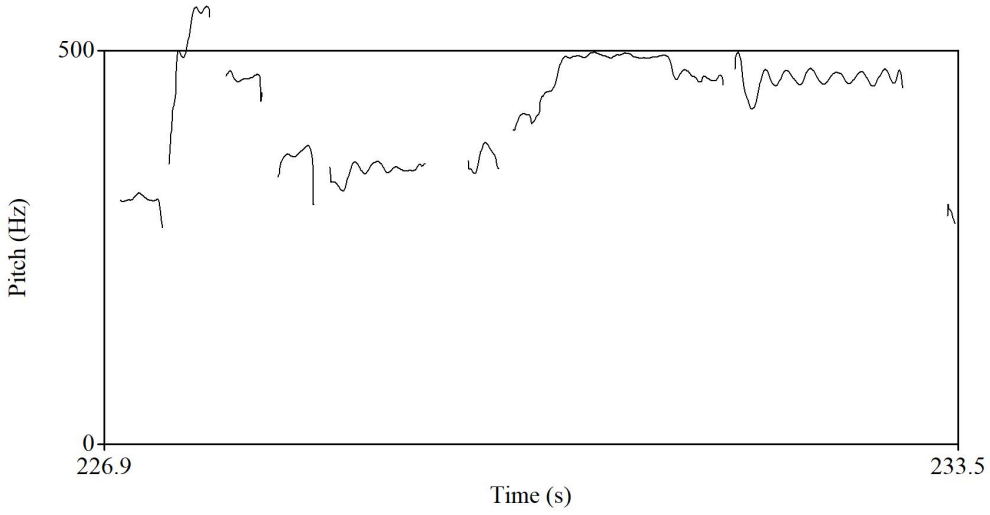
The sudden leap might indicate an error in showing pitch contours, so these graphs were discarded because of these technical errors.

3.1.2. Variability

Consistent with what Warren et al. (2025) found, synthetic audio could not perfectly mimic and generate the correct F0 sequence. Although the clone adaptation mirrored the broad structure of its cloned basis, the original version, it appeared less dynamic. Compared with cloned singing voices (Graph 9), original singing voices (Graph 8) had more subtle variations at the onsets and ends of each syllable.



Graph 8 (2059 Original-sentence 40)



Graph 9 (2059 Clone-sentence 40)

This partial discrepancy might indicate that cloned singing voices inhibited phrasing features from the original singing voices in the pursuit of fluency. The micro-vibrations in human singing voice had been observed by Ardaillon (2017), claiming that two types of fluctuations confer naturalness in singing voice. The first type is independent of the singer's skills and connected with voice mechanism and articulation. The second type, different from one singer to another, conveys expressiveness and enhances aesthetics of singing voice.

No matter which type the fluctuations found here belonged to, it could be assumed that the absence of these micro-vibrations might lead to reduced expressiveness, showing a limitation of cloning process in capturing and imitating emotional expressions. This deficiency might finally lead to more emotionally flat and robotic in cloned songs.

Despite how subtle these vibrations were, their importance was recognized. Naturalness would be enhanced if the target song is fitted with vibrations in reference speech (Umbert et al., 2015) .

3.2. Cloned vs. the Average of Cover and Original voices on files level

3.2.1. Intensity Range and Spectral Tilt at the Sentence Level

The intensity range, defined as the difference between the 90^{th} percentile and the 10^{th} percentile intensity, was compared between same sentences from a

set of files (its cloned, cover and original versions). The model used was "range_of_intensity.dB. ~ source + (source| song) + (1 | id)". The average value of cover and original songs represented human voices. The cloned version was perceived to be significantly higher than the human voices version (estimated difference = 2.332 dB, t = 3.470, 95% confidence interval = 0.568dB... 4.096dB, p = 0.02). The human singing voices themselves did not differ significantly from each other (estimated difference = 3.189 dB, t = 1.750, 95% confidence interval = -1.878 dB... 8.256 dB, p = 0.16). No significant distinction existed between human singing voices and cloned singing voice in terms of spectral tilt (estimated difference = 0.060 dB , t = 0.058, 95% confidence interval = -2.816dB...2.935 dB, p = 0.96). The model used was "spectralTilt ~ source + (source | song) + (1 | id)".

The widened intensity might be explained by an intensified mechanism adopted in singing voice synthesis. When humans sing, sub-glottal and vocal-fold pressure increases compared to the spoken status. Thus, the synthesis of singing voices deliberately adds pressure to increase the glottal formant's frequency and decrease spectral tilt (Ardaillon, 2017). It is possible that this mechanism was overused during the synthesis process, and the intensity was amplified in excess.

The random effect for id in terms of intensity range (standard deviation = 7.384 dB) and spectral tilt (standard deviation = 3.486 dB) was great, as the standard deviation of residual was respectively 3.638 dB and 1.719 dB. This

high randomness indicated certain differences took place within segments located at the same positions across versions. The random effect for song was relatively weak, both in intensity range (standard deviation = 2.624 dB) and spectral tilt (standard deviation = 1.366 dB), suggesting similarities across songs.

The above analysis gives a preliminary conclusion that the intensity range plays a part in distinguishing human vocal songs from cloned songs, while spectral tilt may not. Thus, a wider intensity range at sentence level may serve as an acoustic indicator for cloned songs.

3.2.2. Deviation from Medium Intensity and Spectral Tilt at Segment Level

At the segment level, intensity was compared by the deviation from the median value. The model used was "medium_difference.dB. ~ source + (source | song) + (1 | id)". Cloned versions were significantly lower than human voice versions (estimated difference = 1.868 dB, t = 2.821, 95% confidence interval = 0.031 dB...3.705 dB, p = 0.048) in the difference from median intensity, indicating less variation. There was no significant difference within human singing voices (estimated difference = 1.832 dB, t = 1.126, 95% confidence interval = -2.684 dB... 6.348 dB, p = 0.32).

Interestingly, even if intensity range within a sentence was wider in cloned singing voices than human singing voices, the intensity deviation from the median value at the segment level was more concentrated. In other words, the amplitude had been strengthened but the stability had been improved. This

finding was consistent with the observations at the sentence level, and might also be related to an imperfect imitation of the original intensity. The intensity contour had been found to vary the most at the beginnings and ends of segments and note sequences in human singing (Umbert et al., 2015). But since voice synthesis has deficiency in imitation at joint positions, these variations are very likely to be ignored during the synthesis procedure.

In terms of spectral tilt, the used model was “spectralTilt ~ source + (source | song) + (1 | id)”. No significant difference was found between human and cloned singing voices (estimated difference = 0.714 dB, t = 0.302, 95% confidence interval = - 7.550 dB... 6.068 dB, p = 0.78). Thus, no evidence showed that spectral tilt was an acoustic cue distinguishing human and cloned voices.

A higher random effect for id was also observed in deviation from median intensity (standard deviation = 5.316 dB) and spectral tilt (standard deviation = 8.082 dB), consistent with that at the sentence level. This emphasized the necessity to investigate the reason behind this high randomness. Id did not restrict the type of segments, and thus consonants and segments including vowels were investigated together, which might result in this high randomness .

The analysis at the segment level further confirmed that intensity is crucial in distinguishing cloned singing voices from human singing voices, with observations on either intensity deviation or intensity range.

3.2.3. Jitter and Shimmer

Jitter of cloned songs was observed to be significantly lower than that of human singing songs (estimated difference = 0.001, t = 3.333, 95% confidence interval = $-1.42*10^{-4}$...$1.65*10^{-3}$, p = 0.03). No significant difference was found within human singing songs (estimated difference = $5.52*10^{-4}$, t = 1.302, 95% confidence interval = $-6.34*10^{-4}$...$1.74*10^{-3}$, p = 0.26). The used model was "jitter ~ source + (source | song) + (1 | id)". However, contrary to the previous prediction, these two types exhibited inverse trends, with cloned songs having lower jitter.

In clinical linguistics, lower jitter is always connected with stable and healthy voices, while a higher jitter value is related to disordered voices. To explain the result in this study, the feeling dimension was introduced. Consistent with what Norrenbrock et al. (2011) demonstrated, natural synthetic voice displayed more perturbation to convey natural feelings. In their research, they found unit-selection-type synthesis was rated better than diphone synthesis in terms of naturalness, since the first one preserved more original perturbation from the inventory speaker than the latter.

Here, though seemingly steadier in frequency, cloned singing voice did not imitate the original perturbation in an idealized way. Without these variations, singing voices could be perceived as emotionless and consequently less natural.

As for the random factor, the highest influence among the three groups was id, with a standard deviation as 0.005. The standard deviation for residual was slightly higher than 0.005. The other random factor did not influence much on

the model. The random factor song (standard deviation = 0.001) showed a relatively lower value of variance.

Echoing the pattern found in jitter, cloned singing voice had a significantly higher value than its human voice counterparts (estimated value = 0.008, t = 3.509, 95% confidence interval = 0.001...0.015, p = 0.02), while no distinguishing difference was found within human singing voices (estimated value = 0.015, t = 2.326, 95% confidence interval = -0.003...0.033, p = 0.08). Its used model was “shimmer ~ source + (source | song) + (1 | id)”. The relatively lower shimmer value in cloned songs reflected subtle variations in intensity, in line with the observations in spectral tilt. Cloned songs exhibited a more stable output of amplitude, which was also out of expectations. Despite always as an indicator for more stable and cleaner voices, here, the lower value might indicate an over-regulation during the voice synthesis process .

Random factor id in shimmer showed a similar pattern as jitter, with a standard deviation 0.017. The standard deviation for residual was 0.019, so it was undeniable that variance across id was significant. The random factor song (standard deviation = 0.007) still did not show much difference.

From the observations on the difference between cloned songs and human vocal songs in jitter and shimmer, it might indicate over-regulation during the synthetic process. Even if voice cloning systems replicate human voices closely, they appear to fall short in emulating fluctuation in human vocals. For these frequency and amplitude micro-variations, it is very likely that the synthetic

system reduces them to enhance fluency. In this way, however, the innate vibrations involved in human voice are lost and naturalness is damaged.

In contrast, human singers always show more pitch and intensity modulations in their performance to convey feelings and emotions. In the meantime, the imperfect structure of vocal organs determines the inseparable vibrations in frequency and amplitude. During the synthetic process, however, some of the subtle variations are ignored by the models. Alternatively, they are perceived but deliberately suppressed due to the pursuit of smoothness and fluency, even though synthetic sounds are not constrained by physiological needs. This over-smoothing effects had already been perceived in singing voice synthesis and are reported to damage the naturalness of songs (Zhang et al., 2022).

Although the lower values of jitter and shimmer may enhance the fluency of songs, they indicate a damage to emotional expression and consequently diminish the perception of naturalness. The neglect and suppression of these vibrations play a vital role in distinguishing cloned singing voices from real human voices. Even if the songs are phonemically more accurate, they sound less “human”. However, this finding creates a paradox that higher acoustic fidelity in the pursuit of high fluency may come at the cost of emotional lacking.

Based on the above analysis, lower jitter and shimmer can be a phonetic marker for cloned singing voice. Even if synthetic models have made strides in

fidelity, they may still be unable to capture the subtle vibrations that conveys expressions or deliberately suppress them, which leads to a lower level of naturalness.

3.2.4. Cepstral Peak Prominence (CPP)

Cloned versions were found to exhibit higher values than original voices (estimated value = 1.507 dB, t = 3.600, 95% confidence interval = 0.348 dB... 2.667 dB, p = 0.022). The model used was " CPP.dB. ~ source + (source | song) + (1 | id)". A higher CPP value indicated greater periodicity, but when it exceeded a certain range, the voice quality might drop. Norrenbrock et al. (2012) found that the most muffled voice in their data exhibited the highest CPP values, but they claimed no clear borderline could be drawn.

Here, based on the previous assumption that cloning process might over-regularize sounds, a higher CPP value in cloned voices possibly suggested that the cloning process involved an enhancement of voice periodicity, which might be excessive compared to human singing. This aligned with the previous observations that synthesis prioritized signal stability at the expense of subtle irregularities. The micro-variations used to express feelings and style were ignored and discarded. Thus, the vocoders made the output sound clearer and more stable instead of being expressive. In addition, listeners might associate this higher periodicity with indifference.

Random by-song (standard deviation = 1.066 dB) variance indicated no huge difference occurred between songs. Though the standard deviation of id

(standard deviation = 2.639 dB) still remained the highest among them, it was relatively lower than the residual (standard deviation = 2.976 dB).

However, in terms of CPP, a significant difference between original and cover versions was perceived (estimated difference = 1.619 dB, t = 5.598, 95% confidence interval = 0.826 dB... 2.412 dB, p = 0.005). This finding was counterintuitive, given that both original and cover versions were sung by human performers. This difference will be further discussed in the "Exploratory Findings" section.

3.2.5. Harmonics-to-Noise-Ratio (HNR)

Cloned singing voice had significantly higher HNR than human singing voices (estimated difference = 1.868 dB, t= 2.821, 95% confidence interval = 0.031 dB... 3.705 dB, p = 0.047). The model used was "hnr.dB. ~ source + (source | song) + (1 | id)". The elevated HNR value in cloned versions suggested a stable periodicity, which was consistent with the findings on CPP. A distinguishing difference between original and cover versions was also perceived in HNR (estimated difference = 3.248 dB, t = 3.996, 95% confidence interval = -6.348 dB... 2.684 dB, p = 0.02). This counterintuitive finding, together with what was found on CPP, will be discussed in "Exploratory Findings" section.

3.2.6. Exploratory Findings

Based on the above analysis, intensity range at the sentence level, deviation from median intensity at the segment level, jitter and shimmer have been

observed to distinguish cloned singing voices from human singing voices. No significant distinction has been found between the original and cover version in these metrics.

However, the situation becomes difficult when a significant difference was also perceived within human singing voices, i.e. the cover version and the original version. These distinctions existed in CPP (estimated difference = 1.619 dB, t = 5.598, 95% confidence interval = 0.826 dB... 2.412 dB, p = 0.005) and HNR (estimated difference = 3.248 dB, t = 3.996, 95% confidence interval = 0.995 dB... 5.501 dB, p = 0.02). In intensity and spectral tilt at the segment level, although no significant difference between cover and original versions has been found, the random effects for song between them are high. The standard deviation of intensity is 3.632 dB and that of spectral tilt is 5.477 dB, both higher than that of residuals (standard deviation of intensity residual: 3.346; standard deviation of intensity spectral tilt: 5.079). These findings introduced further questions: How can we ensure these acoustic cues play a part in distinguishing cloned songs from human singing songs? Is there any pattern?

To explore these questions, this study set two contrasts. One was between cloned and cover versions, while the other was between original and versions.

3.2.6.1.Intensity Range and Spectral Tilt at Sentence Level

Cloned songs showed a significantly higher number in intensity range than original songs (estimated difference = 4.046 dB, t = 4.815, 95% confidence interval = 1.706 dB... 6.387 dB, p = 0.01) but not than cover songs (estimated

difference = 0.602 dB, t = 0.458, 95% confidence interval = -2.951 dB... 4.155 dB, p = 0.67). In both comparisons, cloned songs exhibited a wider intensity range.

No conspicuous significance was observed in spectral tilt between either type of human singing voices and cloned voice.

3.2.6.2. Deviation from Medium Intensity and Spectral Tilt at Segment Level

Cloned songs exhibited a significant lower value in deviation from medium intensity at the segment level than original songs (estimated difference = 0.963 dB, 95% confidence interval = 0.226 dB... 1.700 dB, t = 3.632, p = 0.02), while no significant difference was found between cover and cloned voices (estimated difference = 2.762 dB, t = 1.910, 95% confidence interval = -1.253 dB... 6.776 dB, p = 0.13). But one commonality lied in that human voices both manifested a wider deviation from medium intensity on average. Same with the findings on research questions, the random effect for id was found to be very obvious.

No obvious difference between any one of human singing voices and the cloned voice was perceived in spectral tilt either.

3.2.6.3. Jitter and Shimmer

In terms of jitter, cloned songs were observed to exhibit significantly lower value than original songs (estimated difference = 0.001, t = 2.937, 95% confidence interval = $4.77*10^{-5}$... $2.23*10^{-3}$, p = 0.04), but not between the cover and cloned versions (estimated difference = 0.001, t = 2.116, 95%

confidence interval = $-1.91*10^{-4}$ ... $1.46*10^{-3}$, p = 0.10). Across the two contrasts, numerically lower jitter in the cloned voice compared with the human voices suggested less fundamental frequency variation in synthetic voices.

Likewise, shimmer values between cloned and cover songs did not differ much (estimated difference = 0.001, t =1.025, 95% confidence interval = -0.002 ... 0.004,p = 0.36). But cloned songs displayed a significant lower shimmer than original songs (estimated difference = 0.016, t = 2.801, 95% confidence interval = $1.68*10^{-4}$ ... $3.15*10^{-2}$, p = 0.048), with shimmer in cloned songs higher.

3.2.6.4.Cepstral Peak Prominence (CPP)

For Cepstral Peak Prominence (CPP), cloned versions were observed to exhibit significantly higher values than original versions (estimated difference = 2.307 dB, t = 6.598, 95% confidence interval = 1.334 dB... 3.281 dB, p = 0.002), but not than cover versions (estimated difference = 0.699 dB, t = 1.331, 95% confidence interval = -0.752 dB... 2.149 dB, p = 0.25).

3.2.6.5.Harmonics-to-Noise-Ratio (HNR)

Cloned versions showed a significantly higher value in HNR than original versions (estimated difference = 3.251 dB , t = 3.708, 95% confidence interval = 0.817 dB... 5.685 dB, p = 0.02). But this distinction was not observed between cover and cloned songs (estimated difference = 0.007 dB, t = 0.037, 95% confidence interval = -0.563 dB... 0.579 dB, p = 0.972)

3.2.6.6.Assumptions

No difference was found between the cover and cloned versions, while more distinctions were observed between the original and cover version. It pointed to a serious problem concerning the comparison between human voices singing and cloned voices singing, which was the difficulty to establish a standard for human voices.

Specifically, the cover and cloned versions shared the same underlying vocal source from the Opencpop corpus. But the original versions were performed by different artists, introducing inter-speakers variability which might contribute to the difference between cover and original versions. With the difference within human singing voices, the persuasiveness of CPP and HNR values has been reduced.

4. Limitations

This study did not control the original version group to have its vocal source from the same singer, which might influence the perceived difference between human singing voices and cloned singing voices. In the future study, researchers can delve into this topic using songs from the same artist as the cloned stems.

Another limitation for this research lay in the gender restrictions. In this study, the data only included female voices. Without male voice samples, it may be difficult to conclude that jitter and shimmer have a tendency for cloned songs values to be lower than human songs.

5. Future Studies

In this study, the random effect for song was found to be weak in terms of

voice quality metrics. But all samples belonged to Mandarin Pop songs. Will the weight of random effect change when it comes to other genres? Or will it change when it comes to other languages? These directions are expected to be explored. For example, other song genres always exhibit different vocal effects. RNB (Rhythm and Blues) features with more melisma where pitch manifests more variations on one syllable, which might cause a difference in the random effect for song.

In addition, the random factor id, which identified segments at the same location across versions, showed a high level of randomness. It suggests that segments in some locations may display certain patterns, and these patterns are likely to be concerned with segment types, referring to consonantal onsets and segments including vowels.

It is also suggested that a combined investigation of perceptual studies and acoustic analyses should be conducted in the future to verify whether their results correlate. Although humans and models can both detect synthetic voices, their error patterns differ: humans are more prone to believe deepfakes as humans while machines tend to misidentify real voices as synthetic (Warren et al., 2024). This divergence underscores the necessity of combining human perception and acoustic cues. Whether certain subjective features of speech correspond to objective acoustic cue is still unclear (Wagner et al., 2019).

Moreover, in songs, the lexical pitch is generated to match a musical score. Human singing may incorporate expressive deviations from the musical note,

while cloned vocals might not replicate them perfectly. Voice quality metrics were proved to vary between lower and higher registers (Meireles & Mixdorff, 2020). Whether this difference will be perceived between segments corresponding with musical notes in lower register and higher register is worthy of being researched.

6. Conclusions

The acoustic features jitter and shimmer can be regarded as distinguishing factors for cloned and human singing voice, since cloned songs always exhibit lower jitter and shimmer. These low values may result from over-regularity during the singing voice cloning process. Intensity is another cue that explains why cloned songs are perceived less natural than human singing songs. However, this study cannot draw conclusions on spectral tilt, CPP or HNR.

To conclude, humans are imperfect, so is our voice. If cloned voices want to be more real, they need to embrace human's imperfection.

References

Ardaillon, L. (2017). *Synthesis and Expressive Transformation of Singing Voice.* (Doctoral dissertation, Université Pierre et Marie Curie-Paris VI).

Arik, S., Chen, J., Peng, K., Ping, W., & Zhou, Y. (2018). Neural Voice Cloning with a Few Samples. *Advances in Neural Information Processing Systems*, *31*. https://doi.org/10.48550/arXiv.1802.06006

Boersma, P., & Weenink, D. (2025). Praat: Doing Phonetics by Computer. http://www.praat. org/.

Cooper, E., Huang, W.-C., Tsao, Y., Wang, H.-M., Toda, T., & Yamagishi, J. (2024). A Review on Subjective and Objective Evaluation of Synthetic Speech. *Acoustical Science and Technology*, *45*(4), 161–183. https://doi.org/10.1250/ast.e24.12

Eyben, F., Wöllmer, M., & Schuller, B. (2010). Opensmile: The Munich Versatile and Fast Open-Source Audio Feature Extractor. *Proceedings of the 18th ACM International Conference on Multimedia*, 1459–1462. https://doi.org/10.1145/1873951.1874246

Garellek, M. (2022). Theoretical Achievements of Phonetics in the 21st Century: Phonetics of Voice Quality. *Journal of Phonetics*, *94*, 101155. https://doi.org/10.1016/j.wocn.2022.101155

Hinterleitner, F., Zander, S., Engelbrecht, K.-P., & Möller, S. (2015). On the Use of Automatic Speech Recognizers for the Quality and Intelligibility

Prediction of Synthetic Speech. *Konferenz Elektronische Sprachsignalverarbeitung*, 105–111.

Kühne, K., Fischer, M. H., & Zhou, Y. (2020). The Human Takes It All: Humanlike Synthesized Voices Are Perceived as Less Eerie and More Likable. Evidence From a Subjective Ratings Study. *Frontiers in Neurorobotics*, *14*. https://doi.org/10.3389/fnbot.2020.593732

Layton, S., Andrade, T. D., Olszewski, D., Warren, K., Butler, K., & Traynor, P. (2024). Every Breath You Don't Take: Deepfake Speech Detection Using Breath. *arXiv preprint arXiv:2404.15143*. https://doi.org/10.48550/arXiv.2404.15143

Mai, K. T., Bray, S., Davies, T., & Griffin, L. D. (2023). Warning: Humans Cannot Reliably Detect Speech Deepfakes. *PLOS ONE*, *18*(8), e0285333. https://doi.org/10.1371/journal.pone.0285333

Malisz, Z., Henter, G. E., Botinhao, C. V., Watts, O., Beskow, J., & Gustafson, J. (2019). Modern Speech Synthesis for Phonetic Sciences: A Discussion and an Evaluation. *Proceedings of the 19th International Congress of Phonetic Sciences ICPhS 2019*, 487–491.

Mayo, C., Clark, R. A. J., & King, S. (2011). Listeners' Weighting of Acoustic Cues to Synthetic Speech Naturalness: A Multidimensional Scaling Analysis. *Speech Communication*, *53*(3), 311–326. https://doi.org/10.1016/j.specom.2010.10.003

Meireles, A., & Mixdorff, H. (2020). Voice Quality in Low and High Registers in Two Different Styles of Singing. Proc. *Speech Prosody*, 5.

Müller, N. M., Pizzi, K., & Williams, J. (2022). Human Perception of Audio Deepfakes. *Proceedings of the 1st International Workshop on Deepfake Detection for Audio Multimedia*, 85–91. https://doi.org/10.1145/3552466.3556531

Nishimura, M., Hashimoto, K., Oura, K., Nankaku, Y., & Tokuda, K. (2016). Singing Voice Synthesis Based on Deep Neural Networks. *Interspeech 2016*, 2478–2482. https://doi.org/10.21437/Interspeech.2016-1027

Norrenbrock, C. R., Hinterleitner, F., Heute, U., & Möller, S. (2012). Towards Perceptual Quality Modeling of Synthesized Audiobooks – Blizzard Challenge 2012. *The Blizzard Challenge 2012*, 59–64. https://doi.org/10.21437/Blizzard.2012-11

Norrenbrock, C., Heute, U., Hinterleitner, F., & Möller, S. (2011). Aperiodicity Analysis for Quality Estimation of Text-to-Speech Signals. INTERSPEECH, 2193–2196.

Nussbaum, C., Frühholz, S., & Schweinberger, S. R. (2025). Understanding Voice Naturalness. *Trends in Cognitive Sciences*, *0*(0). https://doi.org/10.1016/j.tics.2025.01.010

R Core Team. (2024). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. https://www.R-project.org/

Rosi, V., Soopramanien, E., & McGettigan, C. (2025). Perception and Social Evaluation of Cloned and Recorded Voices: Effects of Familiarity and Self-relevance. *Computers in Human Behavior: Artificial Humans*, *4*, 100143. https://doi.org/10.1016/j.chbah.2025.100143

Seebauer, F., Kuhlmann, M., Haeb-Umbach, R., & Wagner, P. (2023).Re-examining the Quality Dimensions of Synthetic Speech. In *12th Speech Synthesis Workshop (SSW) 2023*.

Umbert, M., Bonada, J., Goto, M., Nakano, T., & Sundberg, J. (2015). Expression Control in Singing Voice Synthesis: Features, Approaches, Evaluation, and Challenges. *IEEE Signal Processing Magazine*, 32(6), 55–73. https://doi.org/10.1109/MSP.2015.2424572

Wagner, P., Beskow, J., Betz, S., Edlund, J., Gustafson, J., Henter, G. E., Le Maguer, S., Malisz, Z., Székely, É., & Tånnander, C. (2019). Speech Synthesis Evaluation—State-Of-The-Art Assessment and Suggestion for a Novel Research Program. In *Proceedings of the 10th Speech Synthesis Workshop (SSW10),* 2019.

Wang, Y. (2024). The Effectiveness of Innovative Technologies to Manage Vocal Training: The Knowledge of Breathing Physiology and Conscious Control in Singing. *Education and Information Technologies*, 29(6), 7303–7319. https://doi.org/10.1007/s10639-023-12108-6

Wang, Y., Wang, X., Zhu, P., Wu, J., Li, H., Xue, H., Zhang, Y., Xie, L., & Bi, M. (2022). Opencpop: A High-Quality Open Source Chinese Popular Song

Corpus for Singing Voice Synthesis. *arXiv:2201.07429*. https://doi.org/10.48550/arXiv.2201.07429

Warren, K., Olszewski, D., Layton, S., Butler, K., Gates, C., & Traynor, P. (2025). Pitch Imperfect: Detecting Audio Deepfakes Through Acoustic Prosodic Analysis. *arXiv:2502.14726*. https://doi.org/10.48550/arXiv.2502.14726

Warren, K., Tucker, T., Crowder, A., Olszewski, D., Lu, A., Fedele, C., Pasternak, M., Layton, S., Butler, K., Gates, C., & Traynor, P. (2024). 'Better Be Computer or I'm Dumb': A Large-Scale Evaluation of Humans as Audio Deepfake Detectors. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2696–2710. https://doi.org/10.1145/3658644.3670325

Xiong, Z., Wang, W., Yu, J., Lin, Y., & Wang, Z. (2023). *A Comprehensive Survey for Evaluation Methodologies of AI-Generated Music*. *arXiv:2308.13736*. https://doi.org/10.48550/arXiv.2308.13736

Zhang, Q. (2024). *The Individual Perception in Synthetic Speech*. (Master dissertation, Eindhoven University of Technology).

Zhang, Z., Zheng, Y., Li, X., & Lu, L. (2022). WeSinger: Data-augmented Singing Voice Synthesis with Auxiliary Losses. *arXiv:2203.10750*. https://doi.org/10.48550/arXiv.2203.10750

Appendices

Table 1

Songs for Model Training (Chinese)	Songs for Model Training (English)	Songs for Targeted Stimuli (Chinese)	Songs for Targeted Stimuli (English)
《光年之外》	*Light Years Away*	《半句再见》	*Half a Goodbye*
《我怀念的》	*What I Miss*	《走马》	*Galloping Horse*
《日不落》	*Sun Never Sets*	《宁夏》	*Quiet Summer*
《勇气》	*Courage*	《可惜不是你》	*Unfortunately Not You*
《易燃易爆炸》	*Highly Flammable and Explosive*	《阴天》	*Cloudy Day*

Lists for segments extracted for voice quality metrics

a	e	o	i	u	v	er	ia	ie
ua	uo	ve	ai	ei	ao	ou	an	en
in	vn	iao	iou	uan	van	ian	uai	uei
uen	ang	eng	ong	ing	iang	iong	uang	ueng

Praat Script 1: Intensity and Spectral Tilt at sentence level

```
#Step 1: Read TextGrid Files and Wav Files
appendInfoLine: "fileName source song id range_of_intensity(dB) spectralTilt"
folder$ = "Files"
fileNames$# = fileNames$# (folder$ + "/*.TextGrid")
for ifile to size (fileNames$#)
    textgrid = Read from file: folder$ + "/" + fileNames$# [ifile]
    name$ = fileNames$# [ifile] - ".TextGrid"
    if index (name$, "_Clone") >0
        source$ =   "C"
    elsif index (name$, "_Original") >0
        source$ =    "O"
    else
        source$ =    "V"
    endif
    song$ = name$ - "_Clone" - "_Original"
    sound = Read from file: folder$ + "/" + name$ + ".wav"

#Step 2: Measure and Draw Pitch

    selectObject: sound
    intensity = To Intensity: 100.0, 0.0, "yes"
    intensity_max = Get maximum: 0.0, 0.0, "parabolic"
    intensity_min = Get minimum: 0.0, 0.0, "parabolic"

    selectObject: textgrid
    number_of_sentences = Get number of intervals: 1
    for m from 1 to number_of_sentences
        selectObject: textgrid
        sentence_start = Get start point: 1, m
         sentence_end = Get end point: 1, m
        selectObject: sound
        sound2 = Extract part: sentence_start, sentence_end, "rectangular", 1.0, "yes"
        path$ = "E:\Pitch Graphs\"
        name$ = "pitch_" + fileNames$#[ifile] + string$(m) + ".png"
        selectObject: sound2
        pitch2 = To Pitch (raw cross-correlation): 0.0, 50, 600, 15, "yes", 0.03, 0.45, 0.01,
0.35, 0.14
        Select outer viewport: 0, 6, 0, 3
        selectObject: pitch2
        Draw: 0, 0, 0, 500, "yes"
        Save as 300-dpi PNG file: path$ + name$
        Erase all
        removeObject: sound2, pitch2
    endfor

#Step 3: Measure Intensity Range and Spectral Tilt at sentence level

    selectObject: textgrid
    number_of_sentences = Get number of intervals: 1
    for m from 1 to number_of_sentences
```

```
        selectObject: textgrid
        sentence_start = Get start point: 1, m
              sentence_end = Get end point: 1, m
        selectObject: sound
        sound_sentence = Extract part: sentence_start, sentence_end, "rectangular", 1.0,
"yes"
        selectObject: sound_sentence
        intensity_sentence = To Intensity: 100.0, 0.0, "yes"
        intensity_min = Get minimum: 0.0, 0.0, "parabolic"
        intensity0.1 = Get quantile: 0, 0, 0.1
        intensity0.9 = Get quantile: 0, 0, 0.9
        if intensity_min < 0
            intesity_range$ = "NA"
        else
            intensity_range$ = string$ (intensity0.9 - intensity0.1)
            selectObject: sound_sentence
            spectrum_sentence = To Spectrum: "no"
            spectralTilt_sentence = Get band energy difference: 0.0, 500.0, 500.0, 4000.0
            appendInfoLine: fileNames$# [ifile]," ", source$, " ",    song$, " ", song$ + "_" +
string$ (m), " ", intensity_range$," ", spectralTilt_sentence
            removeObject: spectrum_sentence
        endif
        removeObject: sound_sentence, intensity_sentence
    endfor
removeObject: textgrid, sound
endfor
```

Praat Script 2: Intensity and Spectral Tilt at segment level

```
#Step 1: Read TextGrid Files and Wav Files
appendInfoLine: "fileName source song id segments medium_difference(dB)
spectralTilt"
folder$ = "Files"
fileNames$# = fileNames$# (folder$ + "/*.TextGrid")
for ifile to size (fileNames$#)
    textgrid = Read from file: folder$ + "/" + fileNames$# [ifile]
    name$ = fileNames$# [ifile] - ".TextGrid"
    if index (name$, "_Clone") >0
        source$ =    "C"
    elsif index (name$, "_Original") >0
        source$ =      "O"
    else
        source$ =      "V"
    endif
    song$ = name$ - "_Clone" - "_Original"
    sound = Read from file: folder$ + "/" + name$ + ".wav"

#Step 2: Measure Intensity and Spectral Tilts at segment level
    selectObject: sound
```

```
    intensity = To Intensity: 100.0, 0.0, "yes"
    intensity_median = Get quantile: 0, 0, 0.5
    selectObject: textgrid
    number_of_phonemes = Get number of intervals: 6
    for n from 1 to number_of_phonemes
        selectObject: textgrid
        segment_start = Get start point: 6, n
        segment_end = Get end point: 6, n
        segment_label$ = Get label of interval: 6, n
        selectObject:sound
        sound_segment = Extract part: segment_start, segment_end, "rectangular", 1.0, "yes"
        selectObject: sound_segment
        duration_segment = Get total duration
        duration_min_segment = 0.064
        if duration_segment > duration_min_segment
            selectObject: sound_segment
            intensity_segment = To Intensity: 100.0, 0.0, "yes"
            intensity_min = Get minimum: 0.0, 0.0, "parabolic"
            if intensity_min > 0
                intensity_mean = Get mean: 0.0, 0.0, "energy"
                intensity_difference$ = string$ (intensity_mean - intensity_median)
                selectObject: sound_segment
                spectrum_segment = To Spectrum: "no"
                spectralTilt_segment = Get band energy difference: 0.0, 500.0, 500.0,
4000.0
                spectralTilt_segment$ = string$ (spectralTilt_segment)
                removeObject: intensity_segment, spectrum_segment
            else
                intensity_difference$ = "NA"
                spectralTilt_segment$ = "NA"
                removeObject: intensity_segment
            endif
        else
            intensity_difference$ = "NA"
            spectralTilt_segment$ = "NA"
        endif
        appendInfoLine:    fileNames$# [ifile]," ", source$, " ",    song$, " ", song$ + "_" +
string$ (n), " ", segment_label$, " ", intensity_difference$," ", spectralTilt_segment$
        removeObject: sound_segment
    endfor
    removeObject: intensity, textgrid, sound
endfor
```

Praat Script 3: Jitter, Shimmer, CPP, HNR

```
#Step 1: Read TextGrid Files and Wav Files

appendInfoLine: "song source id segments CPP(dB) hnr(dB) jitter shimmer"
folder$ = "Files"
fileNames$# = fileNames$# (folder$ + "/*.TextGrid")
```

```
for ifile to size (fileNames$#)
	textgrid = Read from file: folder$ + "/" + fileNames$# [ifile]
	name$ = fileNames$# [ifile] - ".TextGrid"
	if index (name$, "_Clone") >0
		source$ =    "C"
	elsif index (name$, "_Original") >0
		source$ =      "O"
	else
		source$ =      "V"
	endif
	song$ = name$ - "_Clone" - "_Original"
	sound = Read from file: folder$ + "/" + name$ + ".wav"

#Step 2: Measure CPP and HNR

	iline = 1
	list$# = {"a","e","o","i","u","v",
		..."er","ia","ie","ua","uo","ve","ai","ei","ao","ou","an","en","in","vn",
		..."iao","iou","uan","van","ian","uai","uei","uen",
		..."ang","eng","ong","ing","iang","iong","uang","ueng"}
	selectObject: textgrid
	number_of_phonemes = Get number of intervals: 6
	for n from 1 to number_of_phonemes
		selectObject: textgrid
		phoneme_start = Get start point: 6, n
		phoneme_end = Get end point: 6, n
		phone_label$ = Get label of interval: 6, n
		if index (list$#, phone_label$) > 0
			selectObject: sound
			sound3 = Extract part: phoneme_start, phoneme_end, "rectangular", 1.0, "yes"
			selectObject: sound3
			duration = Get total duration
			duration_min = 6.0/70
			if duration >= duration_min
				pitch3 = To Pitch (raw cross-correlation): 0.0, 100, 600, 15, "yes", 0.03,
0.45, 0.01, 0.35, 0.14
				selectObject: sound3
				cpp = To PowerCepstrogram: 70, 0.002, 5000, 50
				cpp1 = To PowerCepstrum (slice): 0.15
				prominence = Get peak prominence: 70, 333.3, "parabolic", 0.001, 0.05,
"straight", "robust slow"
				selectObject: sound3
				hnr = To Harmonicity (cc): 0.01, 60.0, 0.1, 1.0
				selectObject: hnr
				hnr_mean = Get mean: 0.0, 0.0
				removeObject: cpp, cpp1, hnr
			else
				pitch3 = 0 ; No objects
				hnr_mean = 0.0
				prominence = 0.0
```

```
            endif

            #Step 3: Measure Jitter and Shimmer

            if pitch3 <> 0
                selectObject: sound3, pitch3
                point = To PointProcess (cc)
                jitter = Get jitter (local): 0, 0, 0.0001, 0.02, 1.3
                selectObject: sound3, point
                shimmer = Get shimmer (local): 0, 0, 0.0001, 0.02, 1.3, 1.6
                appendInfoLine: song$, " ", source$ , " ", song$ + "_" + string$ (n), " ",
phone_label$, " ", prominence, " ", hnr_mean, " ", jitter, " ", shimmer
                removeObject: sound3, pitch3, point
                iline = iline + 1
            endif
        endif
    endfor
endfor
removeObject: sound, textgrid
```

R Markdown File

Read Tables

```{r}
library(tidyverse)
library(lmerTest)
library(lme4)
library(dplyr)

data_intensity_sentence_all <- read.table("data/intensity sentence.txt", header = TRUE,
    stringsAsFactors = TRUE)
data_intensity_sentence_all

data_intensity_segment_all <- read.table("data/intensity segment.txt", header = TRUE,
    stringsAsFactors = TRUE)
data_intensity_segment_all <-
subset(data_intensity_segment_all, !data_intensity_segment_all$segments%in%
(c("SP","AP")))
data_intensity_segment_all

data_phoneme_all <- read.table("data/Phoneme.txt", header = TRUE, stringsAsFactors =
TRUE)
data_phoneme_all$jitter <- as.numeric(as.character(data_phoneme_all$jitter))
data_phoneme_all$shimmer <- as.numeric(as.character(data_phoneme_all$shimmer))
data_phoneme_all
```

Set Contrasts

```{r}
```

```
CODE_VOCAL <- +0.5
CODE_ORIGINAL <- -0.5
CODE_CLONE <- 0
CODE_CLONE1 <- -0.5
CODE_CLONE2 <- +0.5

CODE_AVERAGE_VOCAL <- -1/3
CODE_AVERAGE_CLONE <- +2/3
CODE_AVERAGE_ORIGINAL <- -1/3

contrasts <- cbind(
                        c(CODE_CLONE, CODE_ORIGINAL, CODE_VOCAL),

c(CODE_AVERAGE_CLONE,CODE_AVERAGE_ORIGINAL,CODE_AVERAGE_VOCA
L)
                        )

colnames(contrasts) <- cbind(c("-O+V"),
                                    c("-VO+C"))

contrast1 <- cbind(c(CODE_CLONE1, CODE_VOCAL))
colnames(contrast1) <- c("-C+V")
contrast2 <- cbind(c(CODE_CLONE2, CODE_ORIGINAL))
colnames(contrast2) <- c("-O+C")
```

Intensity and Spectral Tilts (ternary) at the Segment Level

```{r}
contrasts(data_intensity_segment_all$source) <- contrasts
contrasts(data_intensity_segment_all$source)

model_intensity_segment <- lmerTest::lmer(medium_difference.dB. ~ source + (source| song)
+ (1 | id),
    data = data_intensity_segment_all, REML = TRUE)
fixedEffects_is <- lme4::fixef(model_intensity_segment)
cbind(names(fixedEffects_is),
      lmerTest::contest(model_intensity_segment,
                        diag(length(fixedEffects_is)), joint = FALSE))
summary(model_intensity_segment)

model_spectralTilt_segment <- lmerTest::lmer(spectralTilt ~ source + (source| song) + (1| id),
    data = data_intensity_segment_all, REML = TRUE)
fixedEffects_ss <- lme4::fixef(model_spectralTilt_segment)
cbind(names(fixedEffects_ss),
      lmerTest::contest(model_spectralTilt_segment,
                        diag(length(fixedEffects_ss)), joint = FALSE))
summary(model_spectralTilt_segment)
```

Intensity and spectral tilt (between vocal and cloned versions) at the Segment Level

```{r}
data_intensity_segment_all1 <- data_intensity_segment_all %>%
  filter(source %in% c("V", "C"))
data_intensity_segment_all1$source <- droplevels(data_intensity_segment_all1$source)

contrasts(data_intensity_segment_all1$source) <- contrast1
contrasts(data_intensity_segment_all1$source)

model_intensity_segment1 <- lmerTest::lmer(medium_difference.dB. ~ source + (source| song) + (1 | id),
    data = data_intensity_segment_all1, REML = TRUE)
fixedEffects_is1 <- lme4::fixef(model_intensity_segment1)
cbind(names(fixedEffects_is1),
      lmerTest::contest(model_intensity_segment1,
                        diag(length(fixedEffects_is1)), joint = FALSE))
summary(model_intensity_segment1)

model_spectralTilt_segment1 <- lmerTest::lmer(spectralTilt ~ source + (source| song) + (1| id), data = data_intensity_segment_all1, REML = TRUE)
fixedEffects_ss1 <- lme4::fixef(model_spectralTilt_segment1)
cbind(names(fixedEffects_ss1),
      lmerTest::contest(model_spectralTilt_segment1,
                        diag(length(fixedEffects_ss1)), joint = FALSE))
summary(model_spectralTilt_segment1)
```

Intensity and spectral tilt (between original and cloned versions) at the Segment Level
```{r}
data_intensity_segment_all2 <- data_intensity_segment_all %>%
  filter(source %in% c("C", "O"))
data_intensity_segment_all2$source <- droplevels(data_intensity_segment_all2$source)

contrasts(data_intensity_segment_all2$source) <- contrast2
contrasts(data_intensity_segment_all2$source)

model_intensity_segment2 <- lmerTest::lmer(medium_difference.dB. ~ source + (source| song) + (1 | id) ,
    data = data_intensity_segment_all2, REML = TRUE)
fixedEffects_is2 <- lme4::fixef(model_intensity_segment2)
cbind(names(fixedEffects_is2),
      lmerTest::contest(model_intensity_segment2,
                        diag(length(fixedEffects_is2)), joint = FALSE))
summary(model_intensity_segment2)

model_spectralTilt_segment2 <- lmerTest::lmer(spectralTilt ~ source + (source| song) + (1| id),
    data = data_intensity_segment_all2, REML = TRUE)
fixedEffects_ss2 <- lme4::fixef(model_spectralTilt_segment2)
cbind(names(fixedEffects_ss2),
      lmerTest::contest(model_spectralTilt_segment2,
```

```
                                diag(length(fixedEffects_ss2)), joint = FALSE))
summary(model_spectralTilt_segment2)
```

Intensity and Spectral Tilts (Ternary) at the Sentence Level
```{r}
contrasts(data_intensity_sentence_all$source) <- contrasts
contrasts(data_intensity_sentence_all$source)

model_intensity_sentence <- lmerTest::lmer(range_of_intensity.dB. ~ source + (source| song)
+ (1 | id),
      data = data_intensity_sentence_all, REML = TRUE)
fixedEffects_i_s <- lme4::fixef(model_intensity_sentence)
cbind(names(fixedEffects_i_s),
          lmerTest::contest(model_intensity_sentence,
                                diag(length(fixedEffects_i_s)), joint = FALSE))
summary(model_intensity_sentence)

model_spectralTilt_sentence <- lmerTest::lmer(spectralTilt ~ source + (source| song) + (1|
id),
      data = data_intensity_sentence_all, REML = TRUE)
fixedEffects_s_s <- lme4::fixef(model_spectralTilt_sentence)
cbind(names(fixedEffects_s_s),
          lmerTest::contest(model_spectralTilt_sentence,
                                diag(length(fixedEffects_s_s)), joint = FALSE))
summary(model_spectralTilt_sentence)
```

Intensity and spectral tilt (between cover and cloned versions) at the Sentence Level
```{r}
data_intensity_sentence_all1 <- data_intensity_sentence_all %>%
   filter(source %in% c("C", "V"))
data_intensity_sentence_all1$source <- droplevels(data_intensity_sentence_all1$source)

contrasts(data_intensity_sentence_all1$source) <- contrast1
contrasts(data_intensity_sentence_all1$source)

model_intensity_sentence1 <- lmerTest::lmer(range_of_intensity.dB. ~ source +     (source|
song) + (1 | id),
      data = data_intensity_sentence_all1, REML = TRUE)
fixedEffects_i_s1 <- lme4::fixef(model_intensity_sentence1)
cbind(names(fixedEffects_i_s1),
          lmerTest::contest(model_intensity_sentence1,
                                diag(length(fixedEffects_i_s1)), joint = FALSE))
summary(model_intensity_sentence1)

model_spectralTilt_sentence1 <- lmerTest::lmer(spectralTilt ~ source + (source| song) + (1|
id),
      data = data_intensity_sentence_all1, REML = TRUE)
fixedEffects_s_s1 <- lme4::fixef(model_spectralTilt_sentence1)
```

```
cbind(names(fixedEffects_s_s1),
        lmerTest::contest(model_spectralTilt_sentence1,
                          diag(length(fixedEffects_s_s1)), joint = FALSE))
summary(model_spectralTilt_sentence1)
```

Intensity and spectral tilt (between original and cloned versions) at the Sentence Level
```{r}
data_intensity_sentence_all2 <- data_intensity_sentence_all %>%
  filter(source %in% c("C", "O"))
data_intensity_sentence_all2$source <- droplevels(data_intensity_sentence_all2$source)

contrasts(data_intensity_sentence_all2$source) <- contrast2
contrasts(data_intensity_sentence_all2$source)

model_intensity_sentence2 <- lmerTest::lmer(range_of_intensity.dB. ~ source +  (source| song) + (1 | id),
    data = data_intensity_sentence_all2, REML = TRUE)
fixedEffects_i_s2 <- lme4::fixef(model_intensity_sentence2)
cbind(names(fixedEffects_i_s2),
        lmerTest::contest(model_intensity_sentence2,
                          diag(length(fixedEffects_i_s2)), joint = FALSE))
summary(model_intensity_sentence2)

model_spectralTilt_sentence2 <- lmerTest::lmer(spectralTilt ~ source + (source| song) + (1| id),
    data = data_intensity_sentence_all2, REML = TRUE)
fixedEffects_s_s2 <- lme4::fixef(model_spectralTilt_sentence2)
cbind(names(fixedEffects_s_s2),
        lmerTest::contest(model_spectralTilt_sentence2,
                          diag(length(fixedEffects_s_s2)), joint = FALSE))
summary(model_spectralTilt_sentence2)
```

jitter, shimmer, cpp, lmer (ternary)
```{r}
contrasts(data_phoneme_all$source) <- contrasts
contrasts(data_phoneme_all$source)

model_jitter <- lmerTest::lmer(jitter ~ source + (source| song) + (1| id),
    data = data_phoneme_all, REML = TRUE)
fixedEffects_jitter <- lme4::fixef(model_jitter)
cbind(names(fixedEffects_jitter),
        lmerTest::contest(model_jitter,
                          diag(length(fixedEffects_jitter)), joint = FALSE))
summary(model_jitter)

model_shimmer <- lmerTest::lmer(shimmer ~ source + (source| song) + (1| id),
    data = data_phoneme_all, REML = TRUE)
fixedEffects_shimmer <- lme4::fixef(model_shimmer)
```

```
cbind(names(fixedEffects_shimmer),
        lmerTest::contest(model_shimmer,
                          diag(length(fixedEffects_shimmer)), joint = FALSE))
summary(model_shimmer)

model_cpp <- lmerTest::lmer(CPP.dB. ~ source + (source| song) + (1| id),
     data = data_phoneme_all, REML = TRUE)
fixedEffects_cpp <- lme4::fixef(model_cpp)
cbind(names(fixedEffects_cpp),
        lmerTest::contest(model_cpp,
                          diag(length(fixedEffects_cpp)), joint = FALSE))
summary(model_cpp)

model_hnr <- lmerTest::lmer(hnr.dB.    ~ source + (source| song) + (1| id),
     data = data_phoneme_all, REML = TRUE)
fixedEffects_hnr <- lme4::fixef(model_hnr)
cbind(names(fixedEffects_hnr),
        lmerTest::contest(model_hnr,
                             diag(length(fixedEffects_hnr)),joint = FALSE))
summary(model_hnr)
```

jitter, shimmer, cpp, lmer (between vocal and cloned versions)

```{r}
data_phoneme_all_filtered1 <- data_phoneme_all %>%
   filter(source %in% c("V", "C"))
data_phoneme_all_filtered1$source <- droplevels(data_phoneme_all_filtered1$source)

contrasts(data_phoneme_all_filtered1$source) <- contrast1
contrasts(data_phoneme_all_filtered1$source)

model_jitter1 <- lmerTest::lmer(jitter ~ source + (source| song) + (1| id),
     data = data_phoneme_all_filtered1, REML = TRUE)
fixedEffects_jitter1 <- lme4::fixef(model_jitter1)
cbind(names(fixedEffects_jitter1),
        lmerTest::contest(model_jitter1,
                          diag(length(fixedEffects_jitter1)), joint = FALSE))
summary(model_jitter1)

model_shimmer1 <- lmerTest::lmer(shimmer ~ source + (source| song) + (1| id),
     data = data_phoneme_all_filtered1, REML = TRUE)
fixedEffects_shimmer1 <- lme4::fixef(model_shimmer1)
cbind(names(fixedEffects_shimmer1),
        lmerTest::contest(model_shimmer1,
                          diag(length(fixedEffects_shimmer1)), joint = FALSE))
summary(model_shimmer1)

model_cpp1 <- lmerTest::lmer(CPP.dB. ~ source + (source| song) + (1| id),
     data = data_phoneme_all_filtered1, REML = TRUE)
fixedEffects_cpp1 <- lme4::fixef(model_cpp1)
```

```
cbind(names(fixedEffects_cpp1),
        lmerTest::contest(model_cpp1,
                          diag(length(fixedEffects_cpp1)), joint = FALSE))
summary(model_cpp1)

model_hnr1 <- lmerTest::lmer(hnr.dB.    ~ source + (source| song) + (1| id),
     data = data_phoneme_all_filtered1, REML = TRUE)
fixedEffects_hnr1 <- lme4::fixef(model_hnr1)
cbind(names(fixedEffects_hnr1),
        lmerTest::contest(model_hnr1,
                          diag(length(fixedEffects_hnr1)), joint = FALSE))
summary(model_hnr1)
```

jitter, shimmer, cpp, lmer (between original and cloned versions)

```{r}
data_phoneme_all_filtered2 <- data_phoneme_all %>%
   filter(source %in% c("O", "C"))
data_phoneme_all_filtered2$source <- droplevels(data_phoneme_all_filtered2$source)

contrasts(data_phoneme_all_filtered2$source) <- contrast2
contrasts(data_phoneme_all_filtered2$source)

model_jitter2 <- lmerTest::lmer(jitter ~ source + (source| song) + (1| id),
     data = data_phoneme_all_filtered2, REML = TRUE)
fixedEffects_jitter2 <- lme4::fixef(model_jitter2)
cbind(names(fixedEffects_jitter2),
        lmerTest::contest(model_jitter2,
                          diag(length(fixedEffects_jitter2)), joint = FALSE))
summary(model_jitter2)

model_shimmer2 <- lmerTest::lmer(shimmer ~ source + (source| song) + (1| id),
     data = data_phoneme_all_filtered2, REML = TRUE)
fixedEffects_shimmer2 <- lme4::fixef(model_shimmer2)
cbind(names(fixedEffects_shimmer2),
        lmerTest::contest(model_shimmer2,
                          diag(length(fixedEffects_shimmer2)), joint = FALSE))
summary(model_shimmer2)

model_cpp2 <- lmerTest::lmer(CPP.dB. ~ source + (source| song) + (1| id),
     data = data_phoneme_all_filtered2, REML = TRUE)
fixedEffects_cpp2 <- lme4::fixef(model_cpp2)
cbind(names(fixedEffects_cpp2),
        lmerTest::contest(model_cpp2,
                          diag(length(fixedEffects_cpp2)), joint = FALSE))
summary(model_cpp2)

model_hnr2 <- lmerTest::lmer(hnr.dB.    ~ source + (source| song) + (1| id),
     data = data_phoneme_all_filtered2, REML = TRUE)
fixedEffects_hnr2 <- lme4::fixef(model_hnr2)
```

```
cbind(names(fixedEffects_hnr2),
        lmerTest::contest(model_hnr2,
                          diag(length(fixedEffects_hnr2)), joint = FALSE))
summary(model_hnr2)
```