Let's Jam: A Machine Learning Framework For Sample Pairing and Filtration Using Melodic and Tonal Movement Classification

Hritik Patel and Abhinav Damera

ABSTRACT

The search engines of music sampling services are often inadequate when addressing samples that tonally and sonically pair well with each other. Using search filters and other functions within these services, one is able to find samples that are within the same key as the desired sample, and with the same sonic qualities. Current technology offers narrow sample selections, or focuses only on the instrumentation of the sample. This study encompasses a clustering framework and accompanying machine-learning prototype, developed to mirror cognitive musical processing, termed HAPD. HAPD is a sample-sorting system that categorizes samples by tonal and chromatic qualities, such as the chord structure and tonal mean. This method delivers further classification advantages and allows sample libraries to provide sample pairings that are not sonically identical, but still fall under the same movement. The developed prototype, utilizing unsupervised clustering, chroma feature extraction, and cosine similarity functions demonstrates the feasibility of this classification design which will be expanded upon through human evaluation. Future evaluation through human preference and comparative analysis to current search engines is proposed to assist the full implementation of the framework. These trials would also serve as an evaluation of the algorithm in real-life workflow. Execution of this framework would lead to more

diversity and efficiency with sample pairing and selection, while also allowing the user to exercise more tonal specificity when searching, improving workflow.

INTRODUCTION

Convenience is the most important element of music production. Being the most efficient in all aspects aids producers in making more songs in a shorter amount of time. One aspect often overlooked is the places where producers receive their samples. The broad scope and subjectivity of music make the categorization of samples in a digital sample library insubstantial and often confusing. Finding the "right" sample is an issue many producers face. A paper from the Los Angeles Film School details how sampling other copyrighted music depends on, "...how you manipulate the sounds and how creatively you transform them into something new." While the samples in a library are royalty-free, it provides insight into how, "finding the right source material is always a challenge,".[1] Good samples maximize utility and align with the producer's vision. Whilst many sampling services are able to master the use of instrumentation agreement (having features that specify instrument categories), and key agreement (having systems that use the tonal scale the sample falls under in its classification process), an area they have yet to address is the tonal specificities of each individual sample. The movement of the sample: the way the tones are structured (chord progressions) and the general harmonic structure is an important consideration that is overlooked by sampling services.[2] Features like these are not dependent on the scale it is in, nor is it dependent on the instrumentation it has. Two samples that are in the same key are not guaranteed

to sound good together, even with manipulation.[3] Current sampling services fail to match samples through movement similarity, and instead rely on instrumentation.

Recently, several advancements have been made to the field of music regarding the interpretation of music. Advances in machine-learning models for genre classification using vocal, harmonic, and melodic features have improved in-depth music analysis.[4] Sampling service Splice's use of a machine-learning model has released features such as “similar sounds” which offer a newer way to filter samples.[5] However, the algorithm favors instrumental similarity over melody, resulting in mismatched or redundant samples. Artists want to find samples that are variations of the sounds they already have, while still maintaining a similar movement to promote chordal agreement.[6] If a sample is identical to what the producer already has, it does not help because it offers no new creative value. If a sample differs in movement, instrumentation similarity does little to help, as it will sound discordant and will have dissonance.

When comparing the instrumental or melodic qualities of a sample, and determining which factor to consider more in classification, the current body of research offers supporting perspectives. Firstly, in the eyes of an algorithm, it is possible to separate these qualities into two distinguishable and manipulable factors. One study conducted by the International Society for Music Information Retrieval details a model that successfully separates pitch and timbre of inputted samples into independent attributes[7]. Thus, these classification methods can work in tandem, but are still singular organization algorithms that achieve differing results. Moreover, the usage of tonal

features in classification has shown to have a higher degree of accuracy in other uses like identification. For example, in a research project conducted at Columbia University, two models were compared in their ability to identify the artist of a song. One was trained using Mel-Frequency Cepstral Coefficients (MFCCs), and compared to a model that was trained in identifying MFCCs and chroma vectors, and was found to have a higher degree of accuracy in identification when compared to just timbral analysis alone.[13] This perspective suggests that melodic features serve as an addition to analysis that can aid in comprehension when compared to just instrumentation. Both seem to support the usage of melodic movement as opposed to instruments as previous models have done, but both only function to separate and identify. These suggest the utility of chroma-based features, but do not implement their usage in classification. This gap is further identified in scholarly works such as a submission to the SEKE conference in 2022, with them identifying how “frequency domain-based audio features may not represent the music content properly, limiting the recommendation algorithm’s performance”.[9] Their project proposes a merging of chord movement and audience retention, illustrating the importance of pitch in music classification and recommendation, but still exemplifying the present gap in direct feature comparison and parity. Current developments, while advocating for the novelty and possible utility of prioritizing tone over instrumentation, have yet to implement this differentiable factor in organization, especially in sample services. The gap this study attempts to occupy is the need for a machine-learning system that uses chroma

features alone for categorization, while also creating a model that further corroborates the utility proposed by the previous studies.

ALGORITHM DESIGN & METHOD

We have developed a design for a method to categorize samples, hereon referred to as Harmonic-Aware Pairing Detection (HAPD). HAPD is a model based around a pipeline process that analyzes a sample by the harmonic content to compare and cluster samples in orders of similarity. Such allows for a comparison of the melodic content of the sample, not just the instrumentation or “sound texture”, like sampling services do currently. This process is derived from modern machine-learning pipelines and neurological interpretation processes of musical stimuli. This attempts to circumvent preexisting issues of sample libraries providing incompatible samples by utilizing an alternative algorithm, which is meant to produce more melodically compatible results to aid in production workflow.

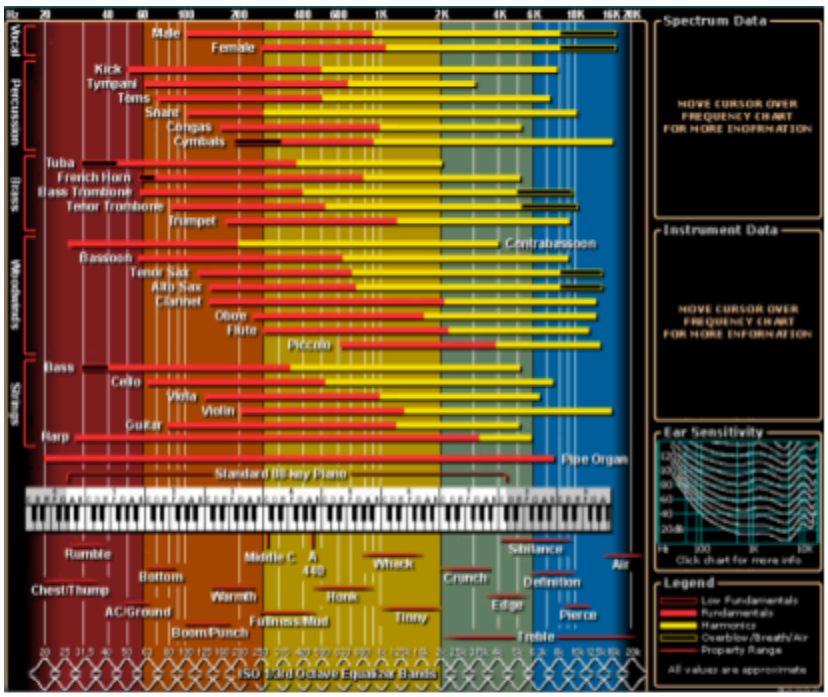

Figure 1. Equalizer frequency table showcasing general frequency ranges for instruments and percussion[10]

Firstly, the design separates the sample into its various components(see Figure 1). These stems are then analyzed for frequencies that “stick out” and are the most prominent. From there, the most commonly occurring notes have chord vectors assigned to it through a vast database of possible chords ranging from multiple scales (major, minor, harmonic, melodic, blues, etc). From there, a chord progression and their degrees of similarity can be extracted from the sample, which can be used for better sample pairing.

The model categorizes musical samples based on harmonic movement and functional chord structure. The core idea is to extract harmonic information from audio samples, map that information to functional chord labels (I, ii, IV, V, etc.), and group samples with similar tonal movement. Our algorithm contains four main stages: First, audio preprocessing and feature extraction; Next, chord and key estimation; Third, functional chord encoding; Finally, machine-learning-based classification and pairing.

The model operates with audio samples in standard format (mono, fixed sampling rate). This pipeline then employs audio (spectrogram) patterns to isolate frequencies in the sound that are occupied by a sound that appears to be an instrument, drums, or vocal.[11] By mapping out frequencies onto a spectrogram, the model is able to visualize these inputs graphically. This is achieved through unsupervised inputs of the parameters and the assignment of malleable decision boundaries to sound inputs for the model to separate these inputs. For example, vocal voicing often has varied frequency ranges (75-400 Hertz-Bass/Tenor, 165-1100 Hz-Alto/Soprano).[12] Additionally,

instruments have general frequency classifications that can be used to feed a predictive model what frequency range(s) the model should separate the audio into.

The system then extracts chroma-based harmonic features, which represent the energy of the twelve pitch classes independent of octave. Using frequencies from the spectrogram, the model would use a Mel-spectrogram-like classification system to “pick out” the most prominent frequencies during each time interval, which are then isolated to form a prediction as to what the notes of the audio input most likely are. Chromagram extraction was used as opposed to traditional waveform analysis as this allows us to focus on the harmonic content of the samples rather than the instrumentation[8].

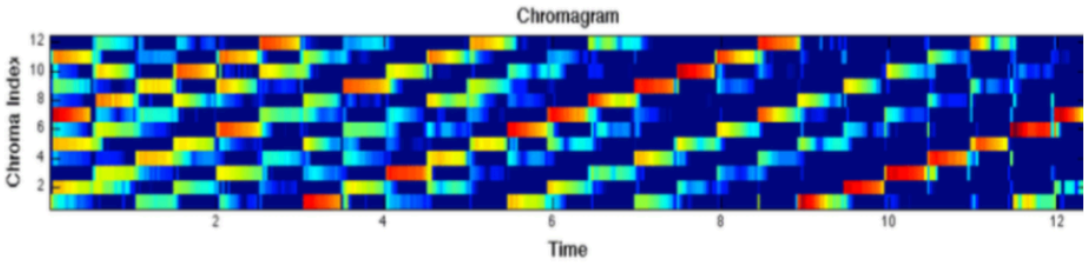

Figure 2. Chromagram of an ascending scale. As shown, each note is able to be translated into a chroma vector, and indexed to a chroma value over a function of time[15].

The model computes the short-time Fourier transform (STFT) of the audio signal. In doing so, the signal is isolated into its component frequencies across many independent time-intervals, which the model then isolates frequencies that appear above a certain relative threshold. The frequencies are then inputted into folding frequency bins, using a 12-dimensional chroma vector:

$c(t) = [c_1(t), c_2(t), \dots, c_{12}(t)]$. [14] These chroma vectors are then mapped onto a chromagram over a function of time (see Figure 2), where the Y axis is the note values of an octave in ascending order (each chroma vector corresponds to a note along that octave).

Using chroma features, the system estimates a local chord sequence and a global key for each sample. Chords and their corresponding notes are fed into the algorithm (for example, if the sample was predetermined to be in G minor, the selection it would choose from would have “i-Chord=G, Bb, D; iidim-Chord=A, C, Eb” and so on). Then, each sample is converted to represent a sequence of functional chords in chord progression format: $S = [f_1 - f_2 - \dots - f_n]$.[16] This allows the sample to be compatible with machine learning because the chord sequences would be turned into computable values. Next, adjacent identical labels are merged together (e.g. I-I-IV becomes I-IV). This would allow us to produce a simplified chord progression. The global key is estimated by analyzing pitch-class and chord frequency over the full sample and selecting the key that maximizes tonal consistency. This step enables later abstraction from absolute pitch to functional harmony.

Much of the neural network development as well as organization of algorithms in the pipeline are modeled after processes the human brain undergoes when leading a melody “from ear”. Neurological functions were used as a functional blueprint to further humanize the process, to deliver the most human-like output. If HAPD follows a closely similar interpretation sequence as the brain does, it can make selections much more accurately and effectively with informed and intentional processing. In a Harvard study,

it was found that the brain utilizes filtration methods of sound in order to classify them as "familiar" and "unfamiliar", when listening to the surrounding environment[17]. The temporal lobe is responsible for this filtration, which can often lead to neural visualization following a process known as a "what pathway"[18]. This is why sound can often be described as having "textures". Similarly, a filtration system was used first in HAPD, not only to separate stems, but to also account for variety in sound texture, similar to how the human brain utilizes the same process.

Additionally, acknowledgement of atonality was considered, similar to how the brain reacts to it. Atonality is perceived as a more unpredictable stimulus in the brain, and as a result, "clashing" chords tend to sound atonal, as they lack a central key and often sound astringent. Hence why two sounds in the same key are not guaranteed to sound cohesive, and why chord classification is necessary. A study in Frontiers utilized an MIR toolbox to model matching pitch-class profiles (similar to our quantification of chromograms) and pulse clarity (how easily a key or general tone can be recognized) as they evaluated varying chords in order of atonality.[19] This modeling system closely followed the process the brain uses when discerning how synonymous two sounds sound together. Basic music theory, which is what our chord classification system is based off of, follows tones that surpass a certain value of pulse clarity, and can be accurately fit into the 12 standardized pitch-class profiles. Moreover, the use of concepts like anticipation, call-and-response, and reward systems in the brain also allow for music theory principles to aid in our classification process.

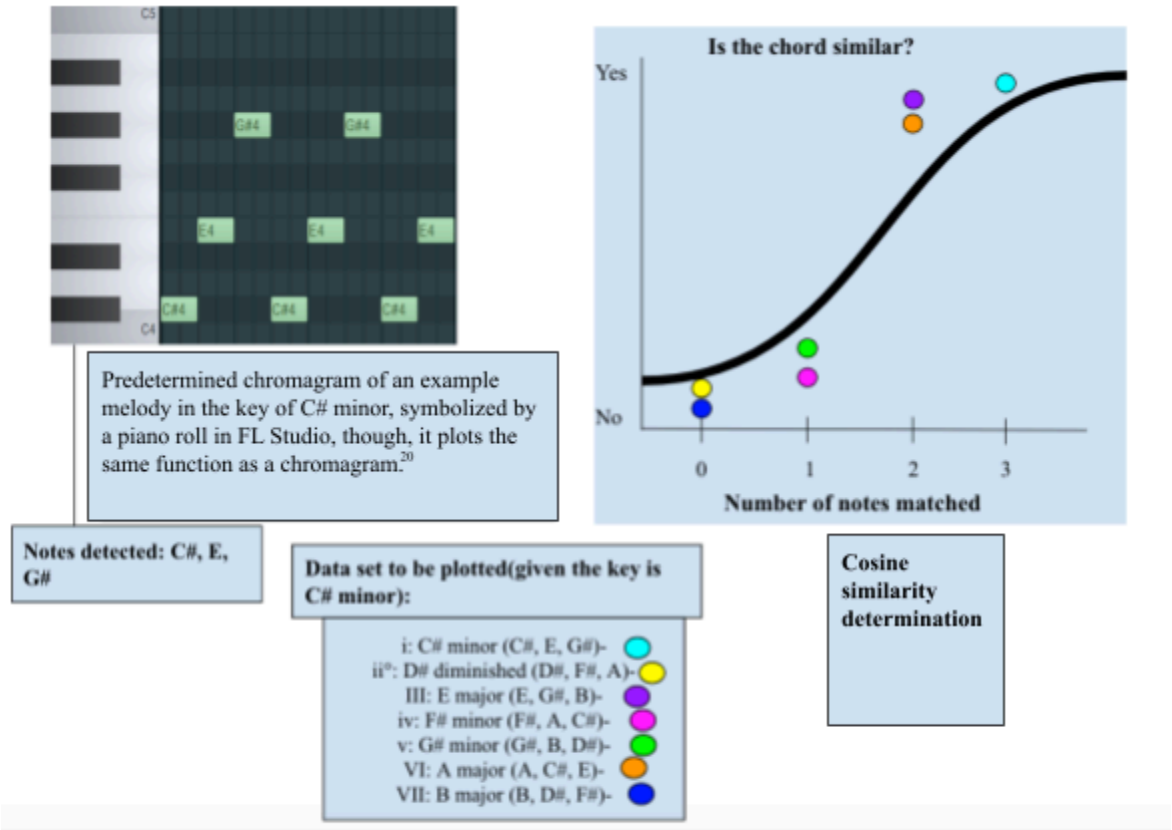

Figure 3. Visualization of cosine similarity process.

HAPD then determines similarity using the chroma vectors to predefined chord templates using standard cosine similarity of the various melody regions: $sim(c,\ t)\ =\ \frac{c*t}{||c||||t||}$. Cosine similarity measures the angular similarity between two vectors in a multidimensional space, producing values closer to one when two samples share similar harmonic features. Samples that are more harmonically different will lead to a lower cosine similarity value.[21] Unsupervised learning is used over supervised learning to allow the system to make natural groupings of the samples based on their tonal structure[22]. We look for relative similarity so producers can combine samples that

sound good, rather than enforcing strict classification (see Figure 3). Using unsupervised learning (clustering), we are making fewer assumptions and there would be less risk of bias from weak labels leading to precise harmonic pairings.[23] This mechanism prioritizes tonal coherence while encouraging diversity in sound selection. By utilizing this sample filtration system, search engines in sample libraries can be upgraded to better pair samples internally in their library, contrary to timbral similarity classification used by current libraries.

RESULTS & PROOF OF CONCEPT

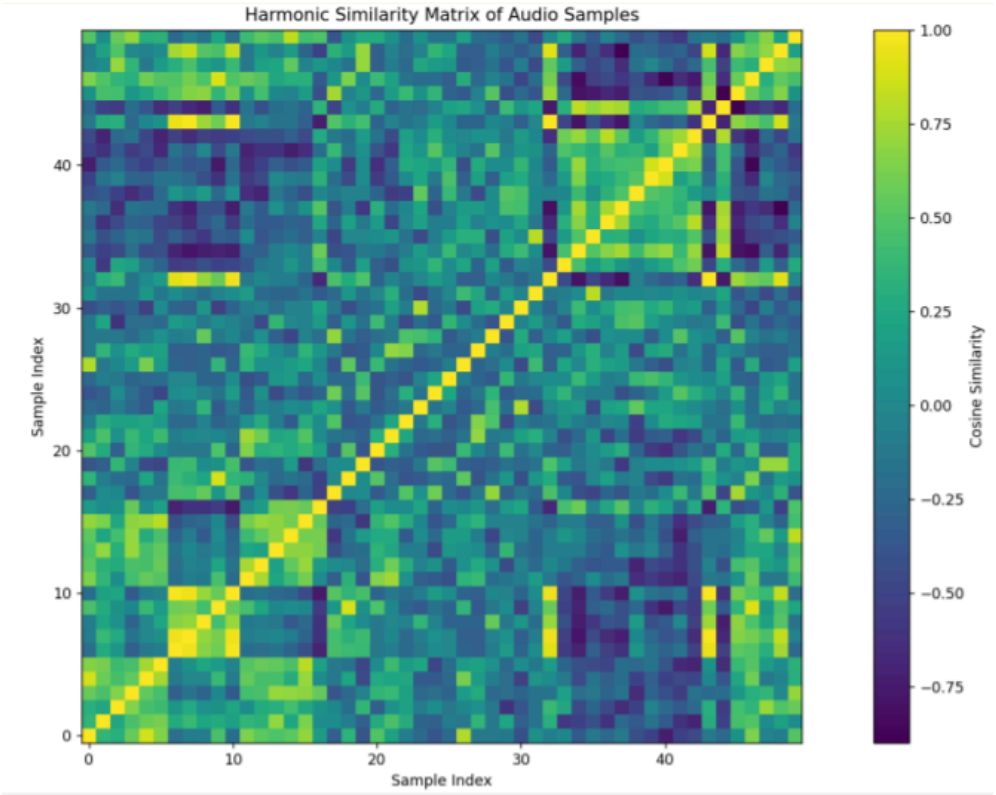

Figure 4. Graph of samples and comparative cosine similarity. The x-axis and y-axis represent the samples included in the dataset, with each square comparing two samples to each other in order of similarity. The diagonal yellow line from the bottom left corner to the top right corner represents self-similarity because each sample is perfectly similar to itself.

A primitive version of our machine learning blueprint was developed using Python in Visual Studio Code as the primary programming environment. This prototype is a simplified implementation of our design, utilizing a clustering framework to analyze vectors, while still utilizing chroma feature analysis and cosine similarity features. Python allows for the simplification of developing harmonic analysis systems. Many libraries were used to create the model: Librosa was used for audio processing and chromagram extraction, Scikit-learn was used for clustering and similarity analysis, NumPy was to compute numerical operations, and Matplotlib generated a visualization. These tools allowed us to develop a model which performs detailed computational data analysis to process audio samples and extract harmonic information as well as to compare the tonal relationships between the samples.

Librosa extracted a chromagram from each sample using a Short-Time Fourier Transform-based chroma analysis process after standardizing them. The model then averages the chroma values across time intervals to make a twelve dimensional feature vector for every sample. Each of the twelve positions of the vector represent one pitch class and the numerical values show how prominent each pitch class is within each sample. The higher the value is, the more harmonic presence there is. The vectors are like fingerprints for each audio sample as the numerical values summarize the tonal structure. Converting the samples from audio to numerical values allows the model to figure out similarity between samples mathematically.

After feature extraction, the vectors are standardized using feature scaling to prevent any individual pitch class from disproportionately affecting clustering. The machine learning model uses K-means clustering (an unsupervised machine learning algorithm) to group the samples based on how similar they are to each other. Clustering demonstrates how the model can independently organize the samples based on the harmonic structures into similar groups.[24] After applying cosine similarity, the model uses the data to make a heatmap of the harmonic similarity of all the samples. The heatmap demonstrates a concise visual relationship between the samples and provides a visual determinant as to which audio samples are more harmonically similar than others.

After the prototype of HAPD was developed, it was then provided with 50 test samples to process and organize into a concise format. Each of the 50 samples were chosen in ascending order of complexity. The first 10 are single-voice sine waves playing simple, stepwise melodies with large, discernable intervals (e.g., C5-A5). With each succeeding 10 samples, there are more notes added in closer intervals (e.g. C5-D5-E5-B5). Additionally, the waveform is altered to a more complex state, utilizing saw waves with multiple voices. Due to the sample number corresponding to complexity, clear patterns can be seen in the resulting visualization (see Figure 4). The clusters and heatmap generated from the machine learning model demonstrate that the model has the ability to find harmonic relationships between the samples. Brighter regions on the heatmap showed samples that were similar in harmonic structure and these samples were often grouped in the same clusters. This supports our original

hypothesis that it is possible to use a machine learning model to pair samples based on tonal structure rather than purely instrumentation.

It is important to note that this model does not currently utilize chord-state tracking and stem separation features, as these will only be effectively implemented during human trials. The model takes the tonal mean of the sample, and assigns it to a vector, which reserves chord separation and stem element separation to a separate function. These are reserved for calibration as they require data from psychoacoustic and creative preferences in human sample analysis. Whilst these steps are important, the current prototype delivers a majority of the tasks outlined in the design for HAPD, while also allowing for areas of further specification that can be expanded upon during trial. Even in its current state, its usage of global harmonic distribution profiles through vector-based similarity differentiates it from current timbral-based analysis. Even without the trial-specific elements, HAPD demonstrates sophisticated pairing and ordering ability that will only be further specified with further implementation. This baseline serves as a baseline prototype for future expansion.

There are several risks associated with developing this machine learning model. One potential risk is inaccuracies with chord detection. Collating multiple chord predictions, removing low-confidence detections and initially training the model with a wide range of samples can help reduce this. Another risk is the subjectivity of harmonic similarity. The system offers multiple outputs and prioritizes expert-validated samples because pairings are context dependent. Additionally, inaccurately identifying chord changes is a risk. Initially, it is addressed with cosine similarity, HAPD can also factor in

time signature information to enable consistent chord separation (e.g., four-bar progressions in 4/4 time).

DISCUSSION & EVALUATION METHODOLOGY

Many of the components of the model are uses of organization systems that currently function in other tasks. For example, algorithms like the STFT and methodology such as linear regression have been present in previous models[25]. However, the novelty of the algorithm is present in its combination of advancements made to the present algorithms, as well as its use of visualization as its main analysis. The use of categorization by order of importance (in this situation, similarity) is useful in search engines for sample libraries, as the organization structure for search engine results can be ordered based on said similarity. The translation of melodic analysis to search engine data is unconventional; however, using advanced algorithms working in a multi-step process can advance classification systems in a way that can integrate with existing search engine filtration. The plausibility of HAPD is further corroborated by the fact that the prototype has analysis similar to those of previous models, only to a differing degree. However, more can be done to evaluate the model in terms of its current effectiveness.

When evaluating the Silhouette score of our vector and clustering metric for the 50 test samples using standard score calculations ($s = \frac{b-a}{max(a,b)}$), the result for our metrics was 0.4. For the presented k-value (k=5), this score indicates that our model has a moderate ability to cluster and separate vectors, which offers a passable metric

for standard preference derivation.[26] While this score still can be raised, the current prototype has still demonstrated capabilities that can be strengthened.

After the preliminary model's creation and demonstration, the next necessary step to evaluate the model's efficacy is to measure its effectiveness against human evaluation, and to advance it further. Because the ultimate goal of the model is to make sample organization more friendly to human melodic preference, the model's organization and clustering must align with general pairings. The initial demonstrated samples show promise, as pairings seem to align with general music theory and movement principles. For instance, all three samples in Cluster 0 align with chromatic movement and interval pairings present in D harmonic minor, making it musically viable. However, human testing is needed to ensure an absence of outliers and other outstanding conclusions.

To evaluate the effectiveness of the model, multiple trials would be run with human testers. Batches of 10 randomly selected samples from a large sample library such as Splice would be processed by HAPD, and out of the other 9, the most and least similar sample would be assigned accordingly. For each sample, a tester would be asked to listen to the original sample and its two pairings and asked to evaluate which sample offers the most concordance. If a supermajority of the 10 testers (six or more) evaluate the sample pairings with a rating that aligns with the machine's selection (indicating a moderate to extremely successful pairing), the next trial can commence with increasingly unique sample distributions. Similar studies such as "Evaluating human pairwise preference judgments", conducted at Macquarie University utilize a

similar human evaluation method. They use an output comparison, and compare two results in the form of preference.[27] This eliminates the common issue of Likert-scale ratings in the form of difference magnitudes. The study describes how the use of preferential ratings eliminates the subjective interpretation of the difference between each tier in the scale, and instead provides the judge with a definitive preference score. Using the samples from both extremes of the output is to ensure that, if the option selected by the listener is the opposite of what HAPD outputs, any disagreement has a large implication on the efficacy, and will require revaluation of the model.

Additionally, comparative analysis between HAPD and other classification methods, such as Splice's "similar sounds" will be used in a similar pairwise judgment trial evaluation. HAPD has shown to differ from these models in its process, but whether it proves to be more or less preferable to these methods can only be evaluated through direct comparison. When comparing whether timbral or tonal congruency is prioritized more in interpretation, studies have shown that mode may affect emotional perception of music more impactfully than timbre. For instance, a behavioural study ran by James Armitage concluded that when mode and timbre matched in emotional tone, listeners were able to identify what the clear emotion was than when they differed. It also concluded that mode in the form of structure had more impact in their determination, suggesting that HAPD prioritizing tonal qualities will lead to more desirable results.[23] The ultimate magnitude of this discrepancy can only be evaluated through this analysis.

CONCLUSION & IMPLICATIONS

This innovation has potential to improve the lives of music producers. For example, SageJournals states, “81% rated their financial stress as high or overwhelming” and “Over 45.4% endorsed significant occupational stress”.[28] Stress of producers may be caused by their inability to develop high quality pieces efficiently enough. One cause of this stress is the time-intensive process that is required to combine harmonically compatible samples. HAPD helps to reduce the time needed to find the “right” sample. Aden Russell, a lead at sampling service EDMProd states, “If it takes you 2 minutes instead of 4 to find a sample that fits during the production process, and you use an average of 15 samples per song, that’s 30 minutes that you’ve saved”.[29] Increased efficiency can accelerate project completion and improve releasing or licensing work, which can aid in reducing financial stress.

In summation, the HAPD is highly achievable, although it requires resources and further evaluation to advance and implement the algorithms necessary to fully realize the classification system. Additionally, the intake of additional data provided in most sampling services, like key, tempo, and genre, are also important considerations that need to work in tandem with our model. However, when realized, HAPD could provide a significant impact to workflow, and innovate the way sampling services classify their samples. It can also provide producers with a newer understanding of the tonal structure of the samples they are using, which can allow them to maximize their efficiency and creativity to an extent and with a level of accessibility seldom seen previously.

The current model can be expanded to include the blueprint's chord and stem separation elements. These were intentionally not implemented into the current design of the model, as these are the parameters which will be the most heavily altered during trial. As trials are run, the results will be used to specify regression and sample selection elements; however, the current model already demonstrates a viable alternative to current classification systems. The extent to such will be further gauged during trial. While our proposed architecture details stem separation, our prototype focuses on melodic comparison, which encompasses the most crucial parts of our design. This still leaves part of our design unimplemented, and though our prototype shows promise, the full extent of the pipeline can not be accomplished without execution of the evaluation present in our proposed methodology.

Though it is a tangible product, it is rudimentary, and has segments of our original design that are not included due to the high variance that can only be specified through the data provided by these trials. Additionally, the standard deviation of the chroma vectors over time could be considered in future iterations of the model to better outline how the tonal qualities change over time. Moreover, the chord separation element especially is a key feature that, while the prototype delivers evaluations moderately well without, could improve the novelty and efficacy of our product significantly if it was utilized. In all, the evaluation could have been expanded to live trial, and more time-based features could have been added.

WORKS CITED

[2] Avise, J. C. (2013). *From perception to pleasure: Music and its neural substrates*. National Library of Medicine. Retrieved December 27, 2025, from https://pmc.ncbi.nlm.nih.gov/articles/PMC3690607/

[25] Bruin, J. (2021, August 22). *Logistic Regression Analysis | Stata Annotated Output*. OARC Stats. Retrieved February 10, 2026, from https://stats.oarc.ucla.edu/stata/output/logistic-regression-analysis/

[28] Berg, L., King, B., Koenig, J., & McRoberts, R. L. (2022). Musician occupational and financial stress and mental health burden. Psychology of Music, 50(6), 1801-1815. https://doi.org/10.1177/03057356211064642 (Original work published 2022)

[4] Bhattacharjee, Utpal & Mannala, Jyoti. (2019). An Experimental Analysis of Speech Features for Tone Speech Recognition. International Journal of Innovative Technology and Exploring Engineering. 2. 4355-4360. 10.35940/ijitee.B7748.129219.

[1] Costa, L., & Los Angeles Film School. (2025). *Music Sampling Explained: Creative Techniques and Copyright Laws Every Producer Should Know*. The Los Angeles Film School. Retrieved December 27, 2025, from https://www.lafilm.edu/blog/music-sampling-explained/

[26] Dalmaijer, E. S., Nord, C. L., & Astle, D. E. (2022). Statistical power for cluster analysis. BMC Bioinformatics, 23(1), Article 205. doi.org

[6] de Haas, W. B., Robine, M., & Hanna, P. (2011). *Comparing Approaches to the Similarity of Musical Chord Sequences*. Hal Open Science. Retrieved December 28, 2025, from https://hal.science/hal-01006469v1/document

[27] Dras, M. (2015). Squibs: Evaluating Human Pairwise Preference Judgments. Computational Linguistics, 41(2), 309–317. https://doi.org/10.1162/COLI_a_00222

[17] Eck, A. (2024). *How Music Resonates in the Brain | Harvard Medicine Magazine*. Harvard Medicine Magazine. Retrieved February 10, 2026, from https://magazine.hms.harvard.edu/articles/how-music-resonates-brain

[13] Ellis, D. P.W., & LabROSA. (2007). *CLASSIFYING MUSIC AUDIO WITH TIMBRAL AND CHROMA FEATURES*. Columbia University. Retrieved December 28, 2025, from https://www.ee.columbia.edu/~dpwe/pubs/Ellis07-timbrechroma.pdf

[24] Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). Cluster analysis (5th ed.). John Wiley & Sons. https://doi.org/10.1002/9780470977811

[9] Feng, W., Li, T., & Yang, Z. (2022). COAT: A music recommendation model based on chord progression and attention mechanisms. In Proceedings of the 34th International Conference on Software Engineering and Knowledge Engineering (SEKE 2022) (pp. 616–621). KSI Research Inc. https://doi.org/10.18293/seke2022-115

[20] FL Studio. (n.d.). *Piano roll Basics*. FL Studio. Retrieved December 28, 2025, from https://www.image-line.com/fl-studio-learning/fl-studio-online-manual/html/pianoroll.htm

[18] Freeman J, Simoncelli EP. Metamers of the ventral stream. Nat Neurosci. 2011 Aug 14;14(9):1195-201. doi: 10.1038/nn.2889. PMID: 21841776; PMCID: PMC3164938.

[3] Grey. (2025). *Major Scale Chord Function: How Chords Behave*. Hub Guitar. Retrieved December 27, 2025, from https://hubguitar.com/music-theory/chord-function

[11] Hilsdorf, M. (2023, September 20). *AI Music Source Separation: How it Works and Why It Is So Hard*. Medium. Retrieved December 17, 2025, from https://medium.com/data-science/ai-music-source-separation-how-it-works-and-why-it-is-so-hard-187852e54752

[12] Huss, P. J. (1983, 12). *Vocal Pitch Range and Habitual Pitch Level: The Study of Normal College Age Speakers*. Western Michigan University. Retrieved February 9, 2026, from https://scholarworks.wmich.edu/cgi/viewcontent.cgi?article=2616&context=masters_theses

[15] Lenssen, Nathan & Needell, Deanna. (2014). An Introduction to Fourier Analysis with Applications to Music. Journal of Humanistic Mathematics. 4. 72-91. 10.5642/jhummath.201401.05.

[7] Luo, Y.-J., Agres, K., & Herremans, D. (2019). Learning disentangled representations of timbre and pitch for musical instrument sounds using Gaussian mixture variational autoencoders. In Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR 2019) (pp. 746–753).

[9] Martin, T. (2025, June 5). *Detailed Frequency Ranges of Instruments and Vocals*. The Absolute Sound. Retrieved February 9, 2026, from https://www.theabsolutesound.com/freqchart/main_display.htm

[19] Mencke, I., Omigie, D., & Wald-Fuhrmann, M. (2019, January 7). *Atonal Music: Can Uncertainty Lead to Pleasure?* Frontiers. Retrieved February 10, 2026, from https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2018.00979/full

[8] Müller, M., & Zalkow, F. (2019). *FMP Notebooks: Educational material for foundations of music processing*. AudioLabs Erlangen. https://www.audiolabs-erlangen.de/resources/MIR/FMP/C3/C3S1_SpecLogFreq-Chromagram.html

[16] Music Theory Academy. (2025). *5 basic rules of Chord Progressions*. Music Theory Academy. Retrieved December 28, 2025, from https://www.musictheoryacademy.com/understanding-music/chord-progressions/

[23] Noble, J. (2024). *What is Hierarchical Clustering?* IBM. Retrieved December 28, 2025, from https://www.ibm.com/think/topics/hierarchical-clustering

[21] Oudre, L., Grenier, Y., & IEEE. (2011). *Chord Recognition by Fitting Rescaled Chroma Vectors to Chord Templates*. Institut de Recherche en Informatique de Toulouse. Retrieved December 26, 2025, from https://www.irit.fr/~Cedric.Fevotte/publications/journals/ieee_asl_deterchord.pdf

[14] Reynal, S. (2013). Signaux musicaux: Pitch & chroma representation [Lecture notes]. Équipe Traitement de l'Information et Systèmes (ETIS). https://reynal.etis-lab.fr/docs/sigmus/lecture_pitch_chroma_amea_2013_6.pdf

[5] Splice Records. (2022, December 1). *Use Similar Sounds to find samples on Splice - Blog*. Splice. Retrieved December 27, 2025, from https://splice.com/blog/introducing-similar-sounds/

[22] Stryker, C. (2024). *What Is Unsupervised Learning?* IBM. Retrieved December 28, 2025, from https://www.ibm.com/think/topics/unsupervised-learning

APPENDICES

Appendix A: Input Repository

Hritik Patel, 2025, *Input Audio Repository*, https://drive.google.com/drive/folders/16rF6R7zHW2gO7h--KtVK6qHWVktHoJsx?usp=drive_link

Appendix B: Generated cluster list

Cluster 0

- Model1Input27F#-C#HARM130BpmSineWave.wav
- Model1Input29G-F#HARM130BpmSineWave.wav
- Model1Input34F-F#HARM130BpmSawWave.wav

Cluster 1

- Model1Input10F#note2beatsFnote2beats130BpmSineWave.wav
- Model1Input11Cnote1beatEnote1beatGnote1beatAnote1beat130BpmSineWave.wav
- Model1Input12Anote1beatFnote1beatDnote1beatCnote1beat130BpmSineWave.wav
- Model1Input13Anote1beatDnote1beatFnote1beatGnote1beat130BpmSineWave.wav
- Model1Input14A-F#-F-C-E(eighth)130BpmSineWave.wav
- Model1Input15A-B-F#-G#-F-D#-C-E[eighth]130BpmSineWave.wav
- Model1Input16Cnote4beats130BpmSawWave.wav
- Model1Input17Cnote2beatsCnote2beats130BpmSawWave.wav
- Model1Input18Cnote2beatsBnote2beats130BpmSawWave.wav
- Model1Input19Cnote2beatsF#note2beats130BpmSawWave.wav
- Model1Input1Cnote4beats130BpmSineWave.wav
- Model1Input20G#note2beatsF#note2beats130BpmSawWave.wav
- Model1Input21G#-A#-F#-D#-C#(eighth)130BpmSawWave.wav

- Model1Input22A-F-G-E-D-C-E-G[eighth]130BpmSawWave.wav
- Model1Input23C#chromatictoG#[eighth]130BpmSawWave.wav
- Model1Input24A#chromatictoD#[eighth]130BpmSawWave.wav
- Model1Input25A#-G-A-F-G-D#-E-D[eighth]130BpmSawWave.wav
- Model1Input26A#-C#HARM130BpmSineWave.wav
- Model1Input2G#note4beats130BpmSineWave.wav
- Model1Input32C#-A#HARM130BpmSawWave.wav
- Model1Input36B-A#HARM130BpmSawWave.wav
- Model1Input3Cnote2beats130BpmSineWave.wav
- Model1Input4Cnote3beats130BpmSineWave.wav
- Model1Input5Cnote2beatsA#note2beats130BpmSineWave.wav
- Model1Input6Cnote2beatsF#note2beats130BpmSineWave.wav
- Model1Input7Cnote2beatsDnote2beats130BpmSineWave.wav
- Model1Input8Cnote2beatsC#note2beats130BpmSineWave.wav
- Model1Input9F#note2beatsDnote2beats130BpmSineWave.wav

Cluster 2

- Model1Input28C-C#HARM130BpmSineWave.wav

Cluster 3

- Model1Input30A-A#HARM130BpmSineWave.wav
- Model1Input31D#-BHARM130BpmSineWave.wav

- Model1Input33C#-EHARM130BpmSawWave.wav
- Model1Input35A-G#HARM130BpmSawWave.wav
- Model1Input38D#-G#-BHARM130BpmSineWave.wav
- Model1Input45F#-E-C_G-D#-BHARMOCT130BpmSineWave.wav
- Model1Input46D#-E-C_G#-D#-AHARMOCT130BpmSineWave.wav
- Model1Input47C-E-G#-A#(step)_C-D#-AHARM130BpmSineWave.wav

Cluster 4

- Model1Input37C-E-GHARM130BpmSineWave.wav
- Model1Input39C-F-G#HARM130BpmSineWave.wav
- Model1Input40D-F-A#HARM130BpmSineWave.wav
- Model1Input41D#-F#-A#_C#-E-G#HARM130BpmSineWave.wav
- Model1Input42C-F-G#_C#-F-G#HARM130BpmSineWave.wav
- Model1Input43C#-F#-B_D#-F#-AHARM130BpmSineWave.wav
- Model1Input44E-F#-A_D#-G-BHARM130BpmSineWave.wav
- Model1Input48C#-F-A-B(step)_C-E-G(step)-A#HARM130BpmSineWave.wav
- Model1Input49D-F#-B_D-G-A#_C#-F-AHARM130BpmSineWave.wav
- Model1Input50C#-F-G_C#-F#-A_C-E-G#_D#-F#-A#HARM130BpmSineWave.wav

Appendix C: Program code

Abhinav Damera, 2025, *Harmonic-Aware Pairing Detection*,

https://github.com/adamera-programming/Harmonic-Aware-Pairing-Detection-HAPD-.git