# AUDIO SIMILARITY ANALYSIS USING DATA SCIENCE

by

JACK CANNON IV

A THESIS

Presented to the Department of Data Science
and the Robert D. Clark Honors College
in partial fulfillment of the requirements for the degree of
Bachelor of Science

May 2025

# An Abstract of the Thesis of

Jack Cannon IV for the degree of Bachelor of Science
in the Department of Data Science to be taken June 2025

Title: Audio Similarity Analysis Using Data Science

Approved: *Dr. Jeffrey Stolet*
Primary Thesis Advisor

The inconsistent application of copyright law is an ongoing issue in the music industry. For much of the past century, music artists have sought to resolve accusations of copyright infringement in the court room. These trials are often seen as 'unfair' and 'inconsistent' because of the basis on which verdicts are determined and how widely the results can vary.

In this study, I analyze a limited number of select infringement cases from a data science perspective. I will employ objective, algorithmic methods to determine the similarities and differences between musical selections considering harmonic, melodic, and other data structures. The goal is to conceptualize and derive a 'Total Similarity Score' that seeks to determine possible copyright infringement without subjective judgment. To do this, I will (1) strip infringement case song pairs down to their "musical building blocks," (*Structured Asset Sales, LLC v. Ed Sheeran*) (2) algorithmically analyze and compare song metadata, (3) calculate a numerical degree of similarity, and (4) estimate an "Infringement Threshold" beyond which a copyright infringement is likely to have occurred. Finally, I will match these results with the corresponding trial verdicts to observe the efficacy of our current legal system using this proposed, objective standard.

To narrow my focus, I will not analyze factors such as lyrics and timbre, although I do recognize that they can be used to evaluate similarity. The songs that I selected are taken from 10

notable copyright infringement cases of the past century (e.g., Led Zeppelin's "Stairway to Heaven" and Spirit's "Taurus"). I will also use Essentia, a C++ and Python library, to analyze the audio, extract metadata, and calculate factor similarity. The final comparative calculations and analyses will be completed using tools found in Microsoft Excel.

Fairness requires a degree of consistency, which is arguably a natural by-product of objective data analysis. Adjudicating infringement based on examination of metadata would theoretically mitigate the perception of unfairness and legal manipulation. While reality is understandably more complex, contrasting the current system with one that is based on objective measures may prove to be both interesting and instructive. With this effort, I hope to present findings that could help future scholars to discuss, and perhaps improve, the manner in which courts decide infringement liability.

# Acknowledgements

I would like to thank my parents and extended family for continuing to support me and my college tenure these past five years, including scholarships and student billing.

I would also like to thank Professor Lydia Van Dreel for agreeing to serve on my Thesis committee and for reviewing the final document.

Professor Brian McWhorter receives my gratitude for helping to narrow the scope of my project and for making suggestions on my Prospectus document in Fall 2024.

Professor Larry Wayte receives my thanks for teaching MUS 346: “Music, Money, and the Law” in Spring 2024, which gave me the idea for this project and helped with understanding the nuances of music copyright law and background information for my project.

I would additionally like to thank Professor Andrew Muehleisen for helping to troubleshoot issues with the installation of Essentia for my project.

Finally, I would like to thank my Primary Thesis Advisor, Dr. Jeffrey Stolet, for helping to select a Thesis topic and for encouraging me not to give up in the face of 1) issues that arose with the project, and 2) the time pressure of getting the work finished in advance of the deadline.

# Table of Contents

Jupyter Notebook: thesis_cover_song_similarity.ipynb (converted to PDF)
Jupyter Notebook: thesis_cover_song_similarity_error_testing.ipynb (converted to PDF)
Jupyter Notebook: thesis_tempo_similarity.ipynb (converted to PDF)
Jupyter Notebook: thesis_chords_similarity_key_match.ipynb (converted to PDF)
Jupyter Notebook: thesis_melody_similarity.ipynb (converted to PDF)
Jupyter Notebook: tutorial_similarity_cover.ipynb (converted to PDF)
Jupyter Notebook: tutorial_rhythm_beatdetection.ipynb (converted to PDF)
Jupyter Notebook: tutorial_tonal_chords.ipynb (converted to PDF)
Jupyter Notebook: tutorial_tonal_hpcpkeyscale.ipynb (converted to PDF)
Jupyter Notebook: tutorial_pitch_melody.ipynb (converted to PDF)

# List of Figures

# List of Tables

# Introduction

What precisely constitutes copyright infringement in the music industry? We generally understand infringement to be when someone has copied all or part of another person's work without permission. If an artist sees another artist's work as being too similar to their own, they can try to take the matter to court to seek the protections afforded under copyright law. There are two possible beneficial outcomes of escalating a copyright dispute. One is of course a trial whereby a 'liable' verdict may result in compensatory payment or royalties from the accused party. The other possibility is a settlement, which can happen before or during trial proceedings and tends to be beneficial for the accuser as well. A defendant, on the other hand, may try to settle a case for several possible reasons, but there are two that tend to be most common. First, there may be implied acknowledgement from the accused that a guilty verdict is likely. To stem the cost of completing a full trial, and to perhaps exercise a degree of control over the financial liability, a settlement may be preferable. A second common reason is that the accused, though certain there was no plagiarism (intentional or otherwise) in the development of their work, is nonetheless unsure about their chances of exoneration in a trial proceeding. Knowing that the cost of a trial could be much higher, a settlement offer may be presented to appease the accuser.

The question of what constitutes 'similarity' and 'infringement' is often difficult to answer because these concepts are both inherently subjective and open to interpretation. Consequently, many court cases have been decided based on extrinsic factors that have little to do with intrinsic (e.g., melody, harmony, rhythm, etc.) similarity. One example of an external factor is determination of the applicable law—the Copyright Act of 1909 or 1976—on which the proceedings are governed. The law chosen by the court determines the type of *deposit copy* a jury can access when comparing compositions. The 'deposit copy' is the documented

representation of the work on which the copyright is based. The 1909 Copyright Act requires the written notation (i.e., sheet music) to serve as the basis for copyright protection. The 1976 Copyright Act expands this restriction to include musical recordings. If one or both songs in a court proceeding were composed and copyrighted before January 1, 1978, the 1909 Act would apply thus requiring the comparison to be based *only* on the notation. This could influence the outcome by greatly limiting accessibility for those who cannot read sheet music, or by ignoring musical elements that lack formal notation. For those works copyrighted after January 1, 1978, the 1976 Act would take precedence—thereby allowing a comparison of *sound recordings.* Accessibility is largely mitigated in this case because juries may not need to be well-versed in music analysis or composition to judge similarity. It also exposes them to musical nuances and details that may be absent on the sheet music.

Another extrinsic consideration is **the 'copyrightable' or 'protectable' nature of the musical elements comprising a song.** *Protectability*, in the context of a copyright proceeding, questions if there is sufficient originality to give the accusing party legitimate grounds to claim ownership of a copyrightable asset. This is a highly debated matter because originality is extremely subjective when considering chord progressions, motifs, instrumentation, etc. for a given genre. This is further complicated when the elements in question are short in duration or as minute as a singular sound, tone, non-lexical vocables (sounds that lack semantic meaning), etc. that may or may not be truly unique. The ultimate question is whether such elements matter enough to cement the originality of a song.

Finally, we have **the *de minimis* exception** (referencing a Latin saying that means "the law does not concern itself with trifles"), used for any claim that is deemed 'too trivial' to go to court. Related terminology includes the *standard of substantial similarity*, or the potential for a

copied expression to be deemed infringement. Theoretically, if comparing two songs leads to a result that falls below either standard, it is subject to dismissal. The issue, of course, is that 'de minimis,' 'substantial,' and 'too trivial' are not clearly defined in U.S. copyright law (see Literature Review).

Extrinsic factors introduce complications in every copyright proceeding, but there are other basic considerations that challenge consistency and fairness. Venue is one example. Different courts, judges, and juries can mean differing opinions about 'similarity,' 'infringement,' and the basic elements of music. Discretion, or the freedom to decide what should be done in a particular situation, is another consideration. The selection and interpretation of applicable copyright law determines what is to be presented to juries, yet courts have historically exercised discretion in this regard. Finally, appeals and re-trials introduce even more judges, juries, interpretations, and opinions.

Perhaps the most daunting of issues complicating copyright enforcement is technology. The rate of technological advances in recent decades has far outpaced that of the corresponding legal guardrails. One such advancement is *digital sampling*, which allows an artist to directly and seamlessly appropriate elements of another song into their own work with little effort. Combined with digital looping—or repeating a sample rhythmically—and other sound manipulation techniques, the issue of de minimis becomes even more obscure. Another example is the advent of *streaming and Internet influencer marketing.* The Internet is an ever-expanding library of content that allows any user to access practically any song that is posted there. It is unclear, however, if posted content enjoys the same protection under current copyright law. As only two versions of the Copyright Act were created in the 20th century, the law has simply not kept pace.

These complications foster inconsistencies that make it difficult to establish legal precedent, and that in turn adds to the perception of unfairness. Contrast *Pharrell Williams, et al. v. Bridgeport Music, et al.*, which used the 1976 Copyright Act, and *Michael Skidmore v. Led Zeppelin*, which instead used the 1909 Act. Both cases involved one or more songs that were copyrighted prior to January 1, 1978, and yet court discretion differed in selecting the applicable law to govern the proceedings. Many artists and critics were understandably outraged by the Bridgeport ruling. Other cases appear to have been decided purely due to technicalities. The initial ruling of the Zeppelin trial was appealed because the jury had not been allowed to compare sound recordings despite the ongoing popularity of "Stairway to Heaven." The 3-judge appellate court agreed, nullifying the verdict and remanding it to the lower court for retrial. Zeppelin's attorneys pursued a secondary appeal featuring a higher *en banc* (eleven judge) panel—a technicality within the appellate system. This panel reversed the order to remand, thereby reinstating the original verdict of 'not liable.'

A data science-based approach to defining 'copyright infringement' would differ significantly from the current legal system. While unable to address many of the subjective complications described above, the benefits of such an approach are no less compelling. Chief advantages include:

(1) Objective vs. Subjective Decisioning

(2) Intrinsic vs. Extrinsic Factor Consideration

(3) Repeatability

(4) Immunity to Extraneous Factors (e.g., Venue, Judge, Jury, etc.)

In short, this study examines how well an objective simulation that measures and calculates the degree of musical similarity would match real-world verdicts. I hypothesize that this alternative approach would better answer the original question ("What precisely constitutes copyright infringement?") because an assessment of infringement by objective means would be more consistent, measurable, and perhaps by extension, fair.

## Literature Review

Existing literature on this subject shows that many factors are considered when trying to determine if a copyright has been infringed, but the guidelines that govern these factors leave much room for discretion, interpretation, and inconsistency.

Cases involving digital sampling are instructive in this regard. The *bright line rule* states that any amount of sampling without permission (regardless of length or recognizability) can lead directly to a liable verdict. John Oswald infamously explored this concept using a technique called *Plunderphonics* (i.e., the sampling and transformation of existing music recordings) to challenge the definition of "authorship" in digital music and the efficacy of copyright protection in this context. "Z," an 18-second recording entirely comprised of tiny samples of New York saxophonist John Zorn, is an example of his work. Oswald, perhaps unsurprisingly, was found liable for plagiarism and all copies of his "Plunderphonic CD" were destroyed (Emmerson 2016). Other cases centered on alleged sampling as well. In *VMG Salsoul, LLC v. Madonna Louise Ciccone*, the court held that "'de minimis' copying does not constitute infringement […] because the copying (if it occurred) was trivial." (9th Cir. Opinion by Judge Graber) Here, the bright line rule seemed to have no effect on the verdict.

Another aspect often considered is perceived **character and intent**. Several pertinent questions help to shed light on the topic: (1) What if someone wrote a *parody* of another artist's song—the same exact melody but with different lyrics? In *Campbell v. Acuff-Rose*, Luther Campbell was sued for "Pretty Woman" which had the same melody, harmony, and similar title as "Oh, Pretty Woman" by William Dees and Roy Orbison. The song was purposefully intended as a parody that satirizes the original work. Despite the opportunistic copying involved, the court decided that it was fair use and ultimately ruled in favor of the defendant. Exposed by this

outcome is the lack of any clear delineation between intentional infringement and what is considered "fair use" (e.g., parody).

(2) What if someone wrote an *homage* to your work—an original song that intentionally sounds like something you would have written? When Chuck Berry threatened to sue the Beach Boys for their song "Surfin' U.S.A.," based on Berry's "Sweet Little Sixteen," they admitted that it was directly based on his work and paid him before the matter could go to trial. Given the aforementioned "fair use" judgment, it would be rather interesting to know the outcome of a court proceeding.

(3) What if the copying was *subconscious*, or accidental? This was the defense's testimony in *Bright Tunes Music v. Harrisongs Music.* George Harrison admitted to "striking similarities" in melody and harmony when comparing his song to that of his accuser, yet he maintained that these similarities were purely accidental. The court rejected Harrison's defense and ruled against him. Wayte asserts that "lack of intent does not constitute a defense to copyright liability—unconscious copying can result in the same liability for infringement as deliberate plagiarism." (Wayte 2023)

The standards of **'substantial similarity'** and **'de minimis,'** coupled with the question of protectability or copyrightable expression, are yet another factor. In *Skidmore v. Zeppelin*, the question was whether an "arpeggiated chordal guitar part in A-minor over a similar descending chromatic bass line" was enough to warrant liability (Wayte 2023). The similarity and alleged infringement are confined to the intros; the balance of the songs differ completely. The defense argued that the elements in question are considered "commonplace" in music and, as such, are not protectable. The court agreed. Similarly, the discussion during *Williams v. Bridgeport* centered around the 'copyrightability' of an infamous cowbell rhythm and instrumentation not

included on the sheet music. In that case, the court disagreed with the defense's argument that these elements were common and therefore not subject to copyright protection.

In *VMG Salsoul v. Madonna*, the question was whether a sampled horn hit, lasting just 0.23 seconds, was substantial and original enough to be protectable. For cases such as these, the court must decide which guideline is applicable and will govern the outcome—the bright line rule or the de minimis exception? The opposing nature of these rules complicates this task.

As discussed previously, **the Copyright Act used to govern specific disputes** has generated much confusion and controversy in recent court proceedings. Many jurists have never studied music. Given the task of assessing infringement based on sheet music alone can be problematic and can adversely affect the final judgment. In *Skidmore v. Zeppelin*, had the jury been allowed to hear the sound recordings instead of relying solely on sheet music and the testimony of an 'expert' witness, the outcome may have been different.

Assessing copyright infringement based on inherent structural elements such as tempo, timbre, rhythm, chords, and melody, is often clouded by extrinsic factors as well. This was famously discussed in *Structured Asset Sales, LLC v. Sheeran*, wherein Ed Sheeran's "Thinking Out Loud" was accused of infringing on the copyright to Marvin Gaye's "Let's Get It On." The premise of the lawsuit was simple: Because the two songs used similar "building blocks" (i.e., harmonic chord progression), Sheeran must have appropriated Gaye's work. Sheeran was, of course, outraged:

> "I am unbelievably frustrated that baseless claims like this are allowed to go to court at all. We have spent the last eight years talking about two songs with dramatically different lyrics, melodies and four chords which are also different and used by songwriters every day, all over the world. These chords are common building blocks which were used to create music long before 'Let's Get It On' was written and will be used to make music long after we are all gone. I am just a guy with a guitar who loves writing music for people to enjoy. I am not and will never allow myself to be a piggy bank for anyone to shake. […] To have someone

> come in and say, ‘We don’t believe you, you must have stolen it,’ I find that really insulting.” (Sisario 2023)

While the court ruled in favor of the defendant (Sheeran) in this trial, the discussion of “common [musical] building blocks” raises the question of how factors like melody, chords, and harmonies are determined to be commonplace when assessing infringement? There is a disturbing lack of resources dedicated to answering this question definitively. Rather, in practice it remains the topic of subjective opinion, which varies widely. Wayte perhaps puts it best:

> “Can a rhythmic groove be copyrighted? A cowbell pattern? A background keyboard part? Does any song that imitates the generic groove or feel of a historical style now potentially violate the copyright of every song that also used those generic elements of that style? What are the limits to that approach to music copyright?” (Wayte 2023)

Since opinions can differ significantly between courtrooms, judges, and juries, and can change over time, funneling them into a comprehensive and consistent standard is understandably difficult. This is the foundational issue that this project seeks to examine.

## Significance: Project Description

Courts have been rather inconsistent in determining what the exact threshold of similarity should be in adjudicating infringement. In this project I approach the topic from a data science perspective. Using analytical techniques, I derive and define a 'Total Similarity Score' for use in comparing songs highlighted in recent infringement cases. The results are then compared with the actual court verdicts and settlements. Throughout this process, (1) important similarities and differences in outcomes will be examined, (2) possible explanations for these comparisons will be explored, and (3) thoughts about the significance of this project in the broader context will be offered.

The specific research questions to be addressed are as follows:

1. What types of features can I extract from different audio recordings for classification?
2. By comparing and analyzing these features for any two songs, can I determine a level of similarity?
3. Can I determine a threshold for substantial similarity in a way that resembles the court's rulings as closely as possible?
4. What types of patterns emerge when comparing this result to real-world verdicts?
5. How can this information help future scholars in music copyright law?

It is important to note that my project aims neither to detect copyright infringement directly, nor to completely solve these complex issues. Instead, the intent is to examine the efficacy of music copyright law enforcement from an alternate paradigm. My hope is that this project will be a foundational tool that helps future scholars design studies that build upon the results. Perhaps someday these efforts will inform and influence music copyright enforcement to be more consistent, while strengthening the underlying legal frameworks to match present day

considerations. This will in turn get us closer to comprehensively defining what constitutes copyright infringement in the music industry.

# Methods and Data Analysis

In this section, I provide a detailed description of how the analysis was conducted, the data collection techniques, and the programming tools used in the process. The employed methodology consisted of four main steps:

- **Create a data science program** capable of analyzing the similarities between pairs of songs based on structural music components (or "*Factors*").
- **Analyze each song pair** using this program to obtain Similarity Rating values for each distinct Factor.
- **Derive a *Total Similarity Score*** based on these similarity ratings for use in determining copyright infringement potential.
- **Compare the results** to the real-world verdicts for each case.

## Data Collection

A total of 20 songs were selected from 10 recent, high-profile copyright infringement disputes. The audio files for each song were gathered and converted to .WAV format using a free trial of ViWizard's Apple Music Converter. Table 1 lists information about the cases.

| Case/Songs | Settle-ment? | Infringe-ment? | Appeal? | Over-turned? | 1909 Copyright Act? | 1976 Copyright Act? | Not Protect-able? | De Minimis Except-ion? | Other Notes |
|---|---|---|---|---|---|---|---|---|---|
| **Michael Skidmore v. Led Zeppelin** Taurus -- Stairway to Heaven | | N/N | Y | N | X | | X | | |
| **Joe Satriani v. Christopher Martin, et al.** If I Could Fly -- Viva La Vida | X | ? | N/A | N/A | | X | | | Claim dropped after undisclosed settlement |
| **Structured Asset Sales LLC v. Sheeran** Let's Get It On -- Thinking Out Loud | | N/N | Y | N | | X | X | | |
| **VMG Salsoul, LLC v. Madonna Louise Ciccone** Vogue -- Love Break | | N/N | Y | N | | X | X | X | |
| **Marcus Gray, et al. v. Katy Perry** Joyful Noise -- Dark Horse | | Y/N | Y | Y | | X | | | Deemed not similar enough |
| **Queen & David Bowie v. Vanilla Ice** Under Pressure -- Ice Ice Baby | X | Y | N/A | N/A | | X | | | Settlement after threatened lawsuit |
| **Chuck Berry v. The Beach Boys** Sweet Little Sixteen -- Surfin' USA | X | Y | N/A | N/A | | X | | | Admitted to copying |
| **Pharrell Williams, et al. v. Bridgeport Music** Got to Give It Up -- Blurred Lines | | Y | Y | N | | X | ? | | Should have used 1909 law? |
| **Bright Tunes Music v. Harrisongs Music** He's So Fine -- My Sweet Lord | Failed | Y | N | | ? | ? | | | Admitted to "striking harmonic and melodic similarities" |
| **Tom Petty v. Sam Smith** I Won't Back Down -- Stay With Me | X | ? | N/A | N/A | | X | | | Settlement before trial verdict |

Table 1: All ten court cases used in the study. (Similarity Chart.xlsx)

Presented here are the case names, the specific song titles in question, and other pertinent information that may have affected the outcome.

**Essentia**

Essentia (version 2.1 beta6) is a free downloadable programming library that supports the C++ and Python programming languages. It is used to upload selected audio signals and to implement various methods of audio analysis and feature extraction using algorithmic tools. Essentia can also generate visualizations of the analysis results to support the numerical output. This project uses the following Essentia tools in its calculations:

- **ChromaCrossSimilarity and CoverSongSimilarity.** In Essentia (and perhaps in popular music more broadly), a *cover song* refers to a new rendition of an existing song by a different artist. To the music theorist, both songs are derived from the same composition. The cover can be drastically different in key, tempo, instrumentation, structure, etc., but the compositional aspects of the music will (by definition) remain intact. Essentia's Cover Song ID algorithm uses the ChromaCrossSimilarity and CoverSongSimilarity functions to calculate a *similarity distance value* as a percentage, where greater degrees of similarity generate values closer to 0% and higher percentages indicate dissimilar compositions. Note that this is the only tool of the five which directly calculates a value of similarity.
- **PredominantPitchMelodia.** This algorithm analyzes the *pitch contour* of the predominant melody; in other words, it estimates the pitch values of the melody in Hz as a sequence over a certain time period. The output is a time series containing this sequence of pitch values.
- **ChordsDetection.** This algorithm estimates the harmonic progression of a song. It outputs a time series of strings that describe the chords (e.g., ["C", "Dm", "F", "G"]).

The letters refer to the root note of the chord, and "m" denotes whether it is a minor chord as opposed to a major chord.

- **RhythmExtractor2013.** This algorithm is used for automatic beat and tempo estimation. It outputs the estimated tempo of the entire song in beats per minute, as well as time positions of each beat and the overall *confidence* (probability) of the accuracy.
- **HarmonicPitchClassProfile (HPCP).** While used to detect the harmonic progression, it can also estimate the key and scale of any given song. The generated visualization is a 12-bin histogram (bins corresponding to the pitches A to G#) that shows the intensity of certain pitches over time. Essentia's key detection algorithm outputs a description of the key as a string (e.g.: "C major" and "D minor").

**Similarity Rating Values**

Each Essentia algorithm is processed in a **Jupyter Notebook**. This is a file used to run snippets of Python code in a certain order (as opposed to running the entire program at once) and under different scenarios. Each of the four files (thesis_cover_song_similarity.ipynb, thesis_tempo_similarity.ipynb, thesis_melody_similarity.ipynb, and thesis_chords_similarity_key_match.ipynb) represents one or more Essentia tools that I use to analyze all 10 cases. "Chords Similarity" and "Key Match," two separate factors that I analyze, are part of the same file for simplicity. To easily detect similarities between the chords, I must first detect the keys of each song and transpose them to match. As depicted in Figure 1a and 1b, the process of transposing involves mapping every possible chromatic pitch to a certain number (while also keeping track of minor keys), adding and subtracting the number as necessary to arrive at the same pitch (I chose 'C' = 3, but any key is sufficient), and mapping the numbers back to pitches.

```
[30]: key_to_int = {"A": 0, "Bb": 1, "B": 2, "C": 3, "C#": 4, "D": 5,
                    "Eb": 6, "E": 7, "F": 8, "F#": 9, "G": 10, "Ab": 11,

                    "A#": 1, "Db": 4, "D#": 6, "Gb": 9, "G#": 11} # Extra note names, just in case

      int_to_key = {0: "A", 1: "Bb", 2: "B", 3: "C", 4: "C#", 5: "D",
                    6: "Eb", 7: "E", 8: "F", 9: "F#", 10: "G", 11: "Ab"}

      def f(name):
          return f"Thesis Music/{name}.wav"

      # Gets rid of repeats
      def unique_list(old_li):
          prev = ""
          new_li = []
          for i in old_li:
              if i != prev:
                  new_li.append(i)
              prev = i
          return new_li

      # Ensures that num always stays between 0-11
      def subtract_loop_12(num, op):
          if not 0 <= num <= 11:
              print("Error: number not in range(0, 12)")
              return
          if not (isinstance(op, int)):
              print("Error: op must be -1, 0, or 1")
              return
          num += op
          while not 0 <= num <= 11:
              if num < 0:
                  num += 12
              elif num > 11:
                  num -= 12
          return num
```

Figure 1a: Helper functions for transpose(). (thesis_chords_similarity_key_match.ipynb)

The mapping schemes ‘key_to_int’ and ‘int_to_key’ are used for the transpose() function (see Figure 1b). The function f() simplifies the process of importing songs since the .WAV audio files are all contained within a folder called “Thesis Music.” The formatted string ensures that ‘name’ can be used as a stand in for the name of the audio file. unique_list() is used during the process of determining chord similarity by removing duplicate elements from the input old_li. Lastly, the function subtract_loop_12() is used to ensure that all integers stay in the range (0, 11), which makes transposition easier. For example, a value of –1 will underflow to 11, and a value of 12 will overflow to 0.

```
#The following note names are used in the output: "A", "Bb", "B", "C", "C#", "D", "Eb", "E", "F", "F#", "G", "Ab".
#https://essentia.upf.edu/reference/std_ChordsDetection.html
def transpose(chords, key, return_chords=True):
    if len(chords) == 0:
        print("Error: 'chords' cannot be empty")
        return

    minor_chords = [0 for i in range(len(chords))] # 0 = major, 1 = minor
    chord_nums = [0 for i in range(len(chords))]
    new_chords = []
    op = key_to_int["C"] - key

    # Convert keys to numbers and extract minor keys
    for i in range(len(chords)):
        if "m" in chords[i]:
            chords[i] = chords[i][:-1]
            minor_chords[i] = 1
        chord_nums[i] = key_to_int[chords[i]]

    # Transpose everything to C (3)
    for i in range(len(chord_nums)):
        chord_nums[i] = subtract_loop_12(chord_nums[i], op)

    if return_chords:
        # Convert numbers back to chords, reinserting minor 'm'
            # If the chord is one of the extra chords (see above),
            # for example G#, it will be turned into Ab even if minor
            # However, this likely will not cause any issues with my analysis
        for i in range(len(chord_nums)):
            new_chords.append(int_to_key[chord_nums[i]])
            if minor_chords[i]:
                new_chords[i] += "m"
        return new_chords
    else:
        return chord_nums
```

Figure 1b: The transpose() function. (thesis_chords_similarity_key_match.ipynb)

Applies the mapping scheme (see Figure 1a) to the input list of chords. The input 'key' is the estimated key of the song detected using Essentia, used to determine how far to transpose to reach the target key (in this case, C). After transposing all chords to C (major or minor) using subtract_loop_12(), the numbers are converted back into chord names. The function optionally returns either a list of chords or a list of numbers depending on the value of the input Boolean variable 'return_chords.'

The similarity rating for each Factor is calculated as a percentage value where higher values represent greater similarity. Because each Essentia algorithm outputs a unique result type, I use differing methods to generate comparable values.

- **The Cover Song ID's similarity distance is conceptually close to the desired similarity rating, except inverted.** In this analysis, *lower* percentages will indicate a *lesser* degree of similarity. Essentia's calculated distance output must therefore be subtracted from 100% to get the appropriate similarity rating. For example, if Essentia outputs a distance value of 15%, the similarity rating for this analysis will be 100% – 15% = 85%.

- **There are two possible outcomes for evaluating key similarity: match or no match.** The Key Match similarity rating is therefore a binary percentage value where 25% = "no match" and 75% = "match." Originally 0% and 100% (respectively) were used to represent match status, but initial testing revealed that detecting "no match" (0%) for two songs would completely cancel out this Factor. Using a nonzero percentage regardless of the output ensures that no Factor is ignored, and that non-key matches would still contribute to the calculated Total Similarity Score.
- **All other similarity ratings (harmony, melody, and tempo)** are calculated using the Jupyter function depicted in Figure 2 that compares two sets of values (song data extractions) for their similarities. This calculation is determined by the formula:

$$S = \frac{L(I)}{L(U)}$$

where *S* is the similarity rating value, *I* is the intersection of two sets (containing the *common* elements), *U* is the union (containing all elements found in *either or both* sets), and *L*() is the length function that counts how many elements are contained in *I* or *U*.

```
def similarity(x, y):

    # Error handling
    if not isinstance(x, list) or not isinstance(y, list):
        print("Error: Both x and y should be lists")
        return

    # Find common elements in both lists
    common = set(x).intersection(set(y))

    # Find total unique elements in both lists
    total = set(x).union(set(y))

    # Calculate proportion similarity
    return len(common) / len(total)
```

Figure 2: The similarity() function. (melody_similarity.ipynb and other files)

**Deriving the Total Similarity Score**

The Total Similarity Score is a *weighted average* of the similarity ratings calculated for each Factor. My goal was to make the score (1) easy to comprehend, (2) comparable to an

estimated Infringement Threshold, and (3) comparable to other similarly evaluated cases. Deriving one value to represent five independent tests satisfied these objectives.

Intuitively, each Factor offers a necessarily unique contribution toward the overall similarity calculation. As such, the weight assigned to each Factor when determining the Total Similarity Score should reflect its importance in relation to other Factors. All weight values were originally written as a proportion between 0.01 (defined as *"minimum weighting"*) and 1 (*maximum weighting*). The minimum weighting is nonzero because a weight value of 0 cancels the Factor, meaning it would have no impact on the final calculation. The weights were then normalized to reflect relative importance. For simplicity, I will be referencing the normalized percentage values for Factor weights instead of the proportions that I originally selected. This allows the reader to easily understand the contribution that each Factor provided in deriving a specific Total Similarity Score.

**Comparing Results to Reality**

As a reminder, one objective of this study is to compare the simulation results with corresponding real-world verdicts. Given the calculation described above, the Total Similarity for any analyzed song pair will vary based on the chosen Factor weights. To facilitate this comparison, Factor weightings were adjusted over 15 'Trials' to affect the calculated Total Similarity Score simultaneously for all cases (see Appendix A: Similarity Chart). Each of these 15 Trials incorporated a manually derived weighting distribution based on intuitive (subjective) expectations. In a few additional Trials, the Solver algorithm in Microsoft Excel was used to identify the optimal weightings for a perfect match. This effort proved to be quite informative.

The analysis also revealed that a *margin of error* must be considered in calculating the Total Similarity Score. I noticed early in the process that comparing a song with itself using the

Cover Song ID algorithm does not output a distance value of 0 (i.e., 100% similarity rating) as would be expected. Instead, the value lies around 1-2% (98-99% similar) indicating that the algorithm is not perfectly accurate. To calculate the error, I duplicated the Cover Song ID program file and compared each song *against itself* (20 total analyses). Results varied slightly, but were fairly consistent (see Table 2). Calculating the average yielded the approximate error value of 1.4% that will be used in this analysis.

| Song Titles | Essentia's Similarity Distance | Calculated Similarity Rating |
|---|---|---|
| He's So Fine | 2.0% | 98.0% |
| My Sweet Lord | 1.3% | 98.7% |
| Under Pressure | 1.4% | 98.6% |
| Ice Ice Baby | 1.3% | 98.7% |
| Vogue | 1.2% | 98.8% |
| Love Break | 1.1% | 98.9% |
| Let's Get It On | 1.4% | 98.6% |
| Thinking Out Loud | 1.3% | 98.7% |
| If I Could Fly | 1.1% | 98.9% |
| Viva La Vida | 1.4% | 98.6% |
| I Won't Back Down | 1.6% | 98.4% |
| Stay With Me | 1.6% | 98.4% |
| Sweet Little Sixteen | 1.6% | 98.4% |
| Surfin' U.S.A. | 1.8% | 98.2% |
| Joyful Noise | 1.2% | 98.8% |
| Dark Horse | 1.5% | 98.5% |
| Got to Give It Up | 0.8% | 99.2% |
| Blurred Lines | 1.3% | 98.7% |
| Taurus | 1.7% | 98.3% |
| Stairway to Heaven | 1.0% | 99.0% |

| **Average of Error Percentages:** | **1.4%** | 98.6% |
|---|---|---|

Table 2: Error values calculated for every song in the database. (Similarity Chart.xlsx)

Lastly, I defined and established the *Infringement Threshold* for each trial, above which a Total Similarity Score is likely to signify infringement liability. A successful trial is therefore achieved by satisfying these criteria:

1. All real-world **liable** verdict and **settlement** cases have Total Similarity Scores that *exceed* the Infringement Threshold, and
2. All real-world **non-liable** verdict cases have Total Similarity Scores that *fall below* the Infringement Threshold.

An unsuccessful trial is characterized by having at least one *'breach.'* Specifically, a breach is defined as:

1. A real-world **liable** verdict or **settlement** case with a Total Similarity Score that *falls beneath* the Infringement Threshold, or
2. A real-world **non-liable** verdict case with a Total Similarity Score that *exceeds* the Infringement Threshold.

Figure 3 shows an example trial visualization using specified Factor weightings, an estimated Infringement Threshold, and the average error value.

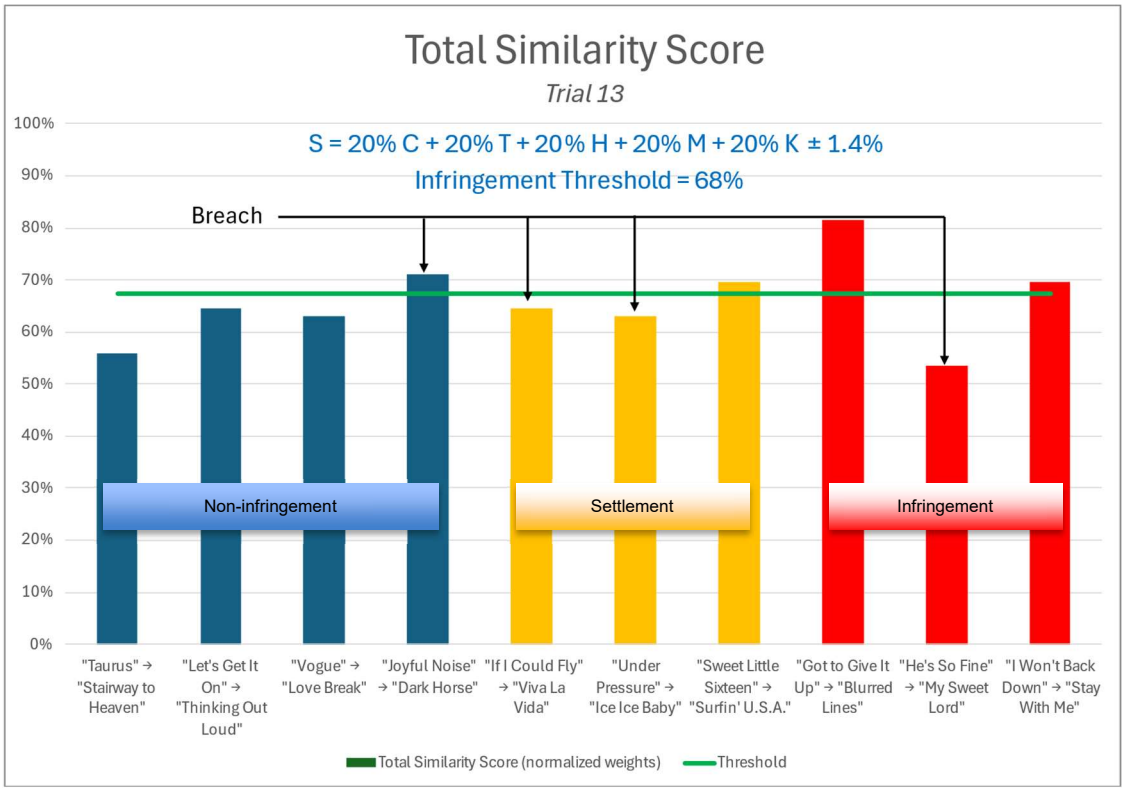



Figure 3: The Total Similarity Scores based on the equal weighting distribution of Trial 13. (Similarity Chart.xlsx)

As will be discussed later, the four breached cases depicted in this graph are the same in all weighting trials that produced exactly 4 breaches, except for Trial 11.

## The Total Similarity Score

To recap, the Total Similarity Score of any song pair featured in a copyright infringement dispute is determined by the formula:

$$S = Cx_C + Tx_T + Hx_H + Mx_M + Kx_K \pm E$$

where:

- ***S*** = the Total Similarity Score (percentage).
- ***C*** = the similarity rating for Cover Song ID (percentage).
- ***T*** = the Rhythmic/Tempo similarity rating (percentage).
- ***H*** = the Harmonic similarity rating (percentage).
- ***M*** = the Melodic similarity rating (percentage).
- ***K*** = Key Match similarity rating (no match = 25%, match = 75%).
- ***E*** = the average margin of error (percentage) ≈ 1.4%.
- ***X*** = $x_C$, $x_T$, $x_H$, $x_M$, $x_K$ = normalized weights applied to each Factor that determine its relative contribution.

As an example, the same formula is written below using Factor weightings from Trial 15 (with the estimated Infringement Threshold set to 58%):

$$S = C(4.35\%) + T(43.48\%) + H(4.35\%) + M(4.35\%) + K(43.48\%) \pm 1.4\%$$

Using this calculation, a case with a 'liable' or 'settled' outcome (i.e., infringement) should have an *S* value that equals or exceeds 58% (± 1.4%). Those with an *S* value less than 58% (± 1.4%) should yield an assessment of 'not liable' (no infringement). Any case for which these criteria are not upheld is considered 'breached.'

## Results and Key Observations

Table 3 shows a consolidated table of Total Similarity Scores (plus breaches) for each trial, and Figure 4 depicts the weighting distributions in graph form. Below I discuss the most striking observations in conducting the analysis and the ramifications thereof.

Total Similarity Score

| Trial | Taurus / Stairway to Heaven | Let's Get It On / Thinking Out Loud | Vogue / Love Break | Joyful Noise / Dark Horse | If I Could Fly / Viva La Vida | Under Pressure / Ice Ice Baby | Sweet Little Sixteen / Surfin' USA | Got to Give It Up / Blurred Lines | He's So Fine / My Sweet Lord | I Won't Back Down / Stay With Me | Number Breached |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 57% | 71% | 71% | 76% | 68% | 62% | 76% | 90% | 55% | 73% | 4 |
| 2 | 61% | 74% | 72% | 80% | 71% | 67% | 79% | 89% | 57% | 77% | 4 |
| 3 | 63% | 67% | 64% | 78% | 66% | 70% | 73% | 88% | 50% | 79% | 3 |
| 4 | 60% | 72% | 69% | 79% | 69% | 65% | 76% | 91% | 54% | 76% | 5 |
| 5 | 62% | 74% | 72% | 81% | 72% | 69% | 79% | 87% | 58% | 79% | 4 |
| 6 | 64% | 75% | 75% | 84% | 77% | 75% | 83% | 80% | 62% | 83% | 4 |
| 7 | 61% | 80% | 82% | 85% | 80% | 71% | 88% | 84% | 67% | 80% | 5 |
| 8 | 61% | 78% | 74% | 81% | 73% | 65% | 80% | 90% | 61% | 75% | 5 |
| 9 | 63% | 74% | 68% | 81% | 71% | 68% | 76% | 88% | 59% | 76% | 4 |
| 10 | 65% | 77% | 72% | 84% | 76% | 73% | 81% | 82% | 64% | 79% | 4 |
| 11 | 65% | 73% | 69% | 82% | 73% | 74% | 79% | 81% | 60% | 81% | 3 |
| 12 | 62% | 73% | 73% | 82% | 73% | 71% | 80% | 85% | 58% | 81% | 4 |
| 13 | 56% | 65% | 63% | 71% | 65% | 63% | 70% | 81% | 54% | 70% | 4 |
| 14 | 58% | 67% | 65% | 74% | 66% | 64% | 72% | 86% | 53% | 72% | 4 |
| 15 | 51% | 56% | 56% | 62% | 61% | 61% | 63% | 68% | 54% | 62% | 2 |
| S1 | 61% | 68% | 67% | 76% | 73% | 73% | 76% | 70% | 63% | 76% | 2 |
| Min | 51.07% | 56.26% | 56.20% | 62.15% | 60.79% | 61.33% | 63.03% | 68.30% | 50.12% | 62.07% | 2 |
| Max | 65.43% | 80.11% | 82.46% | 84.70% | 80.06% | 75.27% | 87.56% | 91.35% | 67.28% | 83.05% | 5 |
| Avg | 60.59% | 71.44% | 69.56% | 78.60% | 70.84% | 68.13% | 76.91% | 83.75% | 58.02% | 76.19% | 4 |

Table 3: Total Similarity Scores. (Similarity Chart.xlsx)

The Total Similarity Scores and number of breached cases (i.e., red cells) for all 15 weighting trials, plus the first Solver experiment ("S1"). The minimum, maximum, and average of each column are also included, as they were relevant in explaining the details I observed.

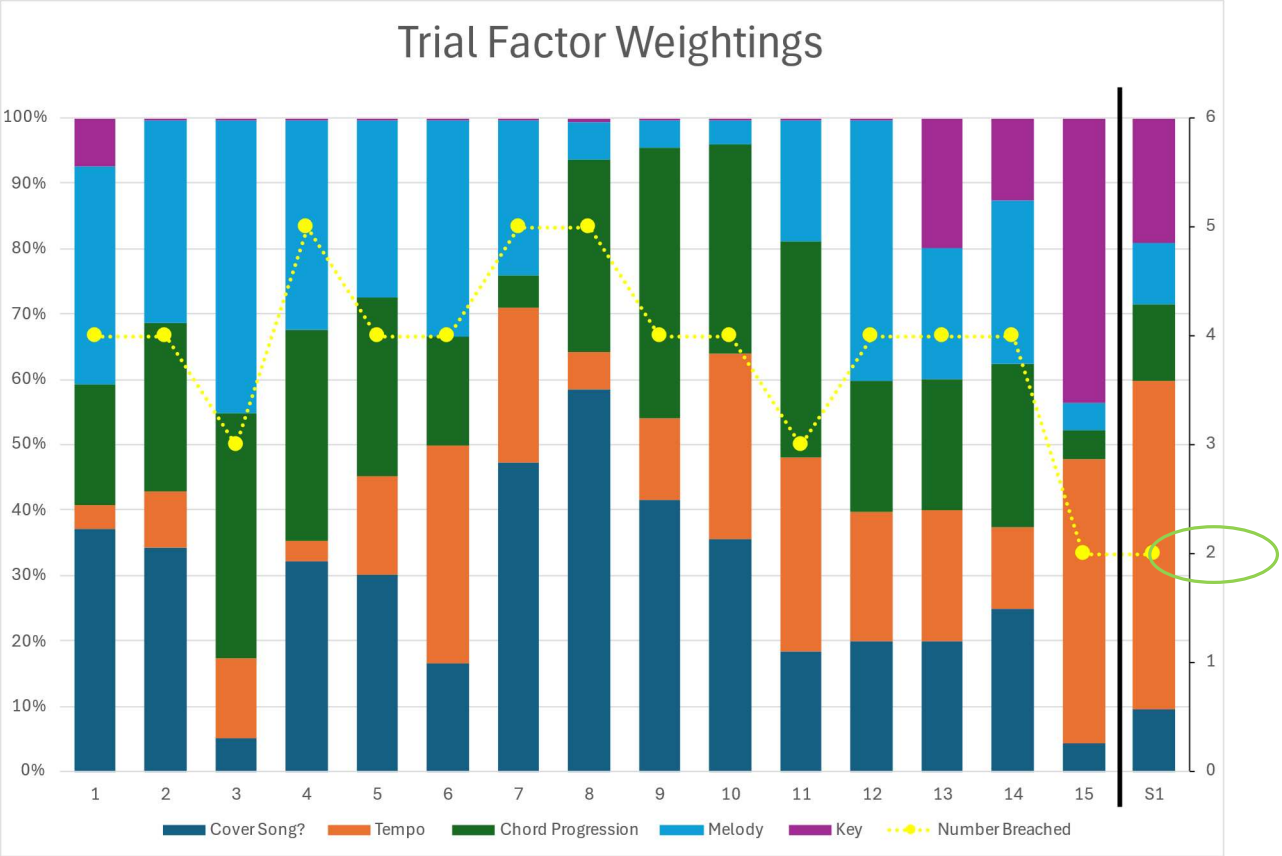



Figure 4: Trial Factor Weights. (Similarity Chart.xlsx)

The trial weighting distributions for all 15 trials, plus the first Solver experiment ("S1"). The breach count is represented by the yellow line. Note that Trial 15 and Solver 1, both emphasizing Tempo and Key weights, resulted in the lowest breaches by a significant margin.

## Numerical Statistics

- Across all trials, the number of breaches ranged from 2 to 5, with an average of 3.8125 (rounded up to 4).
- Most trials (Numbers 1, 2, 5, 6, 9, 10, 12, 13, and 14) resulted in the average number of breaches (4). Interestingly, while the weightings for these trials varied widely, the result was identical to that of Trial 13, which weighted all Factors equally at 20%.
- Trial 1 (the original distribution based on my intuitive assessment of relative importance) originally resulted in 6 breaches with Infringement Threshold = 70%,

(notably the only trial that yielded such a high number), but changing the Infringement Threshold value to 72% caused Trial 1 to only have 4 breaches like many of the others:

$$x_C \approx 37.04\%$$

$$x_T \approx 3.70\%$$

$$x_H \approx 18.52\%$$

$$x_M \approx 33.33\%$$

$$x_K \approx 7.41\%$$

- The Factor weights in Trial 15 resulted in only 2 breaches with Infringement Threshold = 58%, the lowest of the manually derived Factor weightings:

$$x_C \approx 4.35\%$$

$$x_T \approx 43.48\%$$

$$x_H \approx 4.35\%$$

$$x_M \approx 4.35\%$$

$$x_K \approx 43.48\%$$

- This 2-breach result is matched only by the "Solver 1" trial, whose Infringement Threshold is set to 69%:

$$x_C \approx 9.58\%$$

$$x_T \approx 59.24\%$$

$$x_H \approx 11.76\%$$

$$x_M \approx 9.41\%$$

$$x_K \approx 19.01\%$$

- As shown above, more than one Factor weight distribution can minimize the breach count, provided the Infringement Threshold is adjusted accordingly.

**Outliers (and Consistent "Non-Outliers")**

My study contains two major outliers that are consistently breached across all trials: *Marcus Gray, et al. v. Katy Perry* and *Bright Tunes v. Harrisongs.*

*Gray v. Perry* was ruled in favor of the defendant (not liable), yet there were no successful trials in which its Total Similarity Score fell below the estimated Infringement Threshold while settled and liable verdict cases remained above. Similarly, *Bright Tunes Music v. Harrisongs Music* was ruled in favor of the plaintiff (i.e., infringement), yet its Total Similarity Score consistently fell below the Infringement Threshold regardless of Factor weights. Removing these outliers would yield zero breaches in Trial 15 and Solver 1 (which emphasize the Key and Tempo Factors), and just 1 breach in Trial 3 (which emphasizes the Harmonic and Melodic Factors). The maximum breach count would be 3, and the average 2.

Why might these outliers exist? In the Katy Perry case, "Dark Horse" was originally adjudicated as *infringement* in 2019. The 2022 appeal overturned the initial ruling. **If not for this fact, "Dark Horse" would not be an outlier!** As for the Harrisongs case, recall that George Harrison admitted there were "striking similarities" between his and the plaintiff's songs. But 'similarity' does not necessarily mean 'infringement.' If Harrison's claim is true that any copying was purely unconscious and unintentional, it makes sense that a Total Similarity Score based on the structural music components would be lower compared to other verifiable infringement verdicts. Additionally, it is unclear which version of the Copyright Act was used in *Bright Tunes v. Harrisongs*. Assuming the court referenced the 1909 law (given the copyright and trial dates), this choice may have also affected the outcome. With access restricted to *sheet music* for comparison, a jury might have erred in assessing compositional differences if they lacked musical backgrounds. Moreover, they may have been unduly

influenced by (paid) 'expert witness' testimony. Listening to the *sound recordings* may have led to a different outcome. For both outliers, it is quite possible that the court simply made a mistake in their assessment.

In stark contrast to the outliers, there are four cases (or '*non-outliers*') that were consistently in line with my simulation (i.e., no breach regardless of Factor weightings). It is possible that the court judged these cases correctly, and that my system reflects it.

The first non-outlier is *Michael Skidmore v. Led Zeppelin*. As previously discussed, only the intro of "Stairway to Heaven" was similar to that of "Taurus." The songs are completely different otherwise so a 'liable' verdict may not be appropriate. The second example is *Pharrell Williams, et al. v. Bridgeport Music*. This case scored high in both Harmony and Melody similarity throughout the trials, so an assessment of infringement is reasonable. The third is *Tom Petty v. Sam Smith,* which settled before the trial verdict. Recalling possible reasons for a settlement, there may have been a desire to mitigate the financial impact of what could potentially be a liable verdict. With an average Total Similarity Score of just over 76%, that was likely a good choice. Finally, we have *Chuck Berry v. The Beach Boys,* which settled before going to trial. The artists readily admitted paying tribute to Berry's work and wisely agreed to compensate him for it. The average Total Similarity Score of nearly 77% suggests the financial impact of an unfavorable court verdict could have been much greater.

**The Role of Factor Weighting**

Recall from the statistics that most of the trial Factor weightings resulted in 4 breaches. Two of these, of course, were the outliers discussed above. The other two breached cases were the same in all trials: *Joe Satriani v. Christopher Martin, et al.* and *Queen & David Bowie v.*

*Vanilla Ice.* These were both settled outside of court, yet their Total Similarity Scores fell below the Infringement Threshold in most of the trials. Why might this be? Functionally, settling is the analog of receiving a 'liable' verdict because it still ends with the defendant paying the plaintiff a certain amount. The simulations, however, suggest the defendants may have received a favorable verdict had they allowed the cases to go to trial.

Ignoring outliers and 'non-outliers,' why is it that Factor weightings seemed to have little effect on the ability of the simulation to perfectly match the real-world verdicts? An interesting observation is that the Total Similarity Scores for *Satriani v. Coldplay* (70.84% on average) and *Queen & Bowie v. Ice* (68.13% on average), both settlements and considered infringement in this context, were lower on average than some of the non-infringement cases (for instance, *Structured v. Sheeran*'s average Total Similarity Score was 71.44%). It was therefore impossible to estimate an Infringement Threshold that falls below the scores for the Coldplay and Ice cases while simultaneously exceeding that of the Sheeran case. For this reason, breaches were unavoidable. The court may have erred in its assessment of the Sheeran case, or it could be that the others were too hasty in offering to settle.

It was explored earlier that perhaps Vanilla Ice stood a better than average chance in court based on simulation results. Looking closer, the obviously sampled bassline (a repeated 7-note motif) and similar tempo stand out because they are so prominent at the start of both songs. The question is: Might this be the only evidence of copying, and where would it fall between de minimis and the bright line rule? Is a 7-note, looped arrangement *protectable? "Ice Ice Baby"* is of a completely different genre (rap) than "Under Pressure" (pop/rock) and ignoring the common bassline, they are quite different—both in lyrics and in general style.

Would it have been ruled non-infringement for these reasons had the case gone to trial? Could the same hold true in the Coldplay case?

**Considering a Data Science Approach in the Real-World**

Is it possible that a simulation like this one could be used in court as valid evidence in assessing copyright infringement? Perhaps. The data analysis methodology carries many strengths, including:

- **A data science approach ensures consistency.** For any set of optimized Factor weightings and Infringement Threshold, the results are repeatable for any song pair regardless of court room, judge, jury, and number of times tested.
- **Results are based entirely on *intrinsic*, structural music components.** The simulation does not incorporate extrinsic factors based on vague, undefined principles, so cases would not be decided based on technicalities.
- **Used pre-emptively, a data science approach could theoretically settle potential infringement disputes *before* they become legal issues.** Artists could vet their new songs against a database of copyrighted works to determine the degree of originality before release—not unlike Grammarly, Turnitin, and other existing applications that detect plagiarism in writing. Depending upon the results, they could then choose to modify their work before risking a major copyright infringement dispute.

The weaknesses of such a system, however, must also be considered:

- **The values of the Factor weightings were not decided in a way that is 'definitive' or 'objectively correct.'** They were determined by comparison with subjective results (i.e., real-world verdicts). Much more testing and research would be necessary to

develop a comprehensive group of Factors and appropriate weightings to confidently detect copyright infringement.

- **The Infringement Thresholds used in this study were not exact or consistent.** In each trial, the Threshold was derived by minimizing the number of breaches for a given set of Factor similarity ratings. Consequently, they were also determined by comparison with subjective results.
- **There is no simple way to compare *fragments* of audio files; the Essentia algorithms analyze the entirety of the selected songs for similarity.** A system for use in the real-world must consider how to capture potential infringement that is *larger* than de minimis, but *smaller* than a full song. Appropriating part of another artist's song is still considered plagiarism and is subject to copyright enforcement.
- **There is no way to determine the 'protectability' of certain structural music components.** Certain chord progressions are considered commonplace or 'typical' within a particular genre and, in some cases, have been deemed 'not protectable.' While Ed Sheeran's case, for example, had an average Total Similarity Score of more than 71%, it was deemed non-infringement because the harmony in question was considered standard and widely used.

**Other General Observations**

Overall, the data and Factors to consider when creating a simulation of this type is not very clear or immediately obvious. Devising a comprehensive Factor weighting strategy and Infringement Threshold adds additional complexity. Finally, without clearly defined standards for 'de minimis,' 'protectability,' and 'similarity,' questions remain with respect to the applicability of this approach. Notwithstanding, the results contained in this study (outliers,

breaches, etc.) are still useful in observing the strengths and weaknesses of the current system, as discussed above.

This project unfortunately could not be as large-scale as I originally hoped when starting. Due to time constraints, I only used ten court cases (20 songs) and five of Essentia's algorithmic tools. It is typical and expected of any scientific study to use a large enough sample size to accurately represent a particular population. Inserting many more court cases into the study may have significantly improved my results and observations, but would have required more time. Moreover, adding other Essentia tools such as 'mood assessment' and timbral analysis may have provided greater depth and clarity to the results achieved.

# Conclusions

I entered this project wanting a clearer insight into how copyright infringement is determined, and (more specific to the Thesis) to explore if and how it could be objectively simulated. The results provided key takeaways that matched well with this main goal.

## Key Takeaways

The court's decisions are sometimes determined by factors that have little to do with the music itself. Examples of this are the Copyright Act chosen to govern proceedings and consideration for the character and intent of the alleged infringement. A system based on data analysis would focus only on objective Factors such as Tempo and Harmony. This approach represents a different paradigm that comparatively fosters many strengths and weaknesses when assessing infringement. It ensures greater consistency, yet it falls short of capturing all manners of infringement. More extensive programming, testing, and research is necessary to capture the many nuances associated with copyright enforcement, and to ensure fairness in adjudication. Ignoring the two largest outliers from the results of this study allowed the simulation to mimic real-world verdicts to a reasonable degree of accuracy. Factor weightings for at least two of the trials would have generated a perfect match without them. A data science-based system that includes a comprehensive set of weighted intrinsic Factors could theoretically be used as valid evidence in future court cases.

## Possible Questions for Future Research

As a reminder, these results are not meant to be definitive, but to spark further discussion. That said, this project raises a few important questions related to future work in this field of study:

**In the future, could we create a program, using any programming language, that automatically detects copyright infringement by taking every possible contextual Factor into account?**

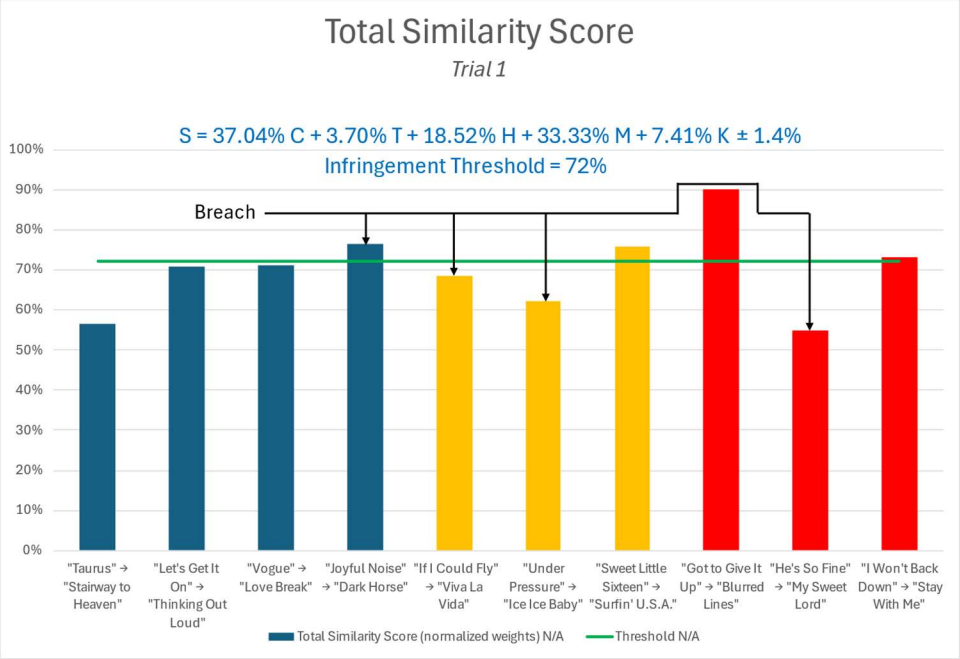
Artificial Intelligence (AI) technology is still in its infancy and is still prone to inaccurate or uncanny conclusions and results (e.g., Google AI overviews, ChatGPT, and AI image generators). Even if the answer to this question is "yes," human intervention would still be necessary to verify the AI response ('liable' or 'not liable'). AI systems must be trained on historical information. Given the inconsistencies highlighted in this work, finding reliable source information on which to train may prove problematic. There is also the need to 'objectify' subjective factors like character/intent and protectability, which means that objective standards should first be discussed and agreed upon. Finally, a system of this type must be continuously updated to keep pace with technological advancements. Addressing these issues would allow us to achieve and preserve fairness in the copyright enforcement system. If the answer to the question is "no," however, another question is worth consideration:

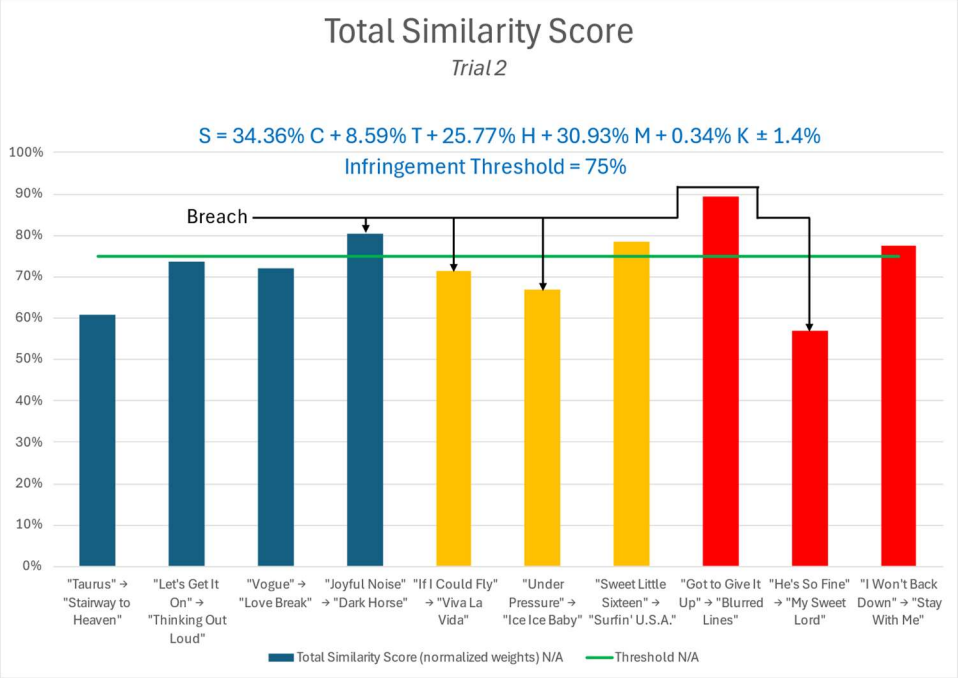
**Could we, without reliance on AI, refine music copyright law such that courts will adjudicate more consistently and "fairly"?**

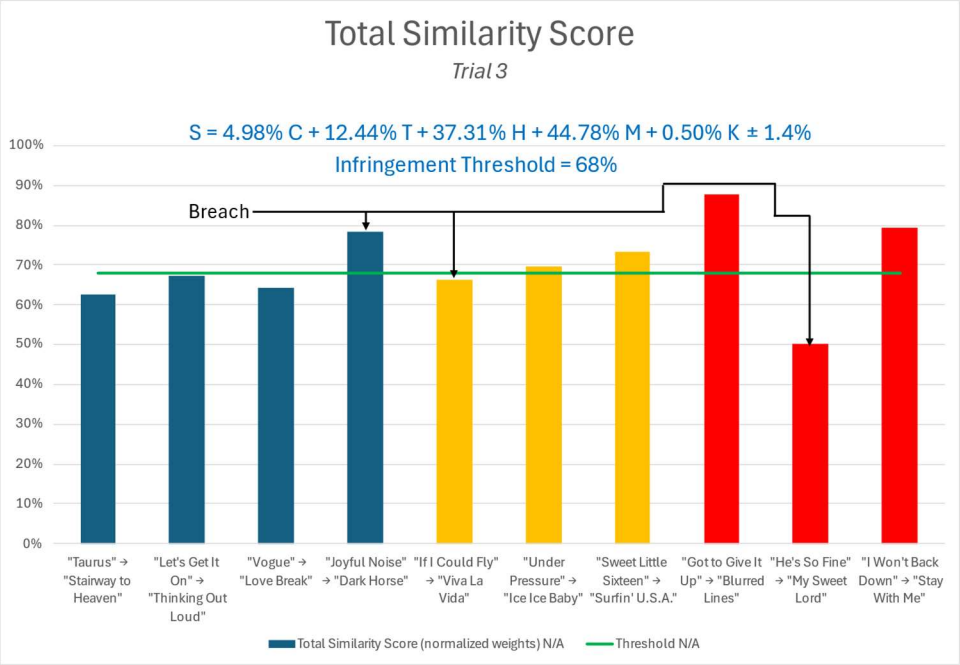
Recall that the 'bright line rule' and the 'de minimis' exception (in general, but especially with regards to digital sampling) directly contradict each other. Until our legal system clearly defines what is permissible for avoiding copyright infringement, there is no algorithmic solution that can satisfy both. My project illustrates the difficulty in setting an Infringement Threshold that clearly distinguishes between 'liable' and 'non-liable' cases. If the law is revised to be much more definitive, then future scholars could use my project as a basis for their own work, taking many more intrinsic Factors into consideration. We could then perhaps avoid further confusion about protectability, ownership, substantial similarity, de minimis, and other extrinsic considerations when assessing copyright infringement.

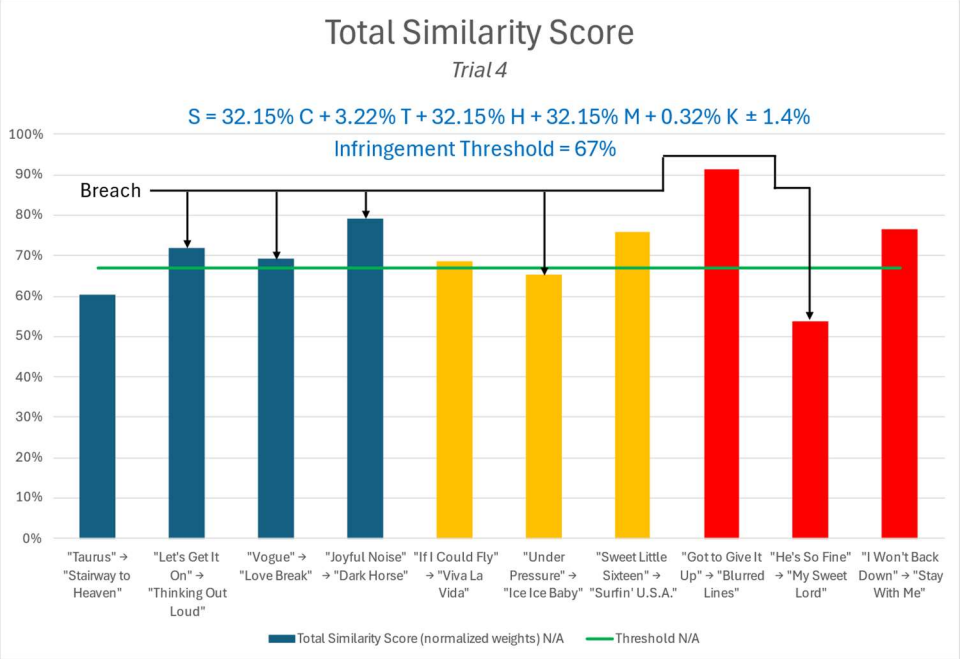
# Appendix A: Similarity Chart

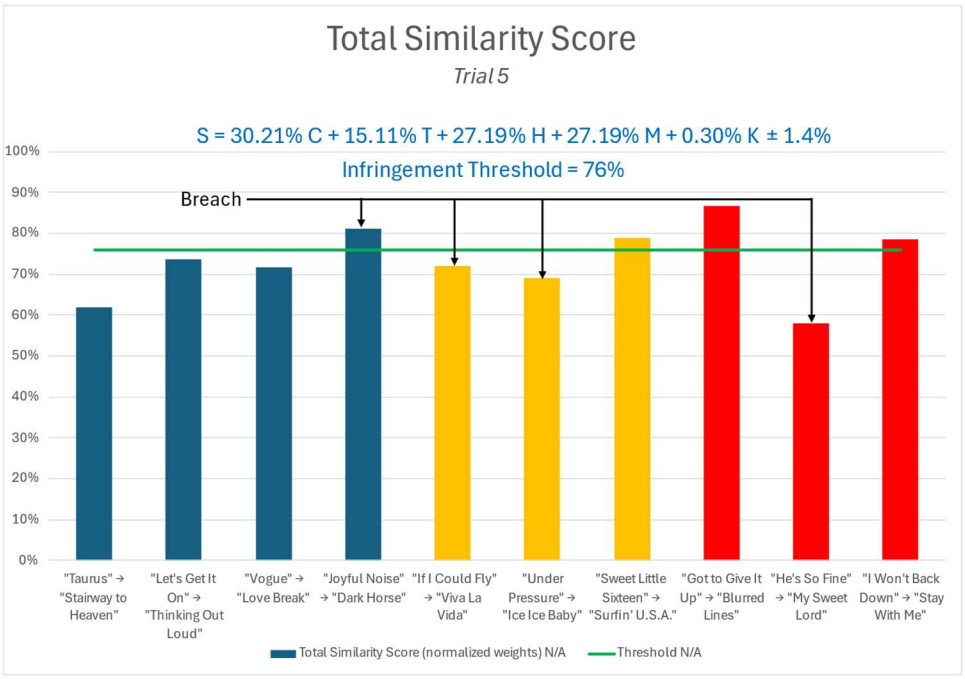
"Similarity Chart.xlsx" contains all eighteen conducted weight distribution Trials: the 15 main Trials and the 3 Solver Trials. The table shows information about each case: song titles, premise for lawsuit ("Similarity"), whether any copying was intentional, the verdict, and all calculated values. In addition to the attached file, the resulting graphs are shown below (Solvers 2 and 3 are omitted because Solver 1 is the most optimized). Also contained in the file is Table 1 (case info), Table 2 (the calculation of the margin of error), and Table 3 (the Total Similarity Scores).

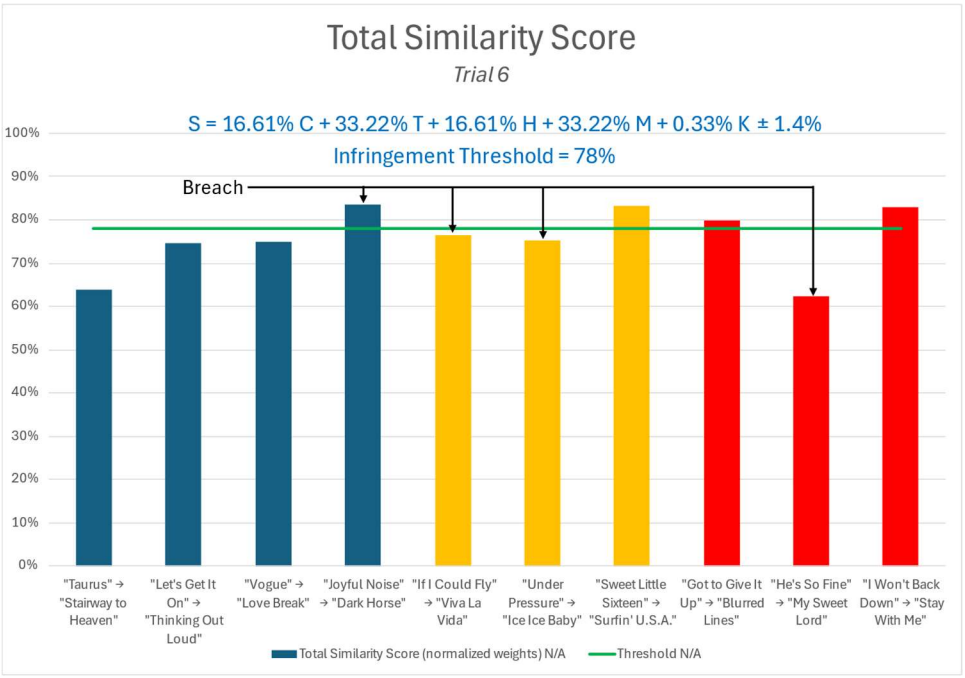

Total Similarity Score
Trial 1
S = 37.04% C + 3.70% T + 18.52% H + 33.33% M + 7.41% K ± 1.4%
Infringement Threshold = 72%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A

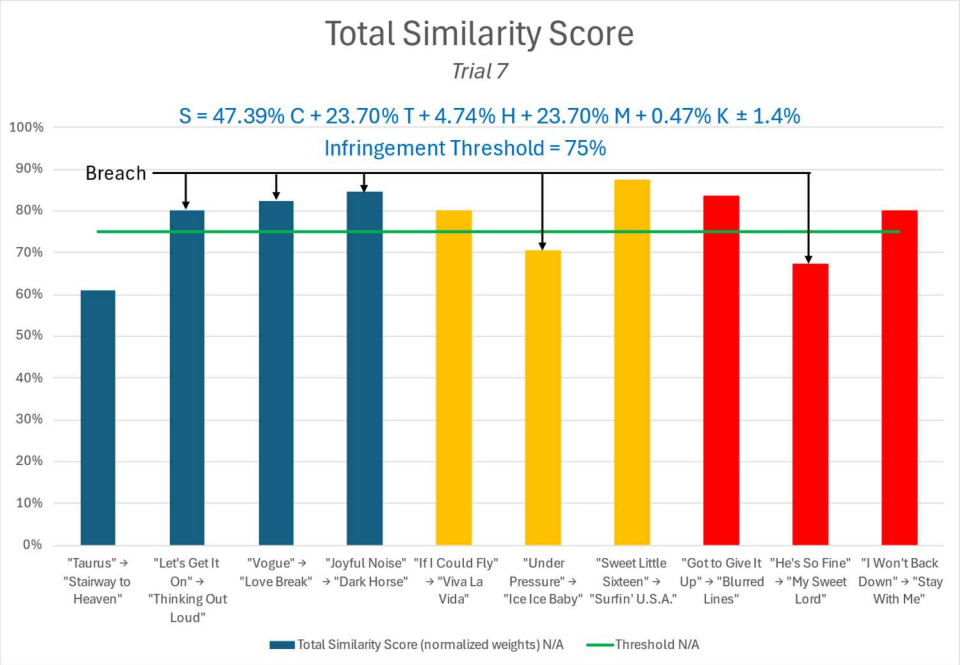

Total Similarity Score
Trial 2
S = 34.36% C + 8.59% T + 25.77% H + 30.93% M + 0.34% K ± 1.4%
Infringement Threshold = 75%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A

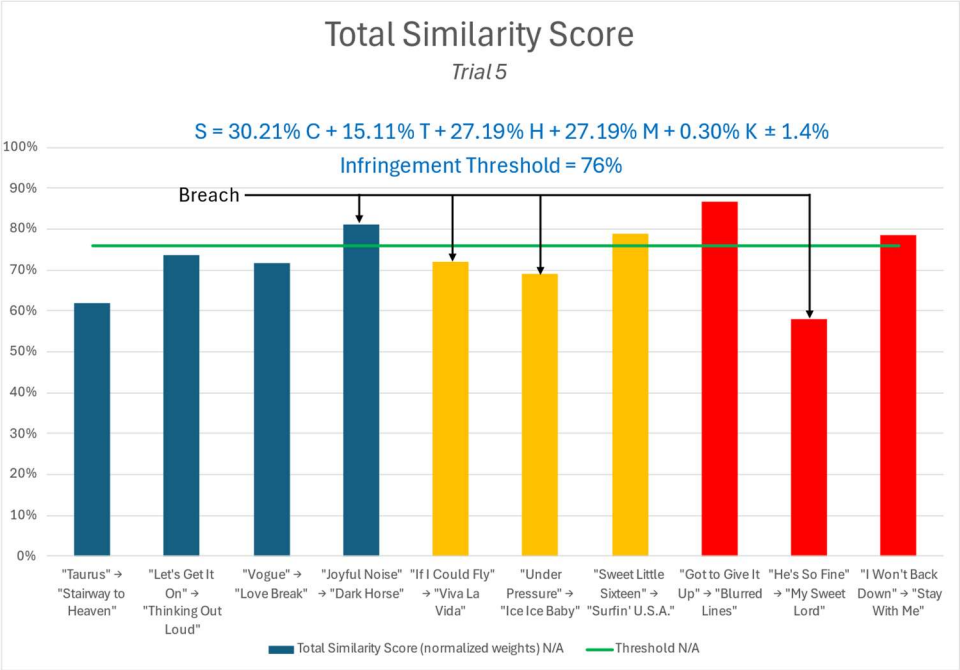

Total Similarity Score
Trial 3
S = 4.98% C + 12.44% T + 37.31% H + 44.78% M + 0.50% K ± 1.4%
Infringement Threshold = 68%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A

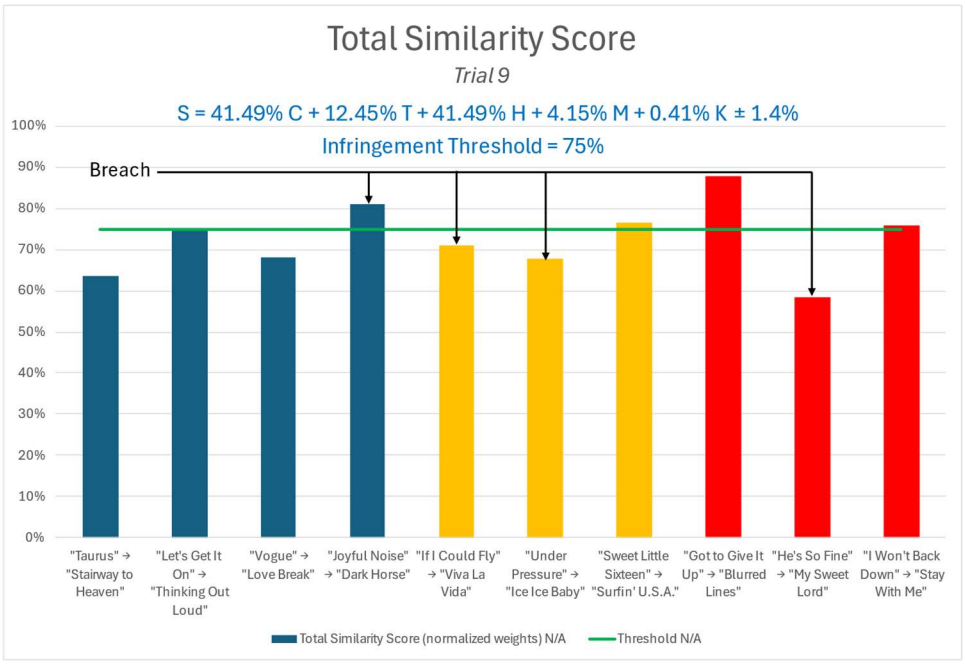

Total Similarity Score
Trial 4
S = 32.15% C + 3.22% T + 32.15% H + 32.15% M + 0.32% K ± 1.4%
Infringement Threshold = 67%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A

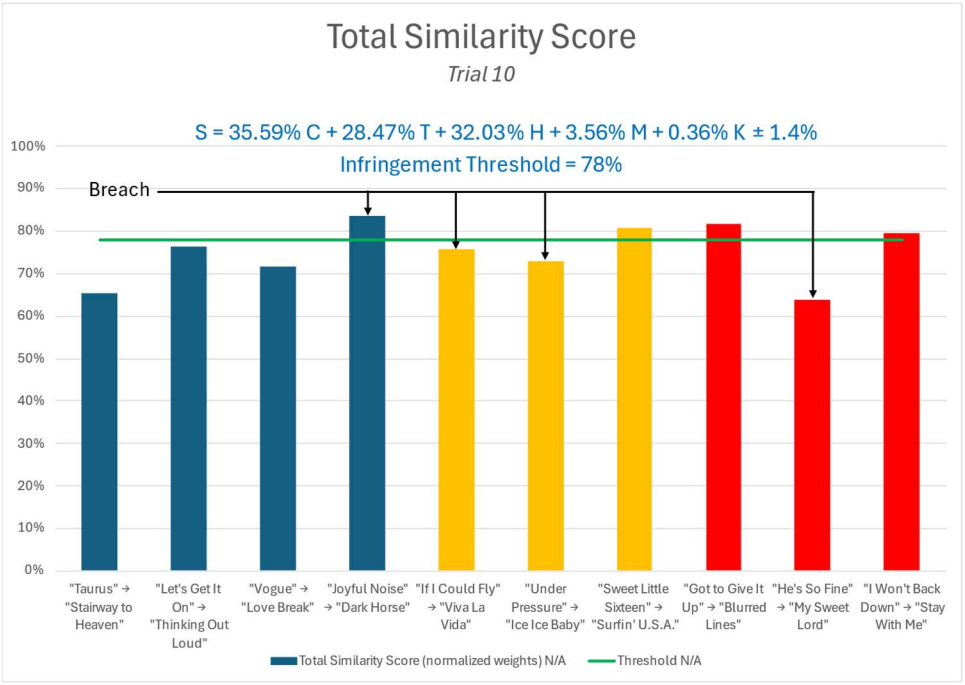

Total Similarity Score
Trial 5
S = 30.21% C + 15.11% T + 27.19% H + 27.19% M + 0.30% K ± 1.4%
Infringement Threshold = 76%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A

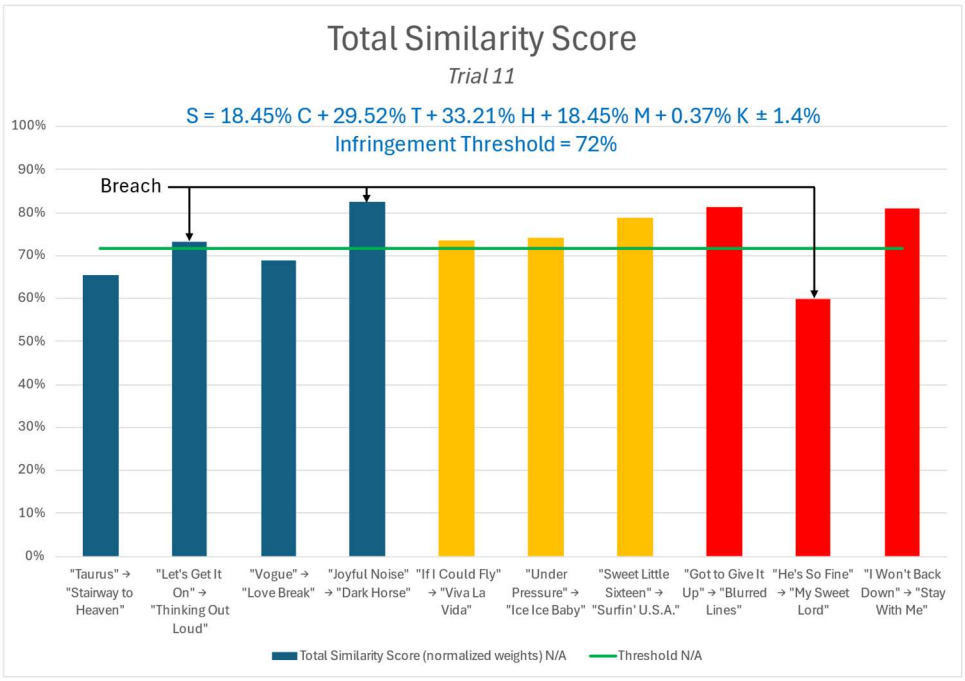

Total Similarity Score
Trial 6
S = 16.61% C + 33.22% T + 16.61% H + 33.22% M + 0.33% K ± 1.4%
Infringement Threshold = 78%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A

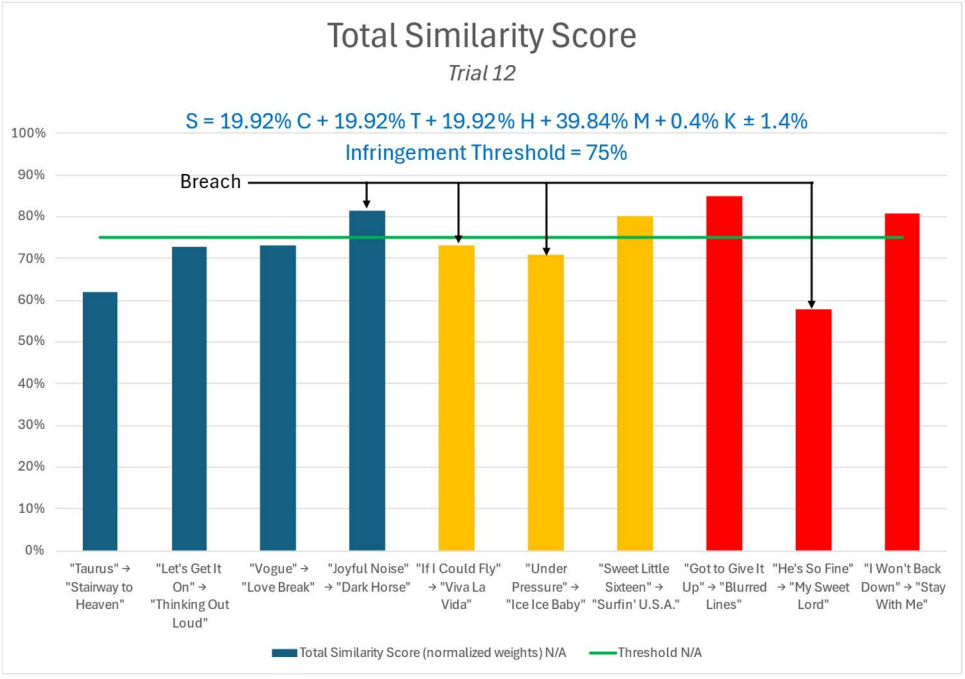

Total Similarity Score
Trial 7
S = 47.39% C + 23.70% T + 4.74% H + 23.70% M + 0.47% K ± 1.4%
Infringement Threshold = 75%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A

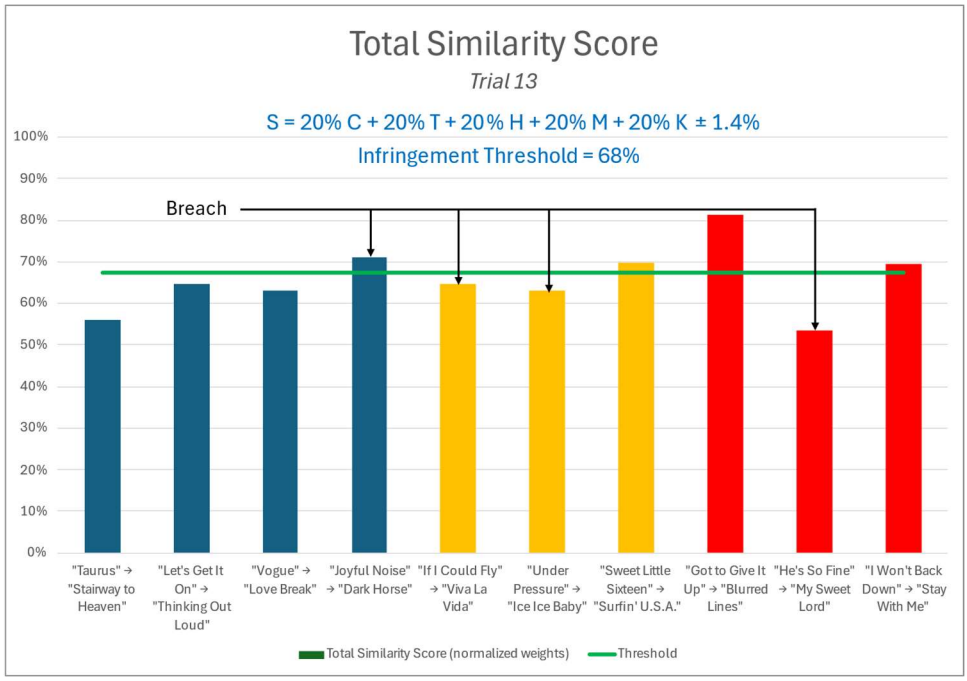

Total Similarity Score
Trial 5
S = 30.21% C + 15.11% T + 27.19% H + 27.19% M + 0.30% K ± 1.4%
Infringement Threshold = 76%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A

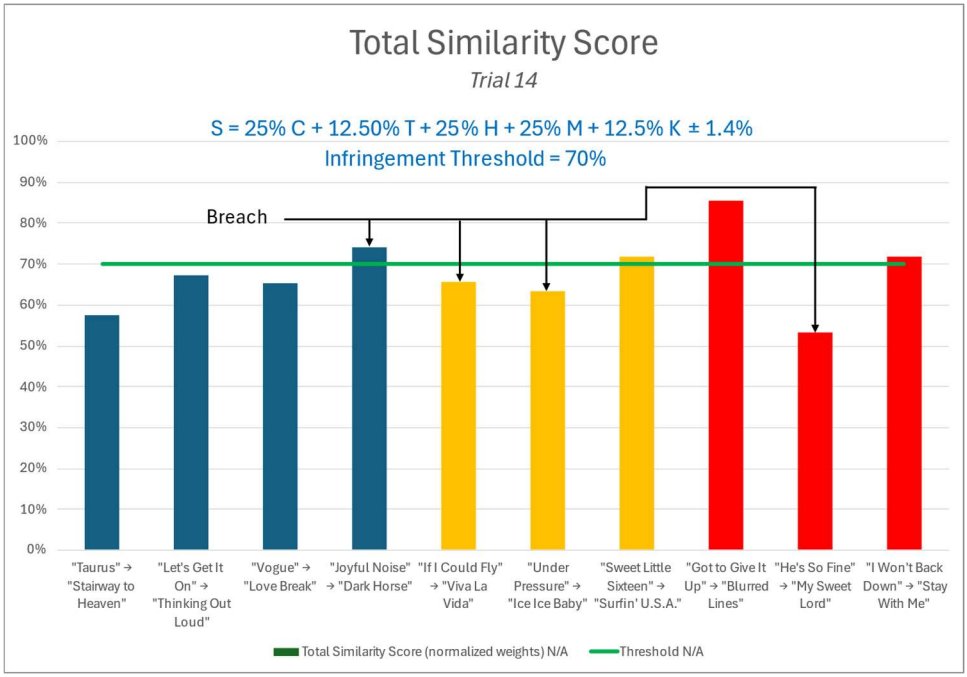

Total Similarity Score
Trial 9
S = 41.49% C + 12.45% T + 41.49% H + 4.15% M + 0.41% K ± 1.4%
Infringement Threshold = 75%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A

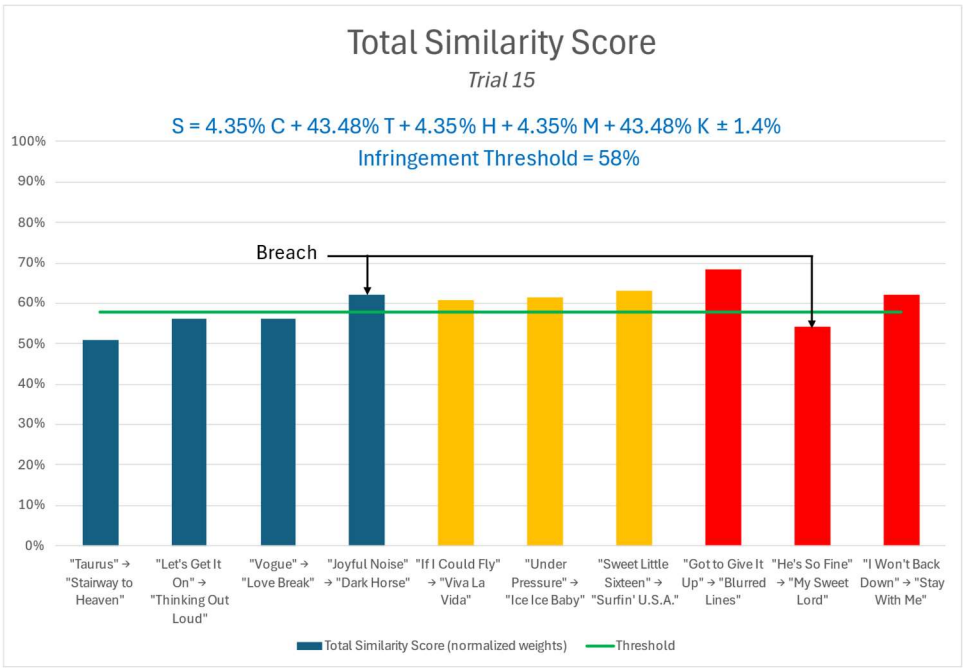

Total Similarity Score
Trial 10
S = 35.59% C + 28.47% T + 32.03% H + 3.56% M + 0.36% K ± 1.4%
Infringement Threshold = 78%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A

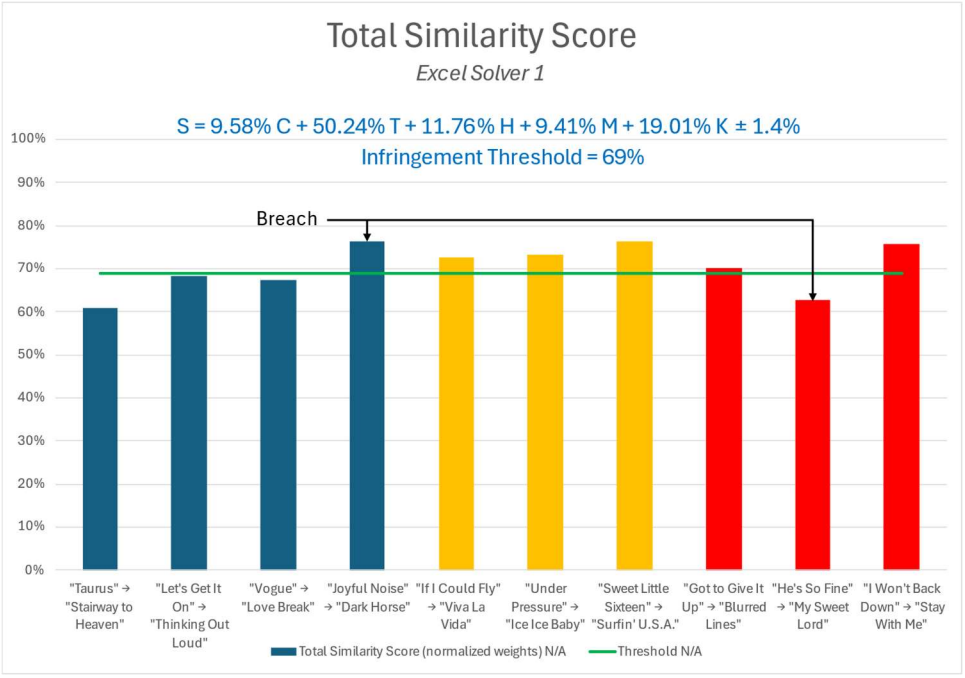

Total Similarity Score
Trial 11
S = 18.45% C + 29.52% T + 33.21% H + 18.45% M + 0.37% K ± 1.4%
Infringement Threshold = 72%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A


Total Similarity Score
Trial 12
S = 19.92% C + 19.92% T + 19.92% H + 39.84% M + 0.4% K ± 1.4%
Infringement Threshold = 75%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A


Total Similarity Score
Trial 13
S = 20% C + 20% T + 20% H + 20% M + 20% K ± 1.4%
Infringement Threshold = 68%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights)
Threshold


Total Similarity Score
Trial 14
S = 25% C + 12.50% T + 25% H + 25% M + 12.5% K ± 1.4%
Infringement Threshold = 70%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A


Total Similarity Score
Trial 15
S = 4.35% C + 43.48% T + 4.35% H + 4.35% M + 43.48% K ± 1.4%
Infringement Threshold = 58%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights)
Threshold


Total Similarity Score
Excel Solver 1
S = 9.58% C + 50.24% T + 11.76% H + 9.41% M + 19.01% K ± 1.4%
Infringement Threshold = 69%
Breach
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
"Taurus" → "Stairway to Heaven"
"Let's Get It On" → "Thinking Out Loud"
"Vogue" → "Love Break"
"Joyful Noise" → "Dark Horse"
"If I Could Fly" → "Viva La Vida"
"Under Pressure" → "Ice Ice Baby"
"Sweet Little Sixteen" → "Surfin' U.S.A."
"Got to Give It Up" → "Blurred Lines"
"He's So Fine" → "My Sweet Lord"
"I Won't Back Down" → "Stay With Me"
Total Similarity Score (normalized weights) N/A
Threshold N/A

# Appendix B: Jupyter Notebook files

The rest of the supporting materials are .ipynb files that were used to run the code, which are converted to PDF documents for accessibility. Each of the five "thesis_{…}" files uses one of the included "tutorial_{…}" files as a base. Various helper functions and constant variables are implemented to reduce repetition and improve readability. Visualizations were omitted from my files because they are not relevant to my project and may be difficult for audiences to read due to the lack of sufficient documentation. Depicted below are many snippets of the Python code. More technical details about the algorithmic process and outputs are defined in the Methods section of this document as well as in the code comments (denoted by '#'). The string "Done! No error!" is printed after each successful test.

## Cover Song Similarity

```
[1]: import essentia.standard as estd
     from essentia.pytools.spectral import hpcpgram
     import IPython as ip
```

```
[2]: # The folder I used for my work contains
         # a subfolder called "Thesis Music"
         # which contains all of the song files;
         # This function makes it much
         # easier to input the filenames
     def f(name):
         return f"Thesis Music/{name}.wav"

     # 2D array: 10 cases, each with 2 songs to compare (20 songs total)
     SONG_NAMES = [["16 Taurus", "4 Stairway to Heaven"],
                   ["Let's Get It On", "11 Thinking Out Loud"],
                   ["3 Vogue", "1 Love Break (Stephan Luke Remix)"],
                   ["16 Joyful Noise (feat. Lecrae & John Reilly)", "12 Dark Horse"],
                   ["6 If I Could Fly", "7 Viva La Vida"],
                   ["2 Under Pressure", "1 Ice Ice Baby"],
                   ["1 Sweet Little Sixteen", "4 Surfin' U.S.A."],
                   ["13 Got to Give It Up", "Blurred Lines (feat. T.I. & Pharrell)"], # < The longest one
                   ["1 He's So Fine", "2 My Sweet Lord (2014 Remaster)"],
                   ["2 I Won't Back Down", "3 Stay With Me"]]

     # For storing all values
     AUDIOS = []
     HPCP = []
     DISTANCES = []
     PAIR_CRP = []

     # Load the audio files
     for i in range(10):
         # Prints a number 0-9 after each iteration to denote progress
         print(i)
         query_audio = estd.MonoLoader(filename=f(SONG_NAMES[i][0]), sampleRate=44100)()
         accused_copy = estd.MonoLoader(filename=f(SONG_NAMES[i][1]), sampleRate=44100)()
         AUDIOS.append([query_audio, accused_copy])

     print("Done! No error!")
```

```
[3]: # Compute the HPCP chroma features of the audio signals
     for i in range(10):
         print(i)
         query_hpcp = hpcpgram(AUDIOS[i][0], sampleRate=44100)
         copy_hpcp = hpcpgram(AUDIOS[i][1], sampleRate=44100)
         HPCP.append([query_hpcp, copy_hpcp])

     print("Done! No error!")
```

```
[4]: # Can be used to create a cross-recurrent plot or CRP (see tutorial_similarity_cover.ipynb for visualizations)
     crp = estd.ChromaCrossSimilarity(frameStackSize=9,
                                      frameStackStride=1,
                                      binarizePercentile=0.095,
                                      oti=True)

     for i in range(10):
         print(i)
         pair_crp = crp(HPCP[i][0], HPCP[i][1])
         PAIR_CRP.append(pair_crp)

     print("Done! No error!")
```

```
[5]: # Computes the similarity distance value using the CRP
     for i in range(10):
         print(i)
         score_matrix, distance = estd.CoverSongSimilarity(disOnset=0.5,
                                                           disExtension=0.5,
                                                           alignmentType="serra09",
                                                           distanceType="asymmetric")(PAIR_CRP[i])
         DISTANCES.append(distance)

     print("Done! No error!")
```

```
[6]: # Compute the similarity rating based on the distance (rating = 1 - distance)
         # In Similarity Chart.xlsx, these values are converted to percentages
     for i in range(len(DISTANCES)):
         print(f"{SONG_NAMES[i][0]} v. {SONG_NAMES[i][1]}: {round(1 - DISTANCES[i], 4)}")
```

## Cover Song Similarity Error Testing

```
[1]: import essentia.standard as estd
     from essentia.pytools.spectral import hpcpgram
     import IPython as ip
```

```
[2]: def f(name):
         return f"Thesis Music/{name}.wav"

     # For the error tests, instead of using
         # a 2D array of the cases and songs,
         # I simply use a 1D array of length 20,
         # listing all of the songs
     """
     # 2D array: 10 cases, each with 2 songs to compare (20 songs total)
     SONG_NAMES = [["16 Taurus", "4 Stairway to Heaven"],
                   ["Let's Get It On", "11 Thinking Out Loud"],
                   ["3 Vogue", "1 Love Break (Stephan Luke Remix)"],
                   ["16 Joyful Noise (feat. Lecrae & John Reilly)", "12 Dark Horse"],
                   ["6 If I Could Fly", "7 Viva La Vida"],
                   ["2 Under Pressure", "1 Ice Ice Baby"],
                   ["1 Sweet Little Sixteen", "4 Surfin' U.S.A."],
                   ["13 Got to Give It Up", "Blurred Lines (feat. T.I. & Pharrell)"], # < The longest one
                   ["1 He's So Fine", "2 My Sweet Lord (2014 Remaster)"],
                   ["2 I Won't Back Down", "3 Stay With Me"]]
     """

     # 2D array: 10 cases, each with 2 songs to compare (20 songs total)
     SONG_NAMES = ["16 Taurus", "4 Stairway to Heaven",
                   "Let's Get It On", "11 Thinking Out Loud",
                   "3 Vogue", "1 Love Break (Stephan Luke Remix)",
                   "16 Joyful Noise (feat. Lecrae & John Reilly)", "12 Dark Horse",
                   "6 If I Could Fly", "7 Viva La Vida",
                   "2 Under Pressure", "1 Ice Ice Baby",
                   "1 Sweet Little Sixteen", "4 Surfin' U.S.A.",
                   "13 Got to Give It Up", "Blurred Lines (feat. T.I. & Pharrell)", # < The longest one
                   "1 He's So Fine", "2 My Sweet Lord (2014 Remaster)",
                   "2 I Won't Back Down", "3 Stay With Me"]
```

```
AUDIOS = []
HPCP = []
DISTANCES = []
PAIR_CRP = []

# Run the same cover song ID algorithms but comparing each song against itself
for i in range(20):
    print(i)
    query_audio = estd.MonoLoader(filename=f(SONG_NAMES[i]), sampleRate=44100)()
    accused_copy = estd.MonoLoader(filename=f(SONG_NAMES[i]), sampleRate=44100)()
    AUDIOS.append([query_audio, accused_copy])

print("Done! No error!")
```

## Tempo Similarity

```
[1]: import essentia.standard as es
     import essentia
```

```
[2]: # The folder I used for my work contains
         # a subfolder called "Thesis Music"
         # which contains all of the song files;
         # This function makes it much
         # easier to input the filenames
     def f(name):
         return f"Thesis Music/{name}.wav"

     # Calculate the similarity between the tempos
     def bpm_ratio(bpm1, bpm2):
         return min(bpm1, bpm2) / max(bpm1, bpm2)

     # 2D array: 10 cases, each with 2 songs to compare (20 songs total)
     SONG_NAMES = [["16 Taurus", "4 Stairway to Heaven"],
                   ["Let's Get It On", "11 Thinking Out Loud"],
                   ["3 Vogue", "1 Love Break (Stephan Luke Remix)"],
                   ["16 Joyful Noise (feat. Lecrae & John Reilly)", "12 Dark Horse"],
                   ["6 If I Could Fly", "7 Viva La Vida"],
                   ["2 Under Pressure", "1 Ice Ice Baby"],
                   ["1 Sweet Little Sixteen", "4 Surfin' U.S.A."],
                   ["13 Got to Give It Up", "Blurred Lines (feat. T.I. & Pharrell)"], # < The longest one
                   ["1 He's So Fine", "2 My Sweet Lord (2014 Remaster)"],
                   ["2 I Won't Back Down", "3 Stay With Me"]]
```

```
[3]: for i in range(10):
         # Prints a number 0-9 after each iteration to denote progress
         print(i)

         # Load the audio files
         audio1 = es.MonoLoader(filename=f(SONG_NAMES[i][0]))()
         audio2 = es.MonoLoader(filename=f(SONG_NAMES[i][1]))()

         # Run the RhythmExtractor algorithm to extract the two tempos
         rhythm_extractor = es.RhythmExtractor2013(method="multifeature")
         bpm1, beats1, beats_confidence1, _1, beats_intervals1 = rhythm_extractor(audio1)
         bpm2, beats2, beats_confidence2, _2, beats_intervals2 = rhythm_extractor(audio2)

         # Ratio between smaller and larger tempo
         tempo_similarity = bpm_ratio(bpm1, bpm2)

         # Print the results
         print(f"{bpm1} ({SONG_NAMES[i][0]}) vs. {bpm2} ({SONG_NAMES[i][1]})")
         print(f"Tempo Similarity = {tempo_similarity}")
         print("~")

     print("Done! No error!")
```

## Chords Similarity & Key Match

[1]:
```
import essentia.streaming as ess
import essentia
import IPython as ip
from essentia.standard import ChordsDetection
```

[2]:
```
# 2D array: 10 cases, each with 2 songs to compare (20 songs total)
SONG_NAMES = [["16 Taurus", "4 Stairway to Heaven"],
              ["Let's Get It On", "11 Thinking Out Loud"],
              ["3 Vogue", "1 Love Break (Stephan Luke Remix)"],
              ["16 Joyful Noise (feat. Lecrae & John Reilly)", "12 Dark Horse"],
              ["6 If I Could Fly", "7 Viva La Vida"],
              ["2 Under Pressure", "1 Ice Ice Baby"],
              ["1 Sweet Little Sixteen", "4 Surfin' U.S.A."],
              ["13 Got to Give It Up", "Blurred Lines (feat. T.I. & Pharrell)"], # < The longest one
              ["1 He's So Fine", "2 My Sweet Lord (2014 Remaster)"],
              ["2 I Won't Back Down", "3 Stay With Me"]]
```

[3]:
```
for i in range(10):
    KEYS = []
    VALUES = []

    # Prints a number 0-9 after each iteration to denote progress
    print(i)

    # Because we are in streaming mode as opposed to standard mode
        # (which affects the loader code significantly),
        # I must use a nested for loop where
        # i = the case number and j = either song in each case
    for j in range(2):
        # Load the audio files
        file = SONG_NAMES[i][j]
        loader = ess.MonoLoader(filename=f(file))
        framecutter = ess.FrameCutter(frameSize=4096, hopSize=2048, silentFrames='noise')
        windowing = ess.Windowing(type='blackmanharris62')
        spectrum = ess.Spectrum()
        spectralpeaks = ess.SpectralPeaks(orderBy='magnitude',
                                          magnitudeThreshold=0.00001,
                                          minFrequency=20,
                                          maxFrequency=3500,
                                          maxPeaks=60)

        hpcp = ess.HPCP()
```

```
        ## Compute HPCP; useful for key match
        hpcp_key = ess.HPCP(size=36, # We will need higher resolution for Key estimation.
                            referenceFrequency=440, # Assume tuning frequency is 44100.
                            bandPreset=False,
                            minFrequency=20,
                            maxFrequency=3500,
                            weightType='cosine',
                            nonLinear=False,
                            windowSize=1.)

        key = ess.Key(profileType='edma', # Use profile for electronic music.
                      numHarmonics=4,
                      pcpSize=36,
                      slope=0.6,
                      usePolyphony=True,
                      useThreeChords=True)

        pool = essentia.Pool()

        # Connect streaming algorithms.
        loader.audio >> framecutter.signal
        framecutter.frame >> windowing.frame >> spectrum.frame
        spectrum.spectrum >> spectralpeaks.spectrum
        spectralpeaks.magnitudes >> hpcp.magnitudes
        spectralpeaks.frequencies >> hpcp.frequencies

        # These lines used for key match
        spectralpeaks.magnitudes >> hpcp_key.magnitudes
        spectralpeaks.frequencies >> hpcp_key.frequencies
        hpcp_key.hpcp >> key.pcp

        hpcp.hpcp >> (pool, 'tonal.hpcp')
        # These three lines used for key match
        key.key >> (pool, 'tonal.key_key')
        key.scale >> (pool, 'tonal.key_scale')
        key.strength >> (pool, 'tonal.key_strength')

        essentia.run(loader)
```

```
        # Output the estimated key and scale
        est_key = pool['tonal.key_key'] + " " + pool['tonal.key_scale']
        KEYS.append(est_key)

        # Estimate chords, transpose them to C, and get rid of duplicates using unique_list()
        chords, strength = ChordsDetection(hopSize=2048, windowSize=2)(pool['tonal.hpcp'])
        unique_chords = transpose(unique_list(chords), key_to_int[pool['tonal.key_key']])
        VALUES.append(unique_chords)

    # Print the results
    print("Keys:")
    print(f"{KEYS[0]} ({SONG_NAMES[i][0]}) vs. {KEYS[1]} ({SONG_NAMES[i][1]})")
    match = KEYS[0] == KEYS[1]
    print(f"Match = {match} ({int(match)})")
    print(f"CHORDS SIMILARITY BETWEEN {SONG_NAMES[i][0]} and {SONG_NAMES[i][1]}: {similarity(VALUES[0], VALUES[1])}")
    print("~")

print("Done! No error!")
```

## Melody Similarity

[1]:
```
import essentia.standard as es
import IPython as ip
%matplotlib inline
import matplotlib.pyplot as plt
from pylab import plot, show, figure, imshow
plt.rcParams["figure.figsize"] = (15, 6)

import numpy
```

[2]:
```
# The folder I used for my work contains
    # a subfolder called "Thesis Music"
    # which contains all of the song files;
    # This function makes it much
    # easier to input the filenames
def f(name):
    return f"Thesis Music/{name}.wav"

# 2D array: 10 cases, each with 2 songs to compare (20 songs total)
SONG_NAMES = [["16 Taurus", "4 Stairway to Heaven"],
              ["Let's Get It On", "11 Thinking Out Loud"],
              ["3 Vogue", "1 Love Break (Stephan Luke Remix)"],
              ["16 Joyful Noise (feat. Lecrae & John Reilly)", "12 Dark Horse"],
              ["6 If I Could Fly", "7 Viva La Vida"],
              ["2 Under Pressure", "1 Ice Ice Baby"],
              ["1 Sweet Little Sixteen", "4 Surfin' U.S.A."],
              ["13 Got to Give It Up", "Blurred Lines (feat. T.I. & Pharrell)"], # < The longest one
              ["1 He's So Fine", "2 My Sweet Lord (2014 Remaster)"],
              ["2 I Won't Back Down", "3 Stay With Me"]]
```

[3]:
```
for i in range(10):
    # Prints a number 0-9 after each iteration to denote progress
    print(i)
    file1 = SONG_NAMES[i][0]
    file2 = SONG_NAMES[i][1]

    # Load the audio files
    loader1 = es.EqloudLoader(filename=f(file1), sampleRate=44100)
    audio1 = loader1()

    loader2 = es.EqloudLoader(filename=f(file2), sampleRate=44100)
    audio2 = loader2()

    # Extract melody pitches from each song
    pitch_extractor1 = es.PredominantPitchMelodia(frameSize=2048, hopSize=128)
    pitch_values1, pitch_confidence1 = pitch_extractor1(audio1)

    pitch_extractor2 = es.PredominantPitchMelodia(frameSize=2048, hopSize=128)
    pitch_values2, pitch_confidence2 = pitch_extractor2(audio2)

    # Pitch is estimated on frames. Compute frame time positions.
    pitch_times1 = numpy.linspace(0.0, len(audio1) / 44100.0, len(pitch_values1))
    pitch_times2 = numpy.linspace(0.0, len(audio2) / 44100.0, len(pitch_values2))

    # Time series of pitches
    pitch_values1 = list(pitch_values1)
    pitch_values2 = list(pitch_values2)

    # Print the results
    print(f"Melody similarity between {file1} and {file2}: {similarity(pitch_values1, pitch_values2)}")
    print("~")

print("Done! No error!")
```

# Bibliography


**Resources for finding court cases**

Wayte, L. *Pay for Play: How the Music Industry Works, Where the Money Goes, and Why.* University of Oregon Libraries, 2023.

Sisario, B. "Ed Sheeran Wins Copyright Case Over Marvin Gaye's 'Let's Get It On.'" *The New York Times*, 4 May 2023. NYTimes.com, https://www.nytimes.com/2023/05/04/arts/music/ed-sheeran-marvin-gaye-copyright-trial- verdict.html.

Music Copyright Infringement Resource |. https://blogs.law.gwu.edu/mcir/.

**Court cases (descriptions found on GWU's Music Copyright Infringement Resource)**

Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 114 S. Ct. 1164, 127 L. Ed. 2d 500, 29 U.S.P.Q.2D (BNA) 1961, 62 U.S.L.W. 4169, Copy. L. Rep. (CCH) P27,222, 94 Cal. Daily Op. Service 1662, 94 Daily Journal DAR 2958, 22 Media L. Rep. 1353, 7 Fla. L. Weekly Fed. S 800 (U.S. Mar. 7, 1994)

The Beach Boys v. Chuck Berry (1963).

Bright Tunes Music v. Harrisongs Music. 420 F. Supp. 177 (S.D.N.Y. 1976)

Pharrell Williams, et al. v. Bridgeport Music, et al. 895 F.3d 1106 (9th Cir. 2018)

Michael Skidmore v. Led Zeppelin. 952 F.3d 1051 (9th Cir. 2020)

Structured Asset Sales LLC v. Sheeran, 2nd U.S. Circuit Court of Appeals, No. 23-905.

VMG Salsoul, LLC v. Madonna Louise Ciccone, et al. 824 F.3d 871 (9th Cir. 2016)

Queen & David Bowie v. Vanilla Ice

Joe Satriani v. Christopher Martin, et al. [Coldplay]. No. 08-7987 (C.D. Cal. Dec. 4, 2008)

Tom Petty v. Sam Smith.

Marcus Gray, et al. v. Katy Perry, et al.

**Plunderphonics**

Emmerson, S. *Music, electronic media and culture.* In Routledge eBooks, 2016. https://doi.org/10.4324/9781315596877

**Cover song identification (found in Essentia Python tutorials)**

Serra, J., Serra, X., & Andrzejak, R. G. (2009). Cross recurrence quantification for cover song identification. *New Journal of Physics.*

Serra, Joan, et al (2008). Chroma binary similarity and local alignment applied to cover song identification. IEEE Transactions on Audio, Speech, and Language Processing.

Serra, J., Gómez, E., & Herrera, P. (2008). Transposing chroma representations to a common key, IEEE Conference on The Use of Symbols to Represent Music and Multimedia Objects.