Music-induced Analgesia Genome Study (IMAGS)

Charlie Lantz, Jadon Lemkin, Dylan Serreyn, Sean Smith

March 6, 2026

Background:

The goal of our project was to explore the application of Music Information Retrieval (MIR) software to the *IMAGS framework* to understand what musical characteristics lead to pain reduction.

Due to various factors beyond our control, we were unable to utilize the existing IMAGS hardware and software in our project and instead developed a prototype independent of the existing system, with the intention of possible future integration by another student group. This, however, did limit our ability to explore the concepts within the stated goal, as we had to devote considerable time to creating a basic framework.

Musical Analysis:

To determine which musical attributes we should focus on attempting to extract from a song, we researched which musical attributes were connected to greater emotional responses from the listener. After researching, it was clear there was a big three: autonomic calming, attention capture, and emotional meaning.

Autonomic calming refers to shifting the patient into a parasympathetic state. Basically, shifting them from fight or flight mode to a calm mode. Most studies did this by playing slower music or music that was more laid back. When aiming to achieve a parasympathetic state, you should get the following results:

- heart rate slows
- breathing deepens
- muscles soften
- digestion resumes
- stress hormones drop
- pain sensitivity decreases
- $\uparrow$ vagus nerve activity

- ↑ acetylcholine
- ↓ cortisol / adrenaline
- ↑ endogenous opioids

Another study aimed to target a patient's natural rhythm. They did this by inviting patients in and asking them to tap a melody on the table with their finger. This instinctual tap they did was deemed as the patient's natural rhythm. They then shifted the music they played for the patient to a BPM that matched the patient's natural rhythm. This was a fascinating study that can be explored in future groups.[1]

A huge correlation between positive results in lower pain ratings was how much the music captured the patient's attention. The shift of focus from pain to what was playing blurred out the pain they were feeling. This brings many factors into play, such as the complexity of the song. If a song was found too complex or too simple for the user, they would not be as attentive to what they were hearing.[2] This also makes the familiarity with the song a factor, as people were much more attentive to a prior connection with the music. This leads into our next point: emotional meaning.

Emotional meaning was the biggest correlation in our research. Familiarity with the song stood out more than any analytical data on the song could. This is what makes the survey questions so vital to future progress. We joked multiple times that the song you have your first kiss to will trigger a much stronger emotional response than one you have never heard. Allowing the patient to have a say in the music they are listening to will also show stronger results. The engagement was much higher when they were allowed to show music rather than be shown music. Making this process easy for patients so they will be able to pick their songs is vital to positive progress.[3]

---

[1] Wenbo Yi et al., "Individualizing Musical Tempo to Spontaneous Rates Maximizes Music-Induced Hypoalgesia," *Pain* 166, no. 8 (2025): 1761–68, https://doi.org/10.1097/j.pain.0000000000003513.

[2] Claire Howlin et al., "Tune Out Pain: Agency and Active Engagement Predict Decreases in Pain Intensity After Music Listening," *PloS One* 17, no. 8 (2022): e271329, https://doi.org/10.1371/journal.pone.0271329.

[3] Xuejing Lu et al., "The Effect of Background Liked Music on Acute Pain Perception and Its Neural Correlates," *Human Brain Mapping* 44, no. 9 (2023): 3493–505, https://doi.org/10.1002/hbm.26293.

Hardware Development:

Our initial hope was to be provided a sensor from the existing project, but when we were unable to do so, we pivoted to creating our own. We began with the original design from the start of the IMAGS project, but iterated to what can be seen in Figure 1. It's relatively simple, with 3 main parts. First, a voltage divider (left) provides a reference voltage to the non-inverting input of the op-amp. Second, the inverting input is connected to the output via a $100k\Omega$ feedback resistor, and is also connected across the skin contacts to ground. This means that as your skin resistance changes, the op-amp will adjust the output to attempt to keep the differential voltage between the two inputs at 0. This output is then measured, and can be used to calculate the skin resistance value. The third part is an RC filter on the output to smooth the signal heading into the arduino.

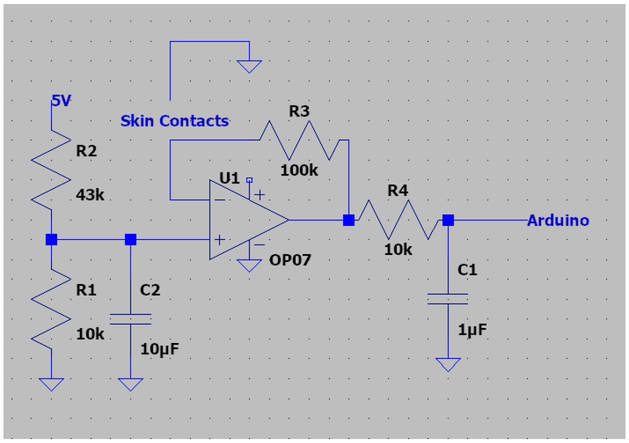

Figure 1: *GSR Circuit Diagram*

The skin resistance can be calculated based on the output voltage sent to the analog pin of the Arduino with the following equations:

$$V_{\text{ref}} = 5V \frac{R_2}{R_1 + R_2}$$

$$R_{\text{skin}} = \frac{V_{\text{ref}} \cdot R_3}{V_{\text{measured}} - V_{\text{ref}}}$$

Our testing often found skin resistance values in the range of $120k\Omega - 350k\Omega$, although the possible range of skin resistance is generally considered to be much larger[4] $(20k - 1M\Omega)$.

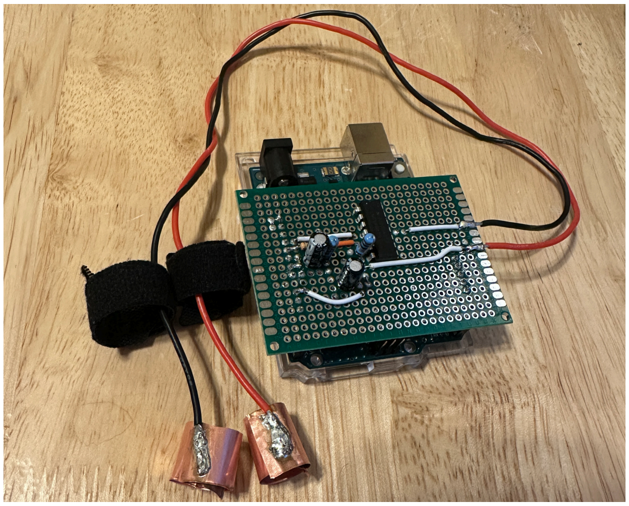

Figure 2: *GSR Sensor Attached to Arduino Uno*

Additional design improvements were focused on improving the physical robustness of the circuit to improve reliability. This involved upgrading the finger electrodes to be made of 0.05 mm copper sheet, with the leads soldered directly to them; as well as soldering the whole circuit on a piece of perfboard with header pins to mount on an Arduino Uno. Due to component availability, an LMC6484 Op-amp chip was used. This package contains 4 op-amps, and therefore could be substituted for a different unit containing only one.

[4]Teledyne Systems Corporation, *Electrical Resistance of the Skin* (1964), https://ntrs.nasa.gov/api/citations/19660021872/downloads/19660021872.pdf.

Software Development:

The software interface is organized into three python modules: gsr_reader.py, analyzer.py and app.py. Each module is responsible for handling different parts of the data pipeline: physiological sensing, audio analysis, and user interface. Together they provide an interface that synchronizes physiological data from the GSR with musical features.

gsr_reader.py:

This file defines a GSRStream class that manages communication with the Arduino micro controller. The class opens the selected serial port, reads the voltage measurements, timestamps each measurement, and stores them in a circular buffer. This uses the computer's system clock so it can later be aligned with the timescale of the audio analysis.

analyzer.py:

The analyzer.py module performs audio feature extraction using the Music Information Retrieval (MIR) library 'Essentia'. When an audio file is uploaded, this module converts it to stereo WAV at a consistent sample rate before running the analysis algorithms. These algorithms extract several key musical features including tempo, energy, momentary loudness (LUFS), and track duration. The main function, analyze_audio_bytes(), accepts the uploaded audio file and returns a structured dictionary containing global statistics and time-series feature data. The time series outputs are aligned to a shared timeline so they can be easily plotted with the GSR data stream.

app.py:

This module is the application user-interface implemented using Streamlit. It integrates the physiological data and the audio analysis and provides an interactive interface for the experiment. The app performs several functions:

1. Audio Analysis
   - User uploads an audio file
   - The file is passed to analyze_audio_bytes() from analyzer.py
   - Extracted features are formatted and stored for visualization
2. Playback Synchronization
   - When the user clicks "Start Playback", the application defines a time reference (t_sec = 0)
   - At the same time, GSRStream is initialized from gsr_reader.py

- User manually starts music playback from their phone or other (Audio playback not implemented)

3. Data Collection
   - While playback is active, the app periodically receives voltage samples from the GSR stream
   - These samples are converted into seconds relative to the playback start time
4. Visualization
   - The application plots Energy (from audio analysis) and GSR voltage on the same graph using a dual axis chart
   - Energy is displayed on the left axis while GSR voltage is on the right axis
   - The shared timescale of the two datasets allow for visual comparison between musical energy and physiological response
5. Automatic Termination
   - The application stops the GSR stream once the song duration has been reached
   - The final overlaid plot remains visible so the user can inspect the correlations between the two datasets

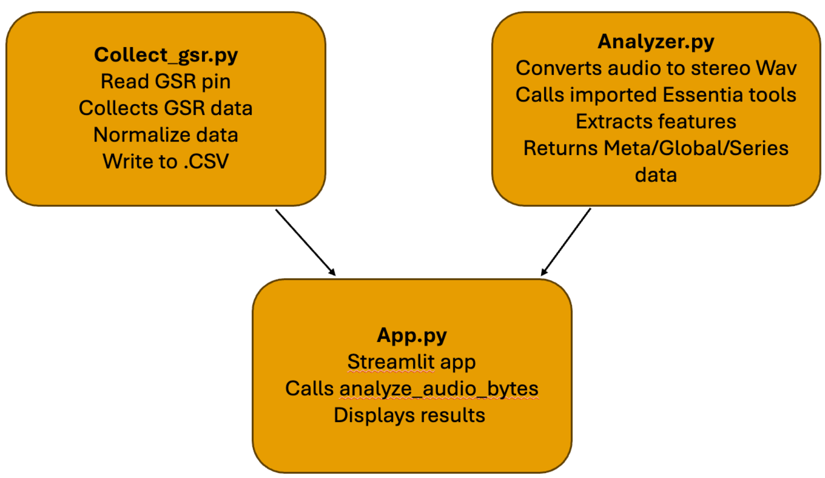

Figure 3: *Software Structure*

Data Collection:

We collected live data from the GSR through the Arduino's provided UART serial port. This was stored in RAM with a rolling buffer, which allowed us to keep a record of the past ~60 seconds of data, assuming an approximate data collection rate of 50hz.

Future research would benefit from writing chunked data to system storage to avoid overconsumption of memory when not performing live regression.

As data was recorded, timestamps were attached to provide us the exact time at which each datapoint was recorded, assuming insignificant UART latency. These timestamps, when compared to the timestamp at which the song was started, allowed us to map music information samples to GSR sensor samples throughout the recording. Where data did not align perfectly, we used bilinear interpolation to approximate values.

Music Information Retrieval:

To analyze the various properties of the music, we used the Essentia analysis library. This provided us both static and momentary data about the songs, including basic properties such as LUFs, key, and tempo, alongside more in-depth data such as the recognized sounds and sentiment analysis.

Recommended Future Work:

1. Integrate MIR and Analysis to Existing IMAGS Framework

The existing IMAGS framework contains numerous additional features and benefits as compared to the interface we created. This was intentional, as the purpose of this project was to focus on the music and data analysis. As such, the benefits of this project can be fully realized by integrating our work into the existing IMAGS system.

2. Expand User Feedback Survey

A major factor in how a user responds to music is their familiarity with the song, artist, genre, instruments, etc. Memories connected to certain songs can easily result in an emotional response based more on the memory than the characteristics of the music itself. As such, correlations between song characteristics and sensor data when a song is familiar become much less insightful. Expanding the survey questions to learn about the user's existing connections to the songs being played can provide an explanation for physiological responses that aren't consistent with a user's musical preferences. These questions are split for prior and after treatment. Prior questions include questions for the patient and environmental questions for GSR accuracy.

Pre-GSR Questions:

- Room Temperature
- Time of day
- Has the patient had any caffeine today?
- Is the patient on medication?
- When was the patient's last meal?
- Has the patient been in any recent physical activity?

Prior Patient Questions:

- Rate your pain level right now on a scale of 0-10.
- What has been your average pain level today on a scale of 0-10?
- How big is music in your life on a scale of 0-10?
- What are your favorite genres?
- Who are your favorite artists?
- Are there any songs you would like to play?

Post Patient Questions:

- Rate your pain level right now on a scale of 0-10.
- Did your pain change noticeably at all during the music?
  - Decreased significantly
  - Decreased slightly
  - No change
  - Increased slightly
  - Increased significantly
- Did any songs stand out from a previous experience?
  - If yes, select them:
- How much did you enjoy song X? … Y? … etc.

3. Expand Music Analysis

Refer to our past research when considering what new technical features of the music would be best to be analyzed for the goal you are trying to achieve. Such as familiarity with the song or looking to achieve a calmer reaction out of a patient. The more data we can get, the better we will be equipped to draw correlations between patient response and music features.

4. Explore Integration of Apple Watch for Additional Physical Sensing

When considering what sensor to use for our development of the project, we briefly looked at the feasibility of using the data available from an Apple Watch, as they contain reliable, high-end sensors. However, we learned that to access Apple Watch data, you must have an app running locally on the paired iPhone, which we determined the development of would be out of scope for our project. With that being said, the development of a simple iPhone app to gather this data and send it to the existing IMAGS framework could certainly be a standalone project. The current generation of Apple Watch, the Series 11, includes an electrical and optical heart sensor, as well as blood oxygen and temperature sensing capabilities.

Bibliography

Howlin, Claire, Alison Stapleton, and Brendan Rooney. "Tune Out Pain: Agency and Active Engagement Predict Decreases in Pain Intensity After Music Listening." *PloS One* 17, no. 8 (2022): e271329. https://doi.org/10.1371/journal.pone.0271329.

Lu, Xuejing, Xin Hou, Libo Zhang, et al. "The Effect of Background Liked Music on Acute Pain Perception and Its Neural Correlates." *Human Brain Mapping* 44, no. 9 (2023): 3493–505. https://doi.org/10.1002/hbm.26293.

Teledyne Systems Corporation. *Electrical Resistance of the Skin*. 1964. https://ntrs.nasa.gov/api/citations/19660021872/downloads/19660021872.pdf.

Yi, Wenbo, Caroline Palmer, Angela Serian, and Mathieu Roy. "Individualizing Musical Tempo to Spontaneous Rates Maximizes Music-Induced Hypoalgesia." *Pain* 166, no. 8 (2025): 1761–68. https://doi.org/10.1097/j.pain.0000000000003513.