Internship Report – Summer 2025

Abstract

This report summarizes my work as a Machine Learning Engineering Intern at CytoHub Inc. during Summer 2025, where I developed a pipeline to detect arrhythmia in cardiac organoids based on contraction data extracted from videos. The project involved preprocessing noisy biological signals, engineering time-series features, training an XGBoost model using labeled ECG data, and applying the model to unlabeled organoid data divided into uniform intervals. My pipeline identified key heart rate variability (HRV) features from peak-to-peak timing and generated arrhythmia probabilities over time. I implemented custom logic to handle missing or corrupted data and visualized the results across numerous organoids and days of data collection. The tools I developed can be integrated into CytoHub's broader analysis platform and be used to assess organoid health at a larger scale. This internship deepened my technical skills in ML and learning to process data while also teaching me how to work collaboratively in a cross-functional, research-driven environment.

Phase 1: Initial Exploration

Before building any models or extracting features, I began by developing a foundational understanding of both cardiac rhythms and the tools available for analyzing organoid behavior.

To evaluate the best way to classify organoid behavior, I began testing various machine learning models on early-stage synthetic and benchmark datasets. I experimented with XGBoost, random forest, and k-nearest neighbors (k-NN) models, observing how they handled different types of input and behaved under various conditions. This phase was critical for understanding how model outputs could vary, what kind of data preparation might be necessary, and how to tune parameters effectively.

I then spent time learning how MuscleMotion works: how it processes time-lapse videos of contracting organoids and extracts motion data from pixel-level changes over time. Each well produced several outputs, including speed-of-contraction.txt and contraction.txt. After analyzing the signals in each, I found that contraction.txt provided the clearest peak-based patterns and sufficient data to find and analyze peak-to-peak intervals. These contraction traces closely resembled heartbeat-like signals and became the core input for downstream analysis. This exploration also helped me understand edge cases (such as inactive organoids or noisy outputs) and highlighted the need for preprocessing steps later.

Finally, I researched about the heart and its various behaviors in correlation to MuscleMotion data. A heartbeat corresponds to a peak in contraction data, and normal rhythms exhibit evenly spaced, consistent peaks. In contrast, arrhythmic signals show irregular spacing, missed peaks, or erratic fluctuations. By reviewing both statistical and visual examples, I became familiar with contraction patterns and began comparing these with motion signals derived from organoid data. This exploration helped me understand the structure and function of a heartbeat, grounding my work in the biological behavior we were aiming to detect.

Phase 2: Predicting the Yield of an Organoid

In this phase, I worked on building a regression model to predict organoid yield based on biological and sequencing-derived features. The process involved five key steps: processing the input data, generating features, splitting the data, training a model, and evaluating its performance.

I began by loading the single-cell RNA sequencing data using Scanpy, processing matrix.mtx, barcodes.tsv, and features.tsv files into an AnnData object. I filtered out low-quality cells and genes, performed normalization, and clustered the cells using the Leiden algorithm. To characterize the resulting cell populations, I identified key marker genes for each cluster.

Next, I generated meaningful features for modeling. These included the percentage of cells per cluster and statistics derived from top marker genes. I merged these with additional features from the metrics.csv file produced by the cellranger_count pipeline to compile a complete feature set.

To prepare for modeling, I split the full dataset into training and testing sets using train_test_split from scikit-learn. This ensured that the model would be evaluated on unseen data, allowing for a more accurate assessment of its predictive performance.

I trained an XGBoost regression model, a gradient boosting algorithm well-suited for tabular data. The model incorporated both biological and sequencing-derived features, learning to predict organoid yield based on patterns in the data.

Finally, model performance was evaluated using Root Mean Squared Error (RMSE), which quantifies the average difference between predicted and actual cell counts. I found that incorporating more training data helped reduce RMSE, highlighting the importance of data quantity and quality in model performance.

Phase 3: Analyzing MuscleMotion Data

After selecting contraction.txt as the most informative output file, I began analyzing the signal patterns in detail to extract meaningful cardiac indicators from organoid contraction data.

The first step was to locate the peaks in the contraction signal, as each peak corresponds to a beat. I developed a script to analyze a given contraction.txt file and identify all peak locations, then calculate the intervals between each consecutive peak. These peak-to-peak intervals represent the time between beats, providing a direct way to measure consistency and rhythm. I visualized the distribution of these intervals using a histogram, which helped reveal how frequently each interval length occurred. This process served as the foundation for further heartbeat-level analysis.

To understand how heart activity varies over time, I then divided each organoid signal into consecutive 2000 ms intervals and computed the heartbeats per minute (BPM) within each window. This approach allowed me to track changes in contraction rate across the duration of the video. I then generated a time-series graph showing the BPM for each interval, offering insight into how steady or variable the organoid's heartbeat was. This visualization was useful for spotting patterns like sudden drops, spikes, or gradual drifts in heart rate.

Together, these analyses enabled dynamic monitoring of cardiac behavior in organoids, setting the stage for feature extraction and classification in later phases.

Phase 4: Detecting Arrhythmia Using Machine Learning

With a strong understanding of cardiac patterns and the ability to extract heartbeat-level features from organoid data, I began to build a model capable of detecting arrhythmia.

I selected the MIT-BIH Arrhythmia Database, a widely accepted benchmark for arrhythmia detection. It contains ECG readings from 47 patients, recorded between 1975 and 1979, with each session lasting 30 minutes. This dataset includes annotations for each beat, such as whether it's normal or arrhythmic. Importantly, it uses a consistent sampling rate (360 samples/second), so the sample number acts as a timestamp, helping align events across the signal.

I began by uploading the dataset and selecting key columns: time, sample number, and arrhythmic labels. After cleaning the data and ensuring consistency, I divided each patient's ECG signal into 5000 ms intervals and extracted heart rate variability (HRV) features from each segment. These features included the average time between beats, standard deviation of intervals, minimum and maximum values, RMSSD (root mean square of successive differences), and pNN50 (proportion of intervals that differ by more than 50 ms). Each interval was labeled as either arrhythmic or not based on the dataset annotations. The data was split into training and test sets (80/20 split), and I trained an XGBoost classifier to evaluate accuracy.

To identify the best-performing model, I compared XGBoost, random forest, and logistic regression using the same HRV features and multiple interval sizes (4000, 5000, and 6000 ms). XGBoost with 5000 ms intervals achieved the highest accuracy of 86.1%, making it the optimal choice for application to organoid data.

I applied the trained XGBoost classifier to organoid data extracted from MuscleMotion. Each contraction signal was split into 2500 ms intervals, and the same HRV features were computed. The model then predicted whether each interval was arrhythmic or not. The output was visualized as a heatmap showing the predicted probability of arrhythmia for each organoid well, allowing researchers to quickly spot high-risk or inactive regions.

Then, for a single organoid well (e.g., CP011, Well 001), I tracked how the predicted probability of arrhythmia changed over time. This helped identify trends such as whether an organoid's rhythm was stabilizing or becoming more irregular over the course of the experiment.

Finally, I extended the analysis to track arrhythmia probability across multiple wells on the plate. This allowed for cross-well comparisons and highlighted both temporal trends and spatial patterns in arrhythmia expression. The visualizations also accounted for inactive wells and cases where no data was available, providing a complete overview of organoid behavior over time.

Reflections and Takeaways

This internship was a deeply rewarding experience that challenged me to apply technical skills to real-world biological data. I gained hands-on experience with machine learning, time-series analysis, and signal processing, while also learning how to build scalable, end-to-end systems for a cross-disciplinary team.

Working closely with engineers, researchers, and product managers, I strengthened my communication skills by explaining technical decisions to diverse audiences. Weekly check-ins with the VP of Engineering taught me how to prioritize, take feedback constructively, and align with broader goals.

My time at CytoHub was a powerful introduction to the intersection of AI and biology. I'm proud that the tools I developed are helping advance arrhythmia research in organoids, and I'm excited to continue pursuing impactful work at this intersection.