AGTD - The AudioGene Translational Dashboard; A Hybrid Machine Learning and Visualization Interface for Genetic Diagnosis of Autosomal Dominant Non-Syndromic Hearing Loss

by

Benjamin DeSollar

A thesis submitted in partial fulfillment
of the requirements for the Master of Science
degree in Electrical and Computer Engineering in the
Graduate College of
The University of Iowa

May 2024

Thesis Committee: Thomas Casavant, Thesis Supervisor
Kishlay Jha
Terry Braun

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Tom Casavant, for his extraordinary patience and help in designing this tool and writing this thesis. I would also like to thank Dr. Richard Smith, Dr. Hela Azaiez, and Kevin Booth, whose expertise and insight into designing the tool were invaluable. Lastly, I would like to thank my family and friends, without whom I would not have been able to complete this body of work.

ABSTRACT

Autosomal Dominant Non-Syndromic Hearing Loss (ADNSHL) presents a major genetic diagnosis challenge due to the complex heterogeneity of the molecular basis of the disease – i.e., multiple distinct gene mutations. This thesis introduces a new approach to diagnosis that combines ensemble and semi-supervised Support Vector Machine (SVM) models with advanced visualization tools to improve the accuracy and confidence of ADNSHL diagnoses. Previous methods explored were AudioGene V4 (AG4), an MI-SVM Classifier, and AudioGene 9/9.1 (AG9 and AG9.1), an Ensemble Classifier, developed by Kyle Taylor (AG4), Chibuzo Nwakama (AG9), and Sean Ryan (AG9.1) respectively. These machine-learning algorithms were insufficient to combat the class imbalance and the lack of representation of the population within our training dataset. This led to poor accuracy within the MI-SVM and ensemble models, with 44% and 47% respectively in LOOCV. Therefore, using our new novel approach of integrating the ensemble and MI-SVM models with an advanced visualization dashboard as part of the AudioGene website. The aim is to enable clinicians to dynamically interact with the model predictions, allowing the clinician to see inside the "Black Box" of AudioGene models and help guide the clinicians in customizing the model to make better predictions. The visualization dashboard features an interactive three-dimensional audiogram plot, a 2-d audio profile plot, a surface profile viewer, and an ethnicity distribution visualization. Our evaluation through case studies reveals that AGDT can enhance diagnosis confidence and improve interpretation of AudioGene predictions. This thesis concludes that visualization tools are helpful in guiding the clinician in understanding AudioGene models and providing a promising advancement in ADNSHL diagnosis. More generally, it demonstrates the power of hybrid AI-human expert systems in delivering *explainable AI*.

PUBLIC ABSTRACT

This research explores a new strategy to tackle the diagnostic complexities and challenges of human deafness (ADNSHL). Previous methods explored using machine learning algorithms needed to be more robust to combat the class imbalance and the lack of representation of the population within our training dataset. This led to poor accuracy within the multi-audiogram-trained model (AG4) and the single audiogram-trained model (AG9/AG9.1), with 44% and 47%, respectively. Therefore, we developed a novel approach to integrate AG9.1 and AG4 with an advanced visualization dashboard. We developed this hybrid method to have the clinicians guide the model to make more accurate predictions and provide statistical significance within their diagnoses. The visualization dashboard features interactive three-dimensional audiogram plots, audio profile charts, surface profile viewers, and ethnicity distribution visualizations. Our evaluation through case studies reveals that AGDT can help in the diagnostic confidence and interpretability of AudioGene's genetic predictions. This thesis concludes that visualization tools are useful in assisting clinicians to guide the AudioGene models and provide promising advancements in ADNSHL diagnoses.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

The theory and practice of diagnostic medicine over the past two to three decades has been revolutionary, especially considering the advent of personalized and evidence-based medicine. The field has gradually shifted from a one-treatment fits all to a more personalized treatment of each individual. The availability of low-cost genome sequencing technologies has ushered in a new era of personalized medicine. Targeted treatments aim to tailor to individual genetic profiles, acknowledging that the best treatment for the majority is not necessarily the best for every individual. For example, conditions such as hearing loss can be attributed to genetic variations between individual patients, and using a treatment well-suited for the majority may not help every patient.

1.1. Genomic Diagnosis in Medicine

Accurate genetic diagnosis is the cornerstone of personalized genomic medicine, requiring sophisticated sequencing and interpretative tools to identify pathogenic variants from raw genetic data. While numerous tools exist for variant interpretation, their limited scope and disagreements on pathogenicity highlight the challenges in achieving reliable and useful genetic diagnoses. This is particularly evident in the domain of heritable hearing loss, where the AudioGene [13] tool represents a significant advancement in phenotype-to-genotype prediction through audiometric data.

1.2. AudioGene: A Milestone in Hearing Loss Diagnosis

First introduced in 2008, AudioGene was a pioneering step towards utilizing machine learning for the genetic diagnosis of heritable hearing loss [13]. By ranking common ADNSHL genetic loci based on audiometric data, AudioGene offered a new direction for sequencing-based

mutation screening. Subsequent versions, including AudioGenev4 and AudioGenev9, have built on this foundation, employing multiple-instance and ensemble approaches to refine genotype predictions [1-3]. Each version has sought to address inherent challenges such as class imbalance, sparse data, and the masking effects of age-related hearing loss, illustrating the ongoing evolution towards more accurate and personalized diagnostic tools.

1.3. Problem Statement and Research Objectives

Despite the advancements made by AudioGene over the past 15 years, significant challenges remain in accurately predicting a genetic diagnosis of heritable hearing loss. The new hybrid approach described in this thesis combines the latest machine learning models and custom visualization interfaces that allow a clinician or clinical researcher to incorporate much more context and add meaning to the model predictions. As discussed previously [1], the accuracy of AudioGene greatly reduces the utility of the predictions as it is known that there is as much as a 30-40% chance of incorrect predictions. However, with this new hybrid approach, a visualization "dashboard" will allow the user to see the "why" behind the diagnosis predictions, allowing them a glimpse into the "AI black box" that produces AudioGene predictions.

This new dashboard (made publicly available on the AudioGene website - audiogene.eng.uiowa.edu) will contribute to improved accuracy of diagnoses from two perspectives. The first is the *visualization of the input patient data* – the audiogram/age- against all other labeled audiograms. This is done through the deployment of five different types of visualizations:

1. Audio-Profile: A two-d plot showcasing the average hearing loss (in dB) over ten frequencies (124 Hz, 250 Hz, 500 Hz, 1000 Hz, 2000 Hz, 3000 Hz, 4000 Hz, 6000 Hz, 8000 Hz) for each age group in each gene, allowing for a comparison line of the patients hearing loss over time.

2. Audio Profile Surface: A Three-Dimensional surface plot that shows the dB loss per frequency over age. It showcases each gene's progression of hearing loss [4].
3. Spatial Analysis and Clustering: This is a combination of a bar chart with 23 clusters, showing the clustering of each gene in each cluster, and a three-D plot, displaying the audiograms in 3-D space. It also displays the patient's audiogram compared to the other audiograms for each cluster.
4. Ethnicity Pie Chart: Displays the ethnicity distribution of each of the 23 genes.
5. Count Bar Chart: Displays the count of each gene in the training data, showcasing the class imbalance.
6. Age Distribution Scatter Plot: Show the training data distribution by age for each of the 23 genes.

These visualizations aim to confirm or highlight the clinician's possible concerns about the prediction(s). For example, suppose the model predicts *COCH* and the appearance of the audio-profile, and the spatial analysis and clustering indicate to the user that the patient audiograms show a similar pattern to that of other *COCH* genes. In that case, this provides added confidence in and understanding of the hypothesized genetic diagnosis.

The second perspective of this dashboard approach involves the customization of AudioGenev9, allowing the clinician to incorporate many other factors not provided in the standard dataset used to train the model and subsequently make machine learning predictions. Each of these visualizations is intended to allow the user to gain further insights into each gene and how it appears in the model, thus allowing them to form a hypothesis regarding which of the genes may or may not be a causal factor in the phenotype (disease) observed in the patient. However, using this approach will likely induce other errors. It is certainly conceivable that a user may incorrectly hypothesize in a way that would remove a particular gene that may play a causal role in the observed phenotype. Nonetheless, this customizability may prove quite powerful, as the model can filter genes out, causing the reduced model to have fewer (incorrect) genes from which to choose, thus inducing less noise in its training data. The net effect of this

“pruning” is to allow the prediction of a gene more confidently due to the lower number of classes in the model.

With the “right amount” of guidance and customization, the model may prove more valuable to experts, guiding them to more plausible (and usable) diagnoses. This thesis focuses on developing and describing use cases of this hybrid approach to illustrate the tools' best-case usefulness.

First, we will present some critical background of machine learning and clinically observed hearing loss. Then, we will examine the details behind each visualization and the customization of AudioGene V9.1. Next, we will discuss specific case studies illustrating AGTD’s effectiveness for various genes. Lastly, we will draw some conclusions and outline recommendations for future improvements to this tool and its methodology to further improve the accuracy of genetic diagnosis for patients experiencing hearing loss.

CHAPTER 2: BACKGROUND

This chapter explores the existing body of knowledge surrounding ADNSHL, machine learning applications in genetic prediction, and the utility of visualization tools in diagnostics, identifying gaps that this research aims to fill.

2.1. Genetics of ADNSHL

Auditory perception (hearing) is the ability to perceive sounds through an organ, such as an ear. It involves detecting vibrations as periodic changes in the pressure of a surrounding medium. Using an audiometric test, decibel hearing levels (dBLoss values) are measured using different volume levels and tone frequencies. Measurements are recorded using an audiometer, a device used to create a reliable and objective measure of hearing loss severity [24-26]. When a person's hearing diminishes by at least 20 dB, this is classified as hearing loss [27]. The severity of the loss is measured in decibels (dB) of hearing loss relative to a "normal hearing" young adult, using a benchmark of 0 dB of hearing reduction. The cause of hearing loss could be from various factors, such as environmental, age-related, or heredity (gene-related). The factor we focus on in this thesis is genetic hearing loss, where an individual's hearing loss (deafness phenotype) is inherited from either one or both of their patients. The genetic factor that is said to be causative is known as the genotype. The phenotype is the manifestation of the genotype as measured clinically.

A further distinction is made between four modes of genetic inheritance:

i) Autosomal Recessive Non-Syndromic Hearing Loss (ARNSHL)

ii) Autosomal Dominant Non-Syndromic Hearing Loss (ADNSHL)

iii) X-linked

iv) mitochondrial inheritance.

There is often sufficient evidence to determine the inheritance pattern and syndromic status through clinical evaluation and family history. Yet, distinguishing among the 63 known causative genes within these distinct subcases from phenotypic evaluation alone remains a formidable challenge. Of these 63 genes, which have varying classifications ranging from limited to definitive, 35 have undergone review by ClinGen. Within this reviewed subset, 23 genes have been identified as being responsible for 75% of all cases in individuals of European descent. It was this particular set of 23 genes, which had specific genes associated with the loci as of 2021, that had adequate numbers of audiograms to enable the machine learning model classification and visualization techniques explored in this thesis.

Of particular interest in this thesis is the application of the AGTD using both AudioGene V9.1 and AudioGene V4, as well as advanced visualization tools, to help clinicians make better diagnoses of ADSNHL. ADSNHL is typically a bilateral, symmetrical, and progressive disorder that accounts for approximately 20% of all inherited hearing loss cases. Due to this, the natural progression of ADNSHL can be masked (by varying degrees) by a form of hearing loss that eventually afflicts all people – presbycusis [1]. Presbycusis, or age-related hearing loss, is a progressive, natural loss of hearing acuity, typically worse at higher frequencies. This can make identifying ADNSHL loss, typically more prevalent at low and middle frequencies, more complex.

2.2. Machine Learning

Machine learning (ML) is a branch of artificial intelligence (AI) that uses computational algorithms to find patterns in datasets, enabling the prediction of diverse outputs tailored to specific applications. The integration of ML into various new domains has surged, propelled by

advancements in computational capabilities and the exponential growth of data (collected incidentally or overtly) since the inception of the World Wide Web in the early 1990s. In this thesis, we discuss two different methods: supervised learning and semi-supervised learning. In supervised learning, models are trained with datasets where each example is paired with the correct label. Semi-supervised learning, particularly with SVMs, operates differently. It uses a small subset of labeled data to set initial boundaries and then refines its understanding with a more significant portion of unlabeled data. This approach allows SVMs to discern and apply the underlying patterns in the data to make predictions, providing a more robust and generalized performance on new, unseen data. The semi-supervised method leverages the strengths of labeled and unlabeled data, while the supervised method uses only labeled data, showcasing the adaptability of machine learning in both methods.

2.3. Ensemble and Semi-Supervised SVM Models

AudioGene's models, AudioGene V9.1 and AudioGene V4, are built from two frameworks: an ensemble classifier and a semi-supervised SVM.

2.3.1. Ensemble Learning in AudioGene V9.1

AudioGene V9.1 represents a novel implementation of machine learning through its selective intra-ensemble data partitioning framework [1-2]. This approach classifies hearing loss genotypes from audiometric phenotypes by selectively partitioning training data based on quantitative meta-features: the volume of gene-specific data, patient age at audiogram measurement, and audiogram shape. The partitioning strategy divides audiograms into categories based on data volume (small, medium, large), patient age (above or below 20 years), and audiogram shape (down-sloping, up-sloping, cookie-bite), with further subdivision based on age groups.

The ensemble architecture of AudioGene V9.1 is comprised of multiple sub-models: 3 K-Nearest Neighbor, 6 Adaptive Boosting, and 2 Random Forest models, with a Logistic Regression model serving as the combining module [1]. This ensemble leverages selective intra-ensemble data partitioning to address the diverse challenges in ADNSHL audiogram classification. Randomized resampling techniques are employed to balance the representation of genes within each size category before training, ensuring that the ensemble models operate on a normalized dataset.

2.3.2. Semi-Supervised SVM in AudioGene V4

In contrast to the ensemble approach of AudioGene V9.1, AudioGene V4 utilizes a semi-supervised SVM model designed for prioritizing genetic loci in patients based on audiogram analysis [3]. The preprocessing of audiograms includes combining bilateral measurements, polynomial fitting, and linear interpolation to handle missing threshold values. The model employs a multi-instance framework, grouping multiple audiograms into 'bags' representing each patient, capturing the age progression of the hearing loss. The semi-supervised SVM model used in AudioGene V4 is designed for multi-instance datasets, ranking loci based on the probability outputs of a modified SVM with a linear kernel. This SVM implementation handles multiple classes through a one-versus-one strategy and uses pairwise coupling for probability generation [3, 40].

AudioGene V9.1 and AudioGene V4 demonstrate the power of advanced machine learning techniques in genetic diagnosis for hearing loss. The selective partitioning and hybrid use of ensemble and semi-supervised learning models provide a refined method for interpreting the complex relationship between genotypes and audiometric phenotypes, advancing genetic testing capabilities for ADNSHL.

2.4. The Role of Visualization in Diagnostics

Visualizations are crucial in diagnostics, offering a means to interpret complex genetic information that can lead to actionable medical insights. They can transform high-dimensional data into visual formats that increase comprehension, foster better medical insights, and help facilitate the diagnostic process for patients.

2.4.1. Interpretation of Complex Genetic Data

The ability to visualize genetic variations and the way in which their phenotype impact is presented to a clinician. By mapping genetic data onto visual plots, practitioners can more easily find patterns that correlate with specific patient health outcomes. Previously, complex datasets required extensive computational analysis to understand. Still, they could not be represented in a manner that highlights key information at a glance, expediting diagnosis and enabling a personalized treatment plan [33].

2.4.2. Review of Existing Approaches

The development of advanced visualization tools has paralleled advances in genomics. Applications such as Circos [34] have proven valuable to users because of their capabilities in visualizing genomics data through its circular layout. This has allowed a comprehensive portrayal of genomes, revealing structural variations and otherwise unseen data patterns. Interactive tools such as Integrative Genomics Viewer (IGV) [38] allow clinicians to analyze genomic datasets in real time, exploring the implications of specific gene expressions and mutations for patients.

2.4.3. Enhancing Clinician Confidence

Visualization not only helps clinicians interpret data but significantly contributes to clinician confidence. The clarity and understanding clinicians gain by using these visual tools ensure that diagnostic decisions are made with a solid foundation of machine learning and visual evidence. These tools are designed to represent data in a way that humans can understand, thus reinforcing the clinician's intuition and expertise with observed visual evidence, helping them to achieve more productive outcomes [33].

2.5. Summary and Research Gap

This section narrows the existing gap within the body of literature surrounding ADNSHL, machine learning, and the utility of visualization tools in genetic diagnostics. It highlights the achievements made so far and looks at the research gap this thesis hopes to address.

2.5.1. Summary of Literature Review

The body of literature presented in this chapter confirms the vital role of machine learning in the genetic prediction of ADNSHL, using tools like the AudioGene website and previously developed ML models. The literature reviewed suggests that while significant strides have been made, the field still struggles with integrating machine learning predictions in clinical practice, especially when employing visualization [36].

2.5.2. Identification of Research Gap

The literature indicates that there is a pressing need for visualization tools that not only help present data but also enable interaction with machine learning models. There is a scarcity of tools that seamlessly combine predictive modeling with user-focused visualizations, providing

clinicians with a comprehensive view for better diagnostic decisions. These tools would not only present the data to the users but also allow for the inclusion of expert knowledge in the interpretation process, enhancing the reliability of the diagnostic decisions [33].

2.5.3. Conclusion

The gap in the field of genetic diagnosis outlined in this chapter forms the cornerstone of this thesis. It motivated the development of a hybrid approach combining cutting-edge machine learning models with an advanced visualization dashboard. This approach aims to enhance the diagnostic process from reliance on ambiguous machine learning models to an interactive dashboard, thus permitting better diagnostics, reducing ambiguity, and increasing understanding [37].

CHAPTER 3: METHODOLOGY AND DESIGN

This chapter explores the way in which the AGTD was designed and implemented. This includes the details of the visual interfaces, the ability to customize AudioGene V9.1, software frameworks, and the use of the Docker [28] containerization platform for deployment.

3.1. Design and Implementation of AGTD's Visualization Interfaces

In the course of implementing the AGTD "dashboard," many decisions were made regarding the number and nature of visualizations that would be included and the characteristics that would be displayed in each. The following six visualizations were implemented for the dashboard:

1) Audio-Profile View
2) Audio Profile Surface View
3) Spatial Analysis and Clustering View
4) Ethnicity Pie Chart View
5) Instance Count Bar Chart View
6) Age Distribution Scatter Plot View

The Plotly library [12] was selected to build each plot graphically in the construction of these visualization interfaces. In addition to being well-suited from a technical perspective, it is also free and open-source. During the original implementation of AGTD, the visualizations were based on our training set of 3,218 audiograms, composed of 1298 patinets, of which 40% were multi-instance and 60% were single instance. The data was preprocessed on the Flask server described in section 3.3.6.

3.1.1. Visualization Preprocessing

As with user input data, dbLoss values are usually missing at several of an audiogram's frequencies in our training data. Two methods were used to fill in the missing values to address this problem: interpolation and extrapolation.

3.1.1.1. Interpolation

When a patient's audiogram has gaps, i.e., missing dbLoss values, between the first and last non-missing values, the tool uses the Pandas interpolate function with linear interpolation to estimate inner missing values [11]. This method computes the inner gaps by considering the slope between adjacent known points. The interpolation formula applied by Pandas can be represented as:

$$y = y_1 + \frac{(y_2 - y_1)}{(x_2 - x_1)} \cdot (x - x_1)$$

Where y is the interpolated value at (known) position x, and y_1 and y_2 are the known values before and after the missing point at positions x_1 and x_2 respectively. The method ensures that only the missing values within the dataset range are filled in [11].

3.1.1.2. Extrapolation

For cases where data is missing at the audiogram's beginning and/or end, we use extrapolation, where the tool manually calculates missing values at the dataset's beginning and/or end. The extrapolation formula for estimating a missing value before the first known point, using the slope from the first two known points, is as follows:

$$y_0 = y_1 - m \cdot (x_1 - x_0)$$

where m is the slope calculated as $m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$, y_1 is the first known value, y_2 is the second known value, x_1 is the position of the first known value, x_2 is the position of the second known value, and

x_0 is the missing (known) value. The extrapolation process is mirrored for estimating values at the end of the dataset by applying the slope in the forward (axis-increasing) direction, where y_2 is the missing datapoint at the end.

$$y_2 = y_1 + m \cdot (x_2 - x_1)$$

3.1.2. Audio-Profile View

The Audio-Profile is a two-dimensional plot that displays decibels of hearing loss (dbLoss) at each of 10 standard audiogram frequencies from 125 Hz to 8000 Hz. It is tailored to depict the hearing loss profile for different age groups by years of age in two-decade intervals (0-19, 20-39, 40-59, 60+), enabling comparisons between a patient's audiogram and aggregated age group data. When a patient's audiogram contains missing values, the tool uses the same interpolation and extrapolation methods discussed previously to estimate the missing data points, ensuring a continuous profile is displayed. In this way, a user can be shown a complete graphical illustration of a patient's audio-profile for top predictions. For example, this tool may help the user identify a patient's audiogram shape as being one of the following general shapes: 'Up-sloping,' 'Down-sloping,' or 'Cookie-Bite,' as shown in Figure 1.

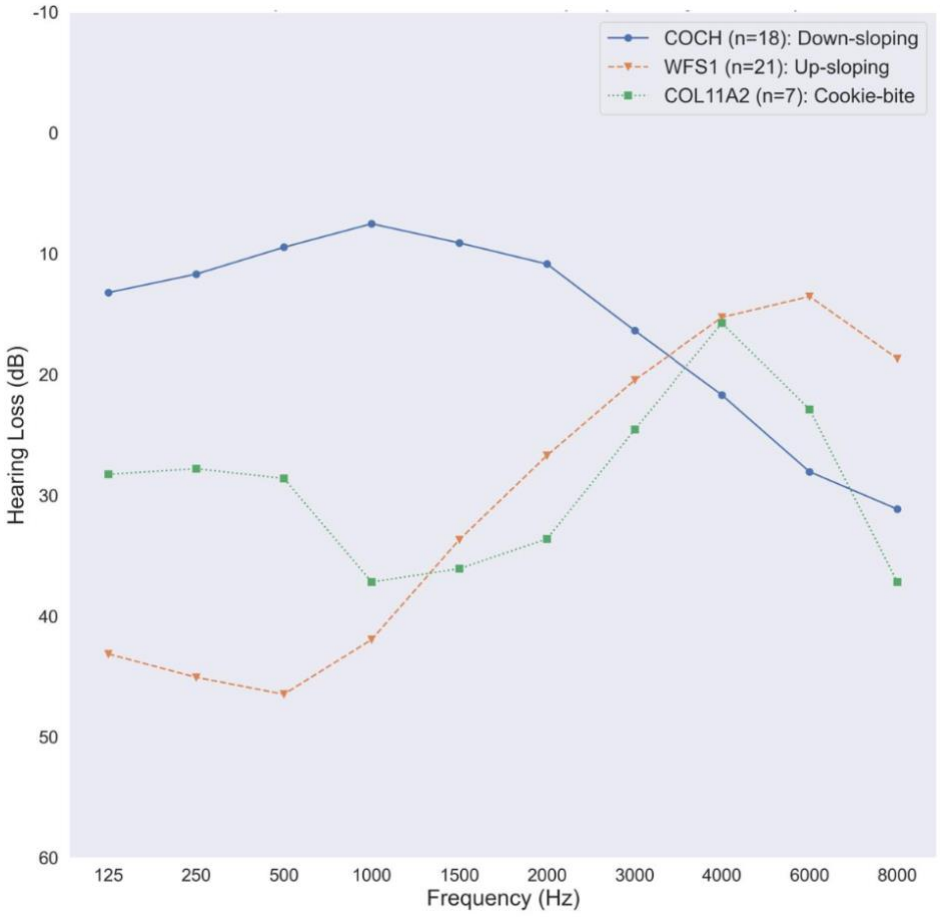

Figure 1. Audio Profiles were created from patients with a mutation in the *COCH*, *WFS1*, and *COL11A2* genes. The hearing test was performed when the subject was between the ages of 20 and 25. The audioprofiles represent the average hearing loss for those patients and are typical examples of the three primary shapes: down-sloping, up-sloping, and cookie-bite [1].

For the case in which a patient has multiple audiograms from which a prediction is being made (thus, AudioGene V4 is being used), the patient's average hearing loss at each of the ten frequencies is plotted. This tool is illustrated in Figure 2 for a patient (ID 8) with audiograms taken at six and seven years of age diagnosed with a *WFS1* genetic mutation.

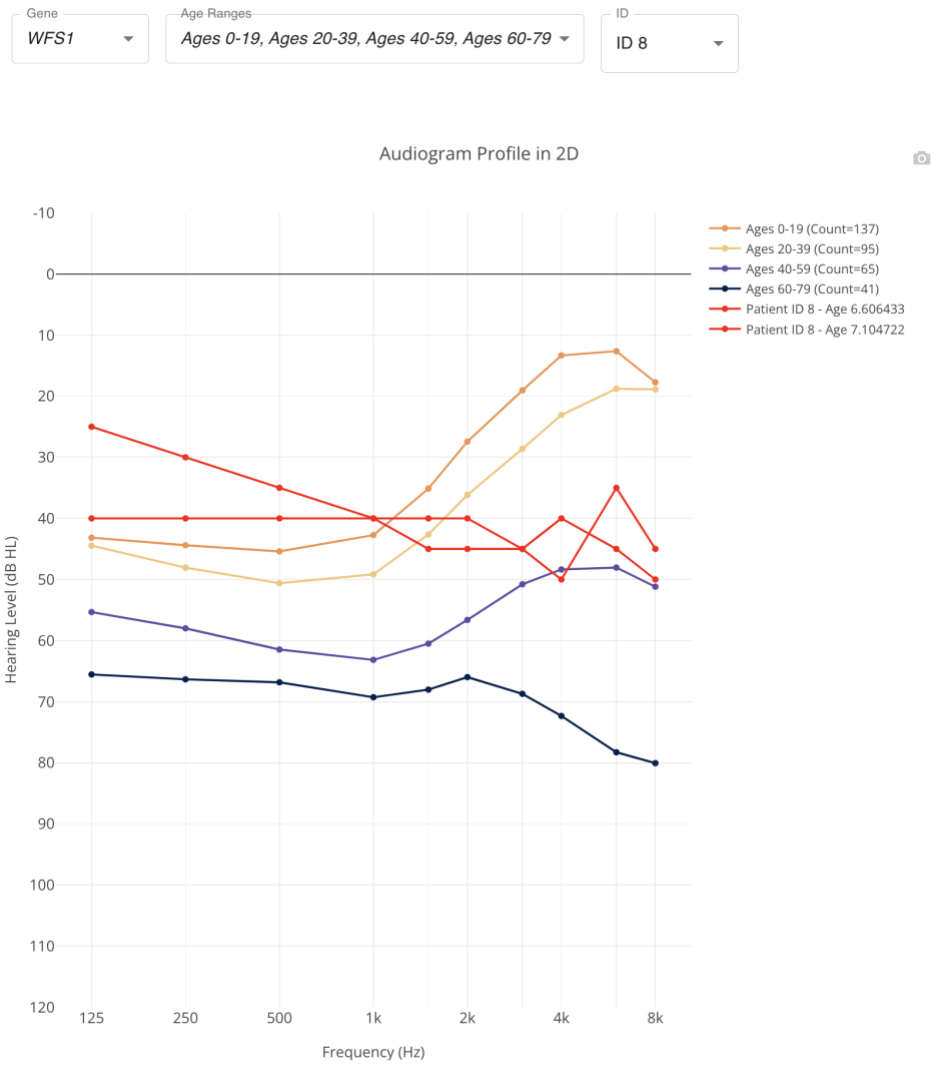

Figure 2. Two-D audio profile of *WFS1* with selections for the gene, age ranges, and patient to view. This profile displays the aggregate data of each age range, the patient's audiograms (in red), and the count of audiograms used to generate the line.

3.1.3. Audio Profile Surface View

The Surface Profile functionality within the AGTD "dashboard" leverages the concepts of an audio profile surface (APS) that can display how hearing loss changes over time for each gene. By projecting each audiogram measurement into points within a three-dimensional model, users can visualize the likely progression of hearing loss over frequencies and time – i.e., the patient at different ages. Utilizing the principles outlined by Kyle Taylor in his foundational

work on APSs, multiple surfaces are fitted using a least-squares regression with bi-squared robustness to create an idealized candidate APS [3]. The fitting process involves a rank-ordering system based on the root mean squared error (RMSE) obtained through k-fold cross-validation. This rigorous statistical approach ensures the surfaces accurately represent the expected patterns of hearing loss, such as 'cookie-bite,' 'down-sloping,' 'up-sloping,' and 'flat.' Specifically, the equations used for candidate surfaces are chosen to capture the various patterns of hearing loss encountered in clinical observations. For instance, a second-degree polynomial along the frequency axis and a third-degree polynomial along the age axis most often best represent a typical APS.

Utilizing Plotly also enabled real-time interaction with the APS. Users can select genes associated with hearing loss, and the tool dynamically generates the corresponding APS. The dashboard also provides an interface to allow users to plot an individual's audiogram overlaid onto this surface as a piece-wise linear curve, providing a visual contrast between the typical progression of hearing loss and the individual's data. There is also an ability to select (freeze) points and focus on a specific data point to help the user focus on which points in the APS they want to examine in detail. This functionality can be seen below in Figure 3, demonstrating the APS of *ACTG* with an inputted audiogram (shown in Figure 4) mapped onto its surface as a red line.

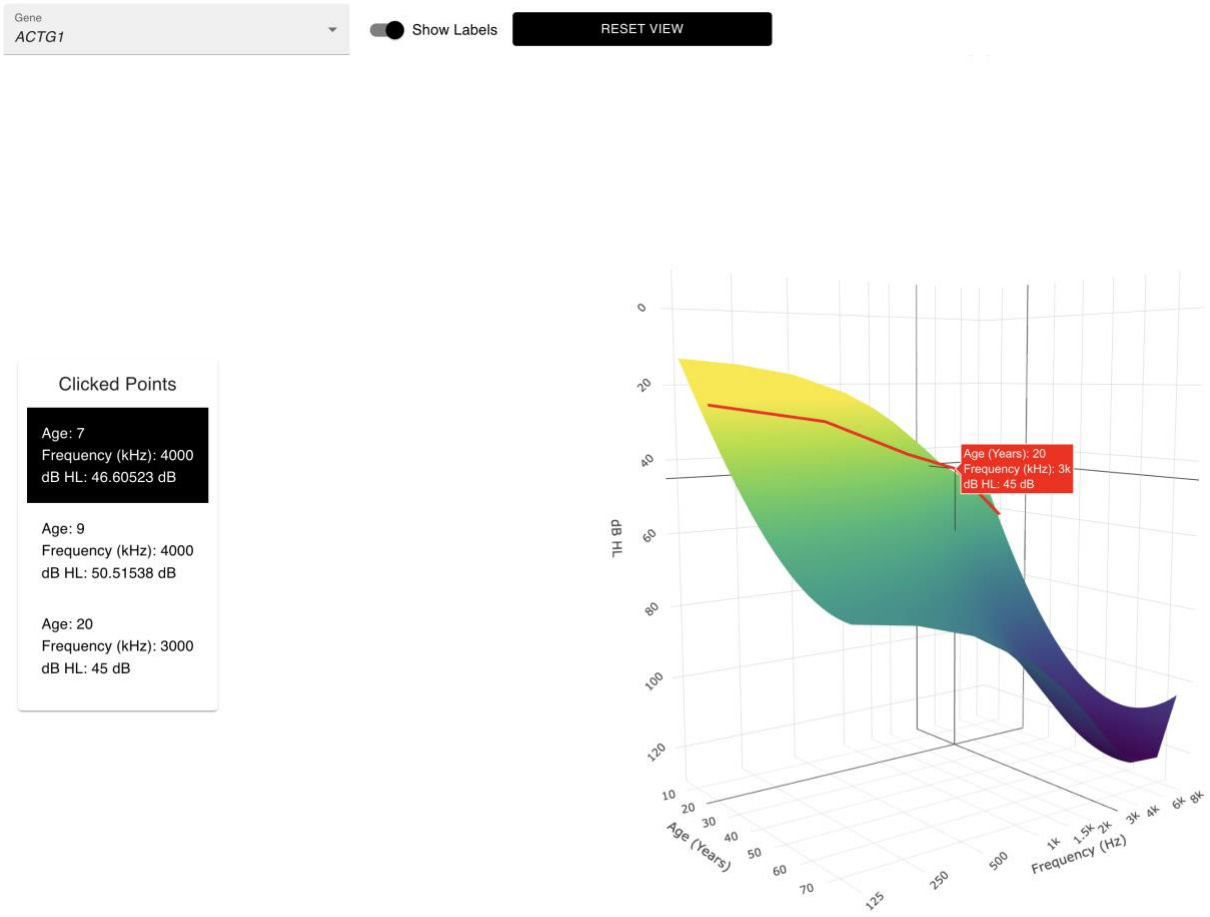

Figure 3. The audio profile Surface View within the 3-D plot and the ability to select genes show labels when hovering over the graph and resetting the view.

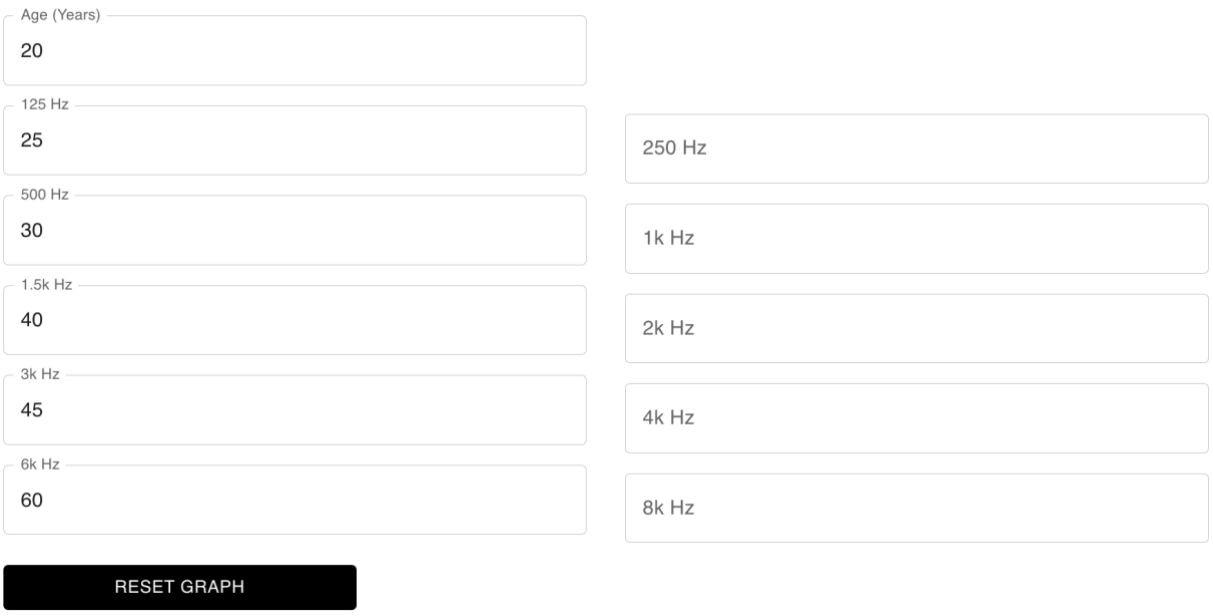

Figure 4. The audiogram patient at 20 years of age is mapped onto the APS in Figure 5.

3.1.4. Spatial Analysis and Clustering Views

Spatial analysis and clustering within the AGTD "dashboard" are essential for helping clinicians uncover patterns and relationships within the audiometric data that are not otherwise clear based on the predictions provided by the models alone [17]. By segmenting the data in clusters, users can identify groups of audiograms that share similar features, aiding the clinician in recognizing why a prediction was made and any other characteristics that may help give confidence in a set of predictions. The dashboard presents this tool in the first visualization seen, as shown in Figure 5, showcasing the two linked components, the Bar Chart (Figure 8) and the Three-Dimensional Plot (Figures 11-14).

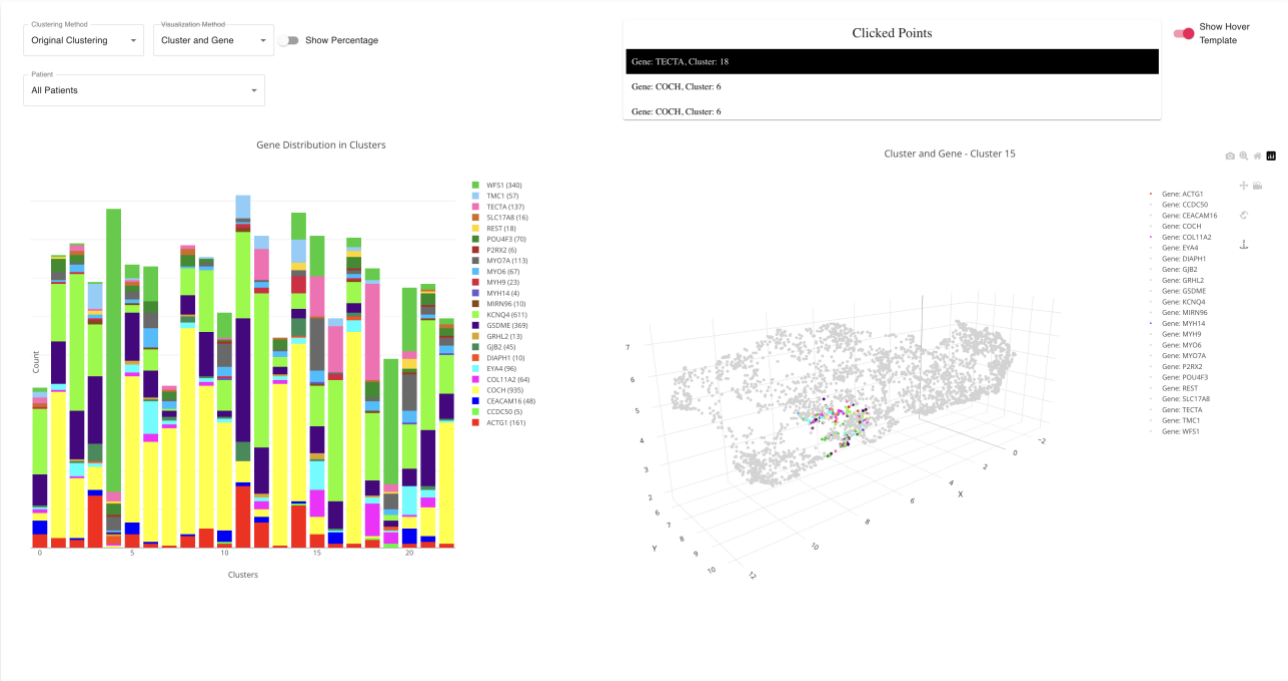

Figure 5. The bar chart on the left displays 23 unique clusters and their gene segments, with height based on the count of the gene in the cluster—three-dimensional clustering on the right with Cluster 15 selected from the Bar Chart.

3.1.4.1. KMeans Clustering Sub-View

The clustering mentioned above is performed using KMeans, a widely-used algorithm that partitions the dataset into K distinct, non-overlapping subsets or clusters [16]. The algorithm iteratively assigns each data point to the nearest cluster centroid (the mean of the points in the cluster). Then, it recalculates the centroids based on where the current cluster belongs [18].

In our approach, we cluster the genes into K=23 distinct clusters, correlating with the 23 genes associated with ADNSHL. Each cluster corresponds to audiograms with similar hearing loss patterns, which could indicate genetic similarities. By analyzing these clusters' distributions and characteristics, a user may be able to infer relationships between the characteristics of an individual's audiogram(s) and others based on the genotype.

3.1.4.2. UMAP Transformation

To convert the 11 dimensions (Age + 10 Frequencies) of our training data into a three-dimensional space (needed for visualization), we use the Uniform Manifold Approximation and Projection (UMAP) method [32]. UMAP is an algorithm that prioritizes the preservation of the global and local structure of the data when transitioning from a high-dimensional space into a lower one. It functions by finding consistent patterns in the data across various scales and then projecting the data into a lower dimensional space to attempt to retain the true relationships among the data [31-32].

UMAP is particularly useful in our case as audiograms present a non-linear relationship within our dataset, where patterns may be lost in more traditional methods like PCA [30]. In our exploration of methods available to use in our dimension reduction, we utilized several tests, including the Silhouette and K Nearest Neighbors (KNN) scoring metrics, which produced several compelling observations:

1. Higher Silhouette Score: UMAP has a silhouette score of 0.367, while PCA only achieved a score of 0.250. The silhouette score measures how similar an object is to its cluster compared to the other clusters, where a higher score indicates better-defined clusters. From these results, we conclude that UMAP's higher score suggests it has done a better job maintaining the dataset's structure and creating distinct, dense clusters than PCA.
2. Better KNN Classification Accuracy: To see which method retained more of the meaningful structure of the data required for accurate classification, we ran a KNN comparison using the reduced data from UMAP and PCA, respectively. Although the KNN classifier accuracy with the highest score was the original 11 features (56.6%), UMAP's reduced features produced a KNN accuracy of 53.3%, which is closer to the

original feature accuracy than the PCA case - 52.1%. This indicated that UMAP was able to retain more of the meaningful structure required for accurate classification than PCA.

3. Visual Clustering Interpretation: Although a more subjective measure, from Figures 6 and 8 it appears that UMAP creates greater separation of classes and more distinct clustering patterns. This was expected as UMAP attempts to preserve local and global structure in fewer dimensions, which often leads to more meaningful visualizations, where similar data points are grouped tighter and may be better-distinguished from other groups [31].

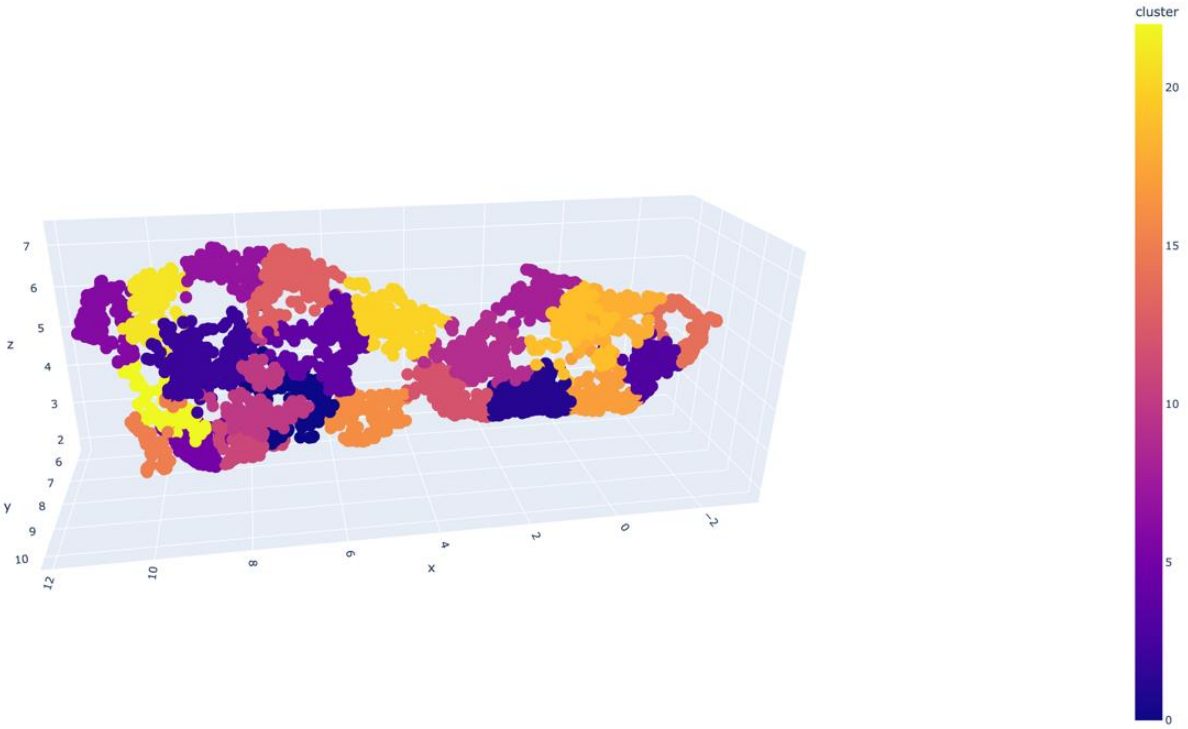

Figure 6. Clustering after reducing the original 11 features into 3 dimensions via UMAP.

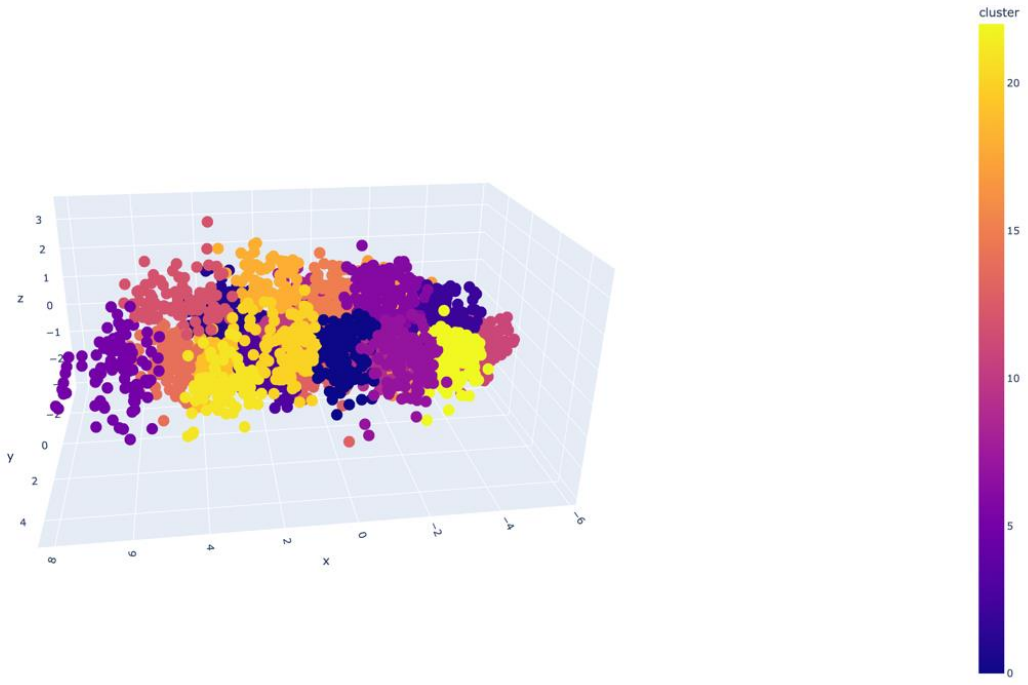

Figure 7. Clustering after reducing the original 11 features into 3 dimensions via PCA.

Using UMAP, we can then construct a useful 3D plot to display clusters of similar audiometric profiles, allowing users to identify and analyze patterns within the data more clearly.

This clustering is then visualized in two linked displays: the bar chart and the three-dimensional plot of the audiograms that were clustered.

3.1.4.3. Bar Chart Sub-View

The bar chart represents each of the 23 distinct clusters mentioned previously, with options for the user to select the Clustering Method, Visualization Method, and the patient they want to analyze, as seen in Figure 8. The bars are shown in two ways: by count, and by percentage. The count (Figure 8) displays the bars by the number of instances of each gene within each cluster, whereas the percentage (Figure 9) displays the total percentage of the gene within each cluster.

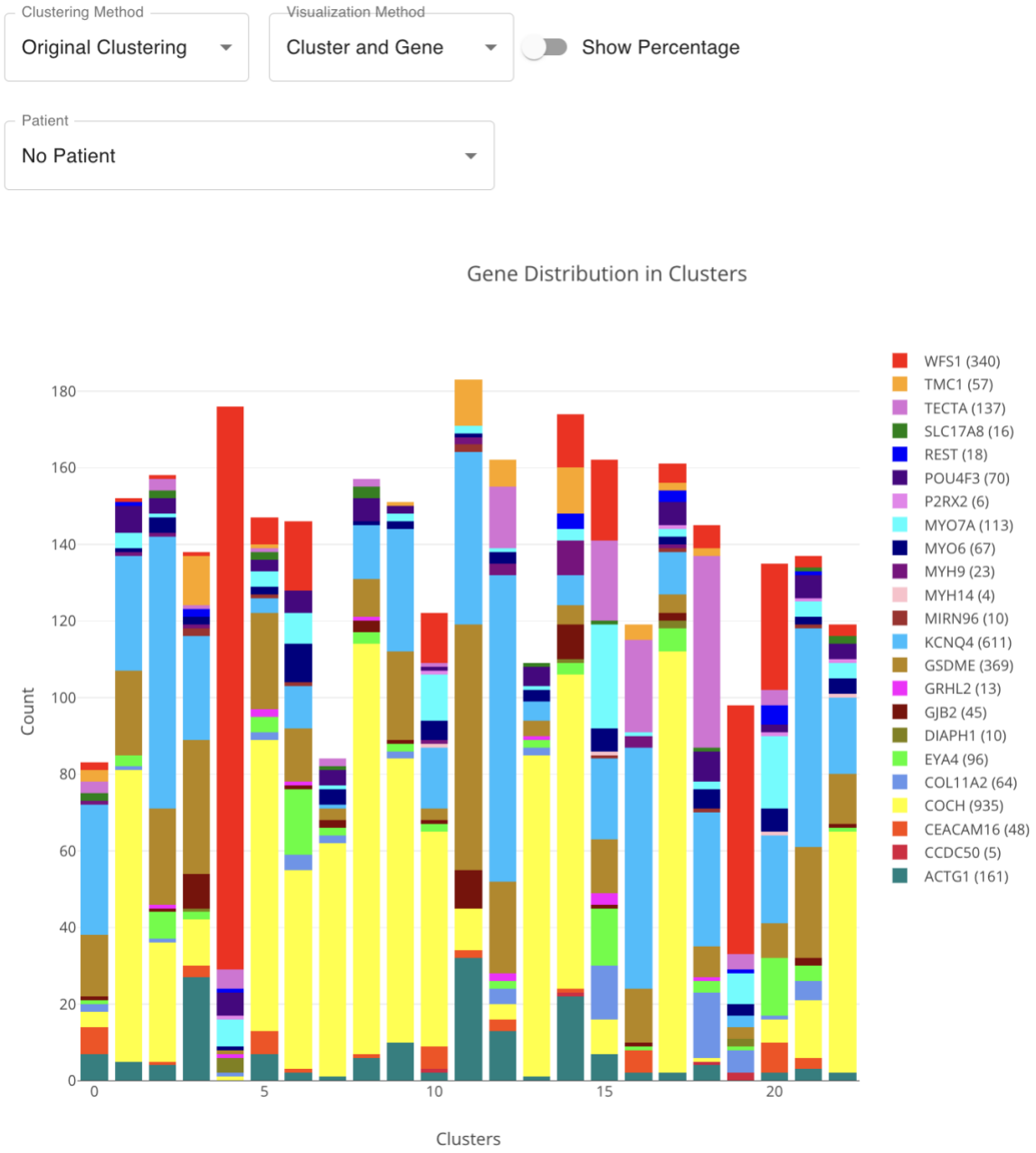

Figure 8. Bar Chart showing the 23 Clusters with the genes segmenting each bar by count of each gene within each cluster (bar).

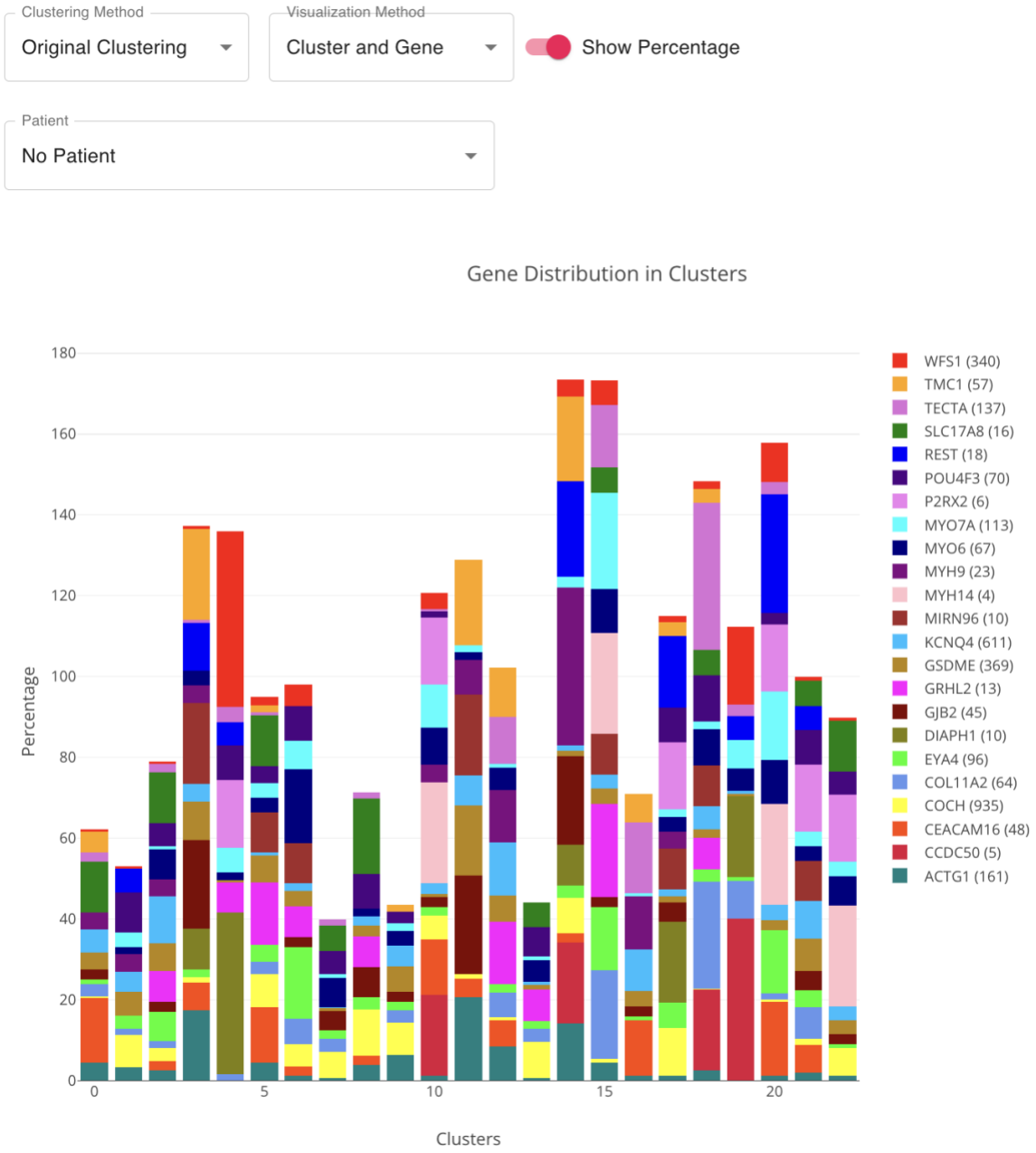

Figure 9. Bar Chart showing the 23 Clusters with the genes segmenting each bar by percentage of gene within each cluster (bar).

The Clustering Method Option

The Clustering method determines what is displayed in the graphical display shown in Figure 8; however, there are two options to choose from: “Original Clustering” and “Greedy Clustering.” The original clustering uses the previously mentioned technique of KMeans to show

each unique cluster and the genes represented within them. However, when the "Greedy Clustering" method is selected, a greedy method of clustering is employed (algorithm shown in Figure 10). This alternate method was developed to assist in finding the gene that best represents the cluster. The algorithm seeks to assign one of 23 genes to each cluster formed by a K-means algorithm based on the prevalence of the gene within each cluster. It does so by iteratively selecting the most prevalent gene for each cluster that has not already been assigned a gene, thereby maximizing the representation of different genes across clusters.

```
Output: assignedGenes, clusteringData, success
assignedGenes ← empty dictionary
geneSet ← empty set
geneOccurrences ← count of genes in clusterCounts
success ← false
allGenesAssigned ← false
for clusterIndex = 0 to numClusters − 1 do
    currentClusterCounts ← counts for current cluster
    sortedGenes ← genes sorted by frequency in descending order
    for gene in sortedGenes do
        if geneOccurrences[gene] > 1 then
            assignedGenes[clusterIndex] ← gene
            geneSet ← geneSet ∪ {gene}
            geneOccurrences[gene] ← geneOccurrences[gene] − 1
            break                          ▷ Assign gene and move to next cluster
        end if
    end for
end for
allGenesAssigned ← size of assignedGenes = numClusters
if allGenesAssigned then
    success ← true
    Filter clusteringData based on assignedGenes
end if
return assignedGenes, clusteringData, success
```

Figure 10. Greedy algorithm for selecting one gene to represent each of the 23 unique genes.

Using this approach, we can easily approximate the gene that would be most likely to be classified within that cluster. The greedy algorithm results are illustrated in Figure 11.

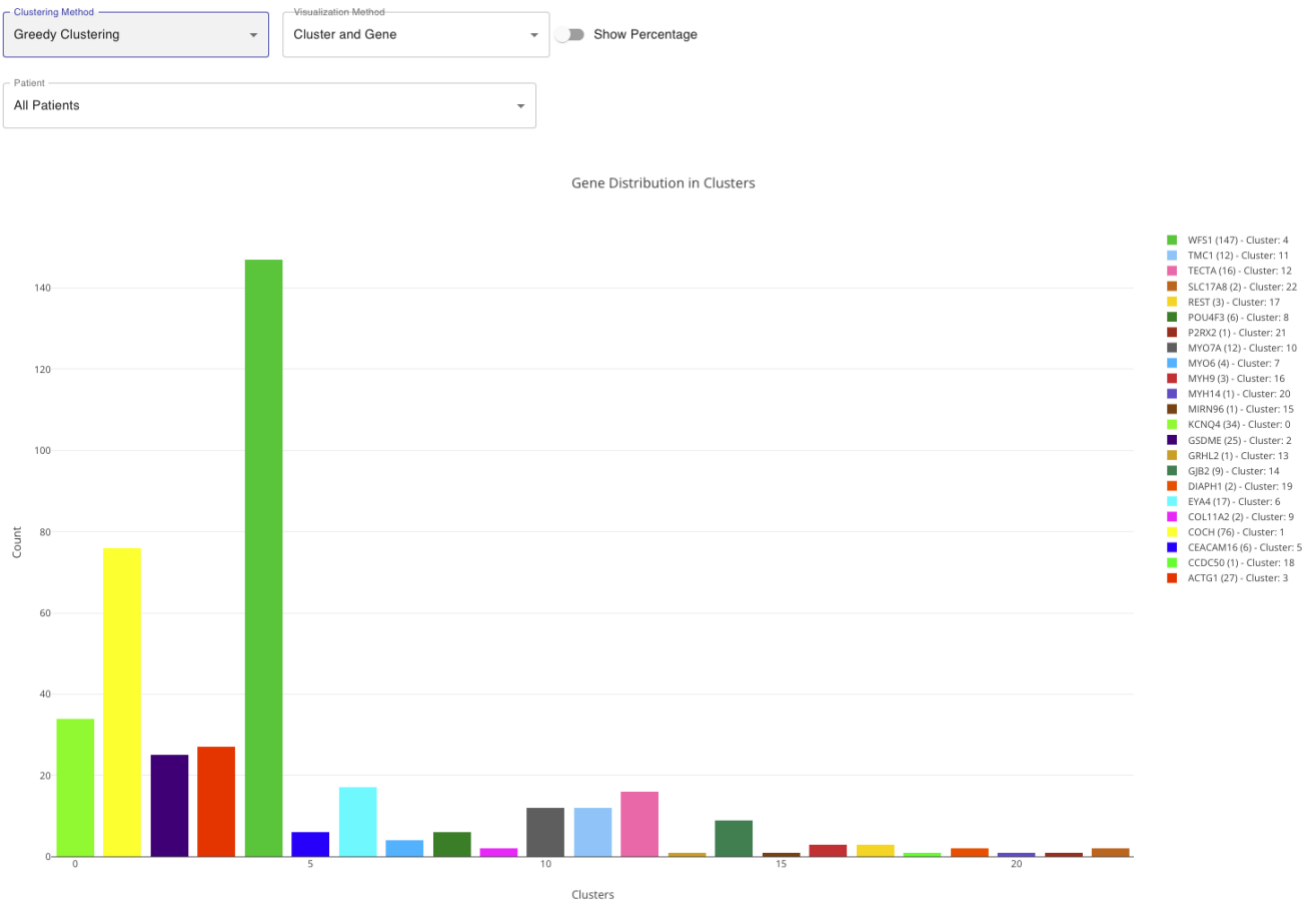

Figure 11. The bar chart with the Greedy Clustering Method chosen. The plot illustrates how each of the 23 genes are assigned to each cluster.

The visualization method allows users to change how they want to see the clusters displayed. Three options are provided to the user: “Cluster and Gene,” “By Cluster,” and “Clusters Visualization.” The visualizations are then seen in the Three-Dimensional Plot.

3.1.4.4. Three-Dimensional Plot Sub-View

Figire 11 shows the clusters and the audiograms in a three-dimensional space after being transformed from 11 dimensions down into 3 dimensions. The plot provides a clear picture of how each cluster is presented and the gene classifications of each audiogram within those clusters. The Plotly library allows the user to dynamically click and hold, then orbit around the

plot, showing the clusters from all perspectives and giving the user a complete view of the clusters. Besides the orbital features, a detailed label is displayed when a user hovers over an audiogram data point. The cluster classification of the nearest point and associated genes are displayed. However, for a larger red point, representing a currently selected patient based on user-inputted data, the patient's ID, Cluster assignment, top three gene predictions from the AudioGene model used (AudioGene V4 and AudioGene V9.1), and the nearest genes to the patient's data point are provided. The plot is shown based on the visualization method selected in the bar chart selection box.

Visualization Method

This option allows users to change how they want to see the clusters displayed in the three-dimensional plot. Three options are provided to the user: "Cluster and Gene," "By Cluster," and "Clusters Visualization":

- Cluster and Gene: This is used to highlight the cluster selected in the bar chart and show the individual audiograms (represented as points in that cluster) with a color unique to the gene as it was classified, as shown in Figure 12.

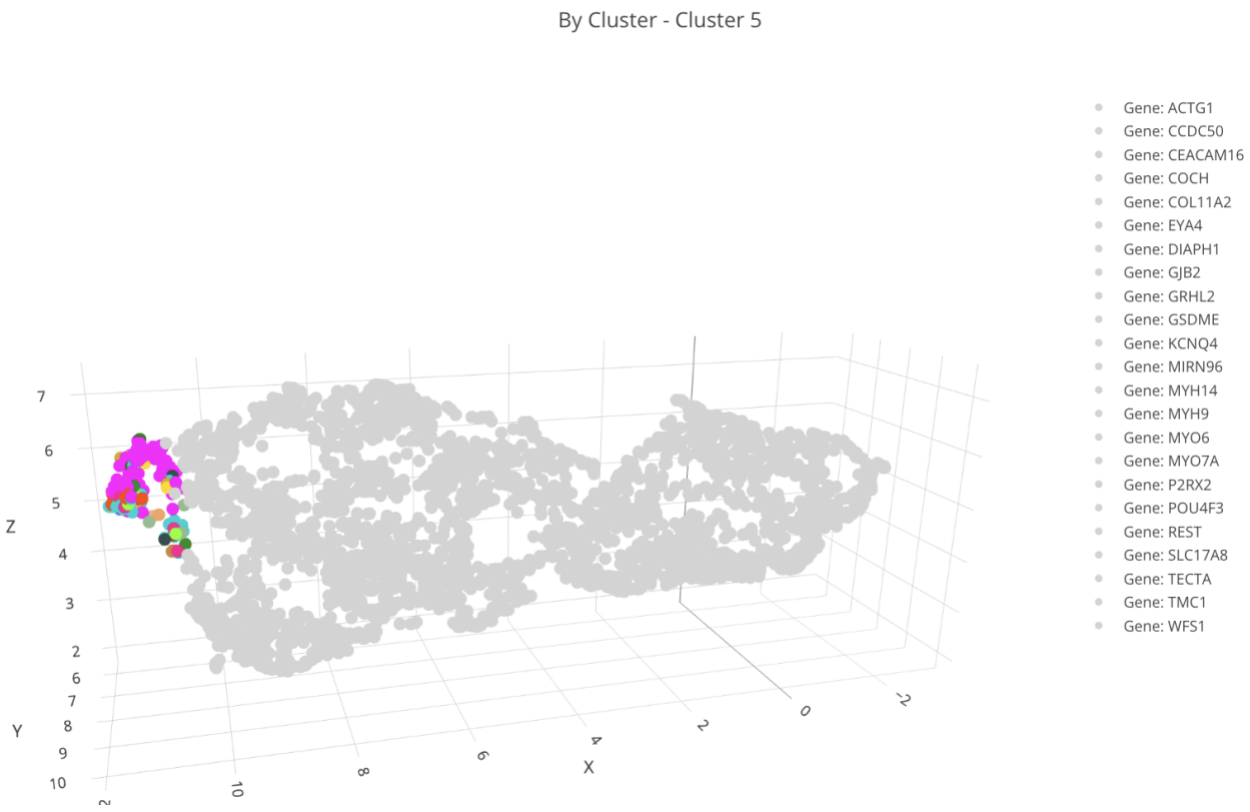

Figure 12. Cluster and Gene method, highlighting the cluster with distinct colors (by gene)

- By Cluster: As is shown in Figure 13, this display highlights the cluster itself, representing each audiogram associated by their genotype by a single color, whereas each audiogram is displayed with the gene-specific color in “Cluster and Gene.”

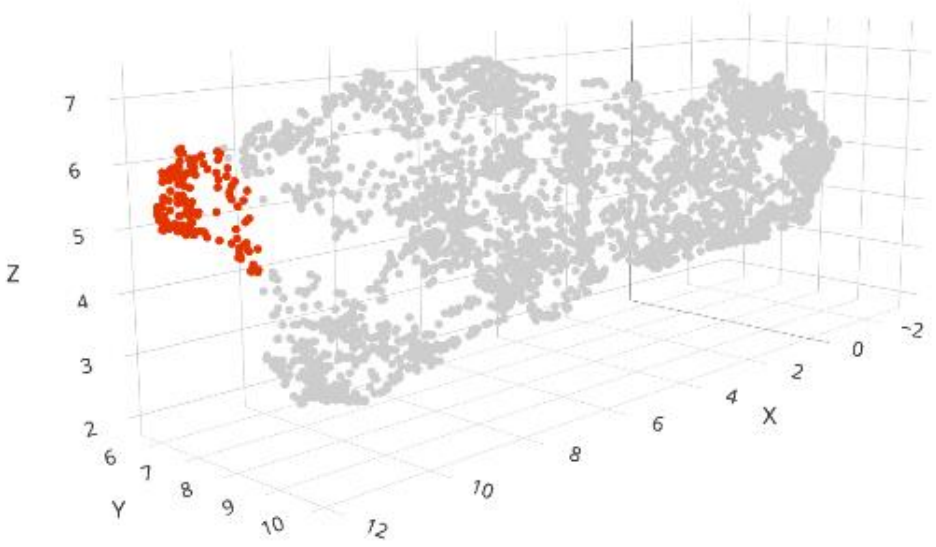

Figure 13. By Cluster Method, the entire selected cluster is highlighted in red.

- Clusters Visualization: This method displays each cluster by a unique color, regardless of which cluster is selected in the bar chart. The visualization can be seen in Figure 14.

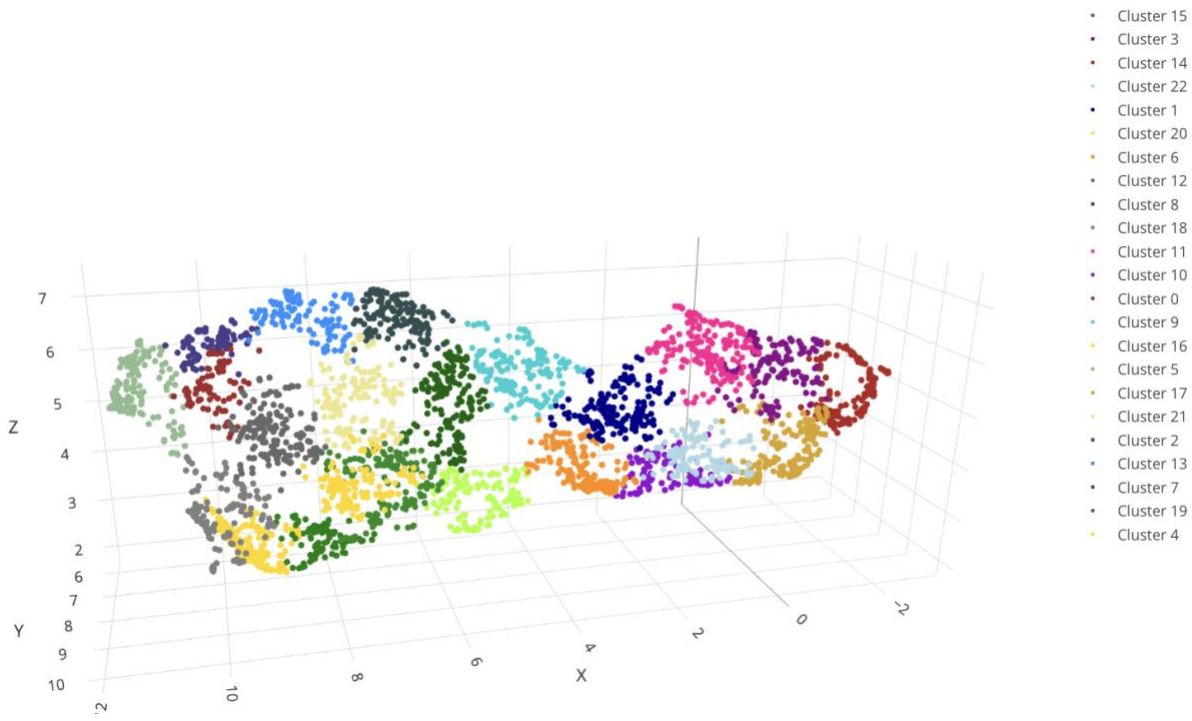

Figure 14. Cluster Visualization, showing all clusters in their own distinct colors.

The goal of the visualization methods provided above is to help the user find patterns and relationships that may be present within the cluster that may be more subtle in other display methods.

Patient

Audiograms being input by a user and representing a new undiagnosed patient can be displayed inside the clustering displays to demonstrate how the user's audiogram relates to other phenotypic representations of each gene, potentially clarifying the user's associated audiogram with a genotype given its surrounding neighbor's classifications. An example of this is shown in Figure 15, with a patient (ID 8) with three audiograms.

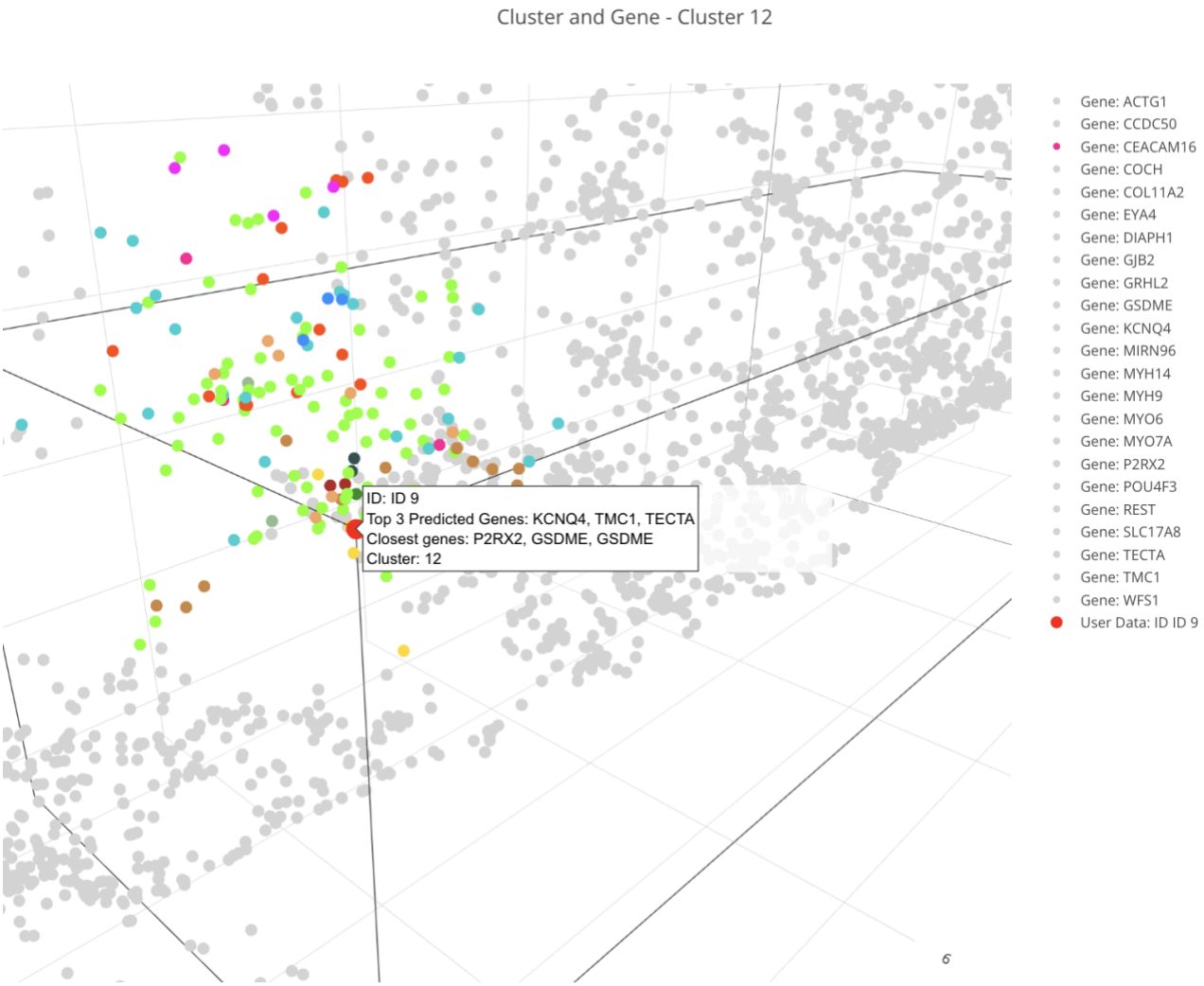

Figure 15. The three-Dimensional Plot with the highlighted cluster is shown in color, and the other points are light grey. It also shows a patient's ID being hovered over with the information display and the points the user clicked.

When a patient has more than one audiogram, their information is condensed so that the patient appears as one singular point instead of multiple points. This is intended to allow for easy visualization of the patient. To do this reduction, the following method is performed:

1. Median Age Calculation: the patient's median age across all audiograms, which serves as the reference point.

2. Selecting Close Audiograms: audiograms from within a two-year age range of the median age found in the previous step. This selection process aims to capture the general appearance of a patient's audiogram "near" that age.
    a. Choosing the Closest Audiogram: If one or more audiograms are within two years of the median age, we select the audiogram closest to the median. This audiogram is considered to be representative of all of the patient's audiograms for the three-dimensional plot (but not the prediction model input).
    b. Interpolating an Audiogram: We create a new, interpolated audiogram if there are no audiograms within two years of the median age. This process involves estimating the hearing loss at each of the ten frequencies for the median age by using the data from the patient's other audiograms.
        i. The interpolation considers the trend in hearing loss across different ages and frequencies, creating a synthetic audiogram that represents the patient's hearing loss as it has been measured at the median age.

These nearest genes are determined using the K Nearest Neighbors (KNN) algorithm, a method widely adopted in machine learning applications for its effectiveness in classification and regression tasks.

KNN algorithm

The KNN algorithm operates by finding the **k** closest data points in the feature space — in this case, the patient's 11 audiogram features — and making predictions based on the distance to those points [20-21]. The distance metric employed to gauge the 'nearness' of genes is the Euclidean distance, or the 'straight-line' distance between two points in multidimensional space [22]. This is a fundamental distance metric widely used in machine learning and statistical

analysis for its simplicity and efficacy in reflecting the actual geometric structure of the data. By fitting the KNN model to the audiometric features and querying it with the patient's data, we obtain the three closest genes that may help the user gain confidence in the prediction provided by AudioGene.

3.1.5. Ethnicity Pie Chart View

The AGTD “dashboard” also includes an ethnicity bar chart, as shown in Figure 16, for two reasons. First, it can inform the user of the population distribution in the training set, which may inform the user regarding the limitations resulting from a mismatch with the predicted patient. The second reason is to give the user an indication of which genes may be associated with specific ethnicities, potentially adding confidence in the model predictions produced by AudioGene. For example, suppose a given patient’s genetic prediction for their audiogram history was *TECTA*, and the user examines the ethnicity pie chart. In that case, they may find that most people with the *TECTA* mutation were Dutch; for example, if their patient were Dutch, this would provide added confidence in the prediction. However, if their patient did not have any of these associated ethnicities, then that could give them a reason to distrust that the model was making an accurate prediction, leading them to consider leaving this gene out of a custom model. The AudioGene V9.1 model customization tool can then be employed to build an appropriate custom model. Figure 16 shows a pie chart of the ethnicity distribution for *TECTA*.

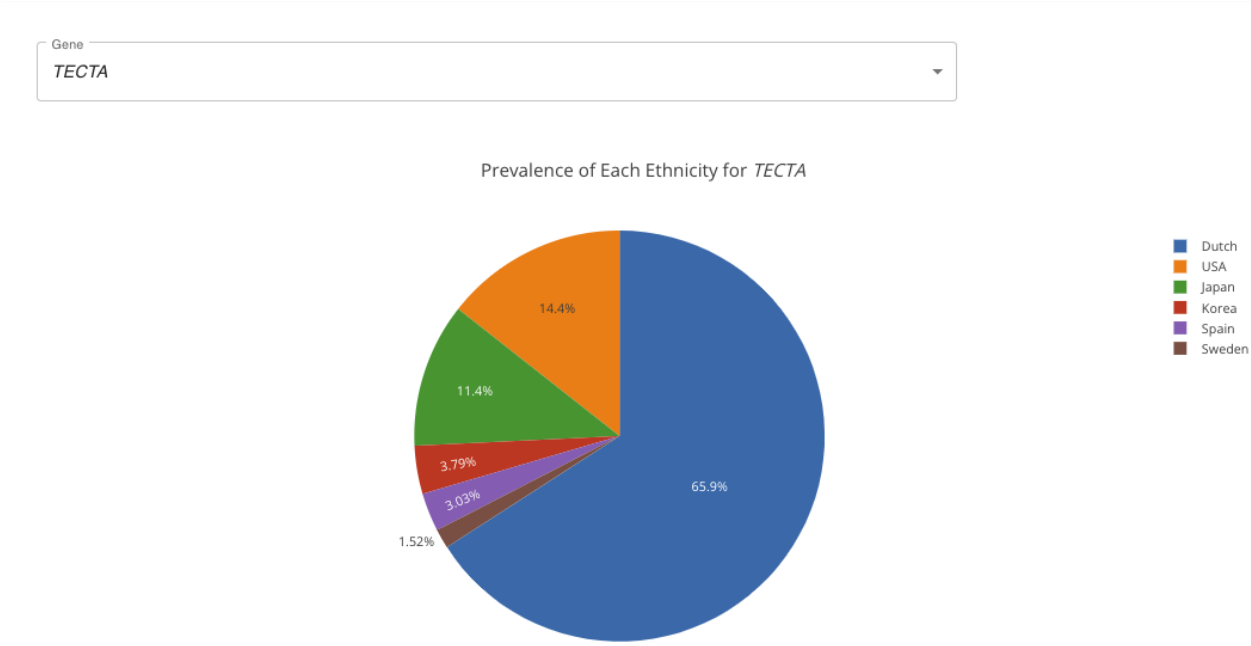

Figure 16. The Ethnicity Pie Chart displays the ethnicity distribution for the *TECTA*.

3.1.6. Count Bar Chart View

AGTD also displays to the user how the training data is spread out and how class imbalance may affect predictions provided by the model. While it is impossible to know the true distribution across a large (e.g., worldwide) population, it is highly likely that the training data available to us that the 2024 datasets may not fully represent the population due to the vast genetic heterogeneity of ADNSHL, as it complicates the process of accurately linking many cases to specific ADNSHL genes, affecting model accuracy for minority genes [39]. Through the use of this dashboard, the user may also be able to see how the AudioGene models could have a lower probability of predicting minority classes and a higher probability of predicting the majority classes (correctly or incorrectly). However, by being made aware of the issue of class balance as it relates to their particular case to be diagnosed, the user may be able to "address" this class balance and better understand how it affects the model and resulting predictions. This

method allows the user to see how the data is clustered when there is a more representative spread among the majority and minority classes, allowing them to find more subtle relationships and make more informed decisions. Figure 17 portrays the count chart of the audiograms in each of the 23 genes.

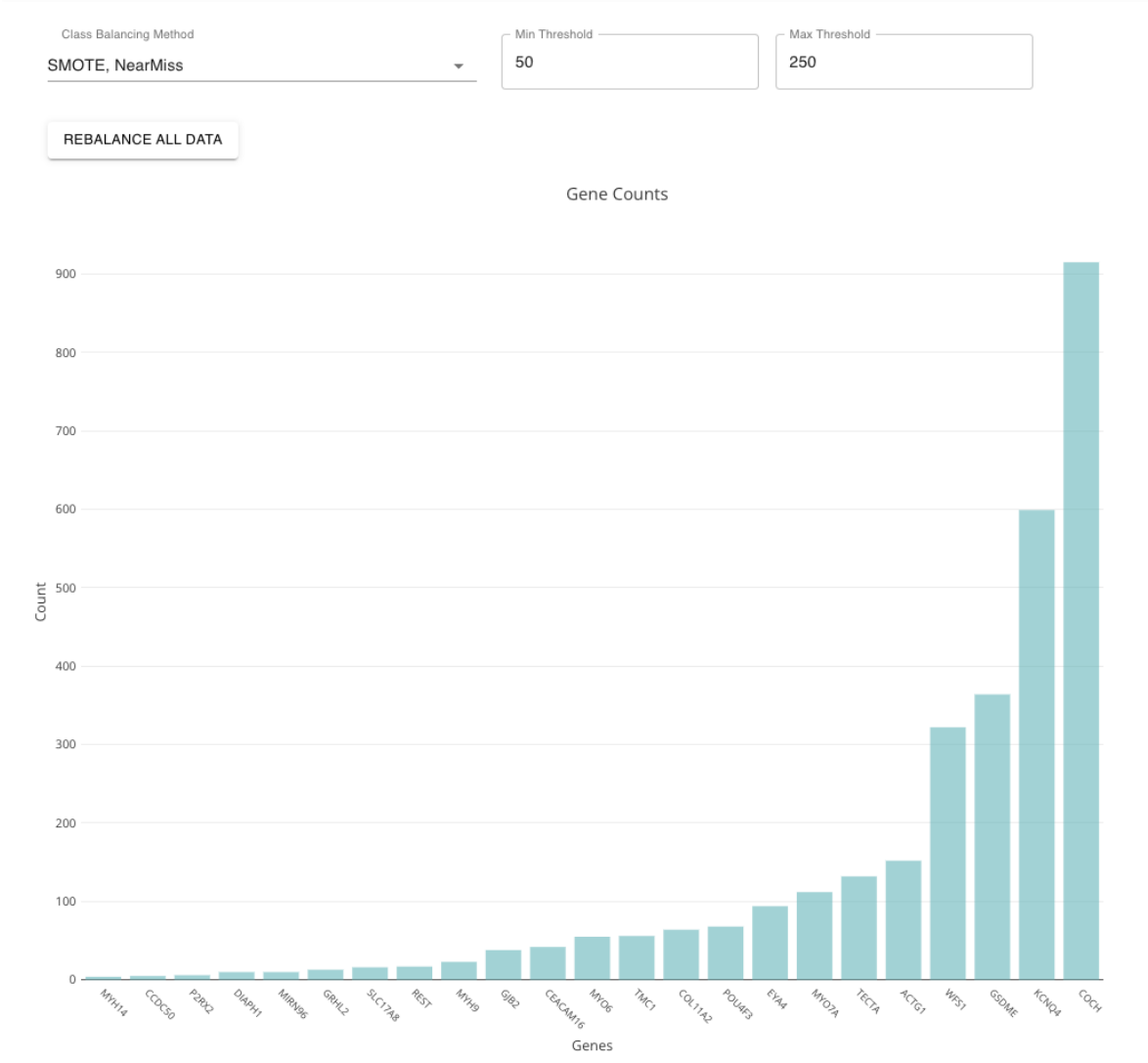

Figure 17. The plot displays the counts of each gene in the training data, a selection of different class rebalancing methods, and input fields to rebalance.

3.1.7. Age Distribution Scatter Plot View

The age distribution plot (Figure 18) shows how the training data is spread out over the age range of 0-100 years for each of the 23 unique genes. This visualization demonstrates the

model's potential weaknesses in predicting different age ranges for each patient and the usual age spread for each gene. For instance, gene CCDC50, for which no data points past age 60, could indicate two things. The first one is that if a patient has a genetic mutation in CCDC50, AudioGene will be unlikely to predict accurately for that age. Secondly, it could indicate that gene mutation never appears in patients of that age, potentially allowing clinicians to rule out this gene in the prediction. While this latter observation is highly unlikely to be known, nor even to be the case for any given gene or patient, the display in Figure 18 illustrates the distribution of audiograms for the ages within the gene, and it allows the user (patient or clinician) to use that information as they see fit.

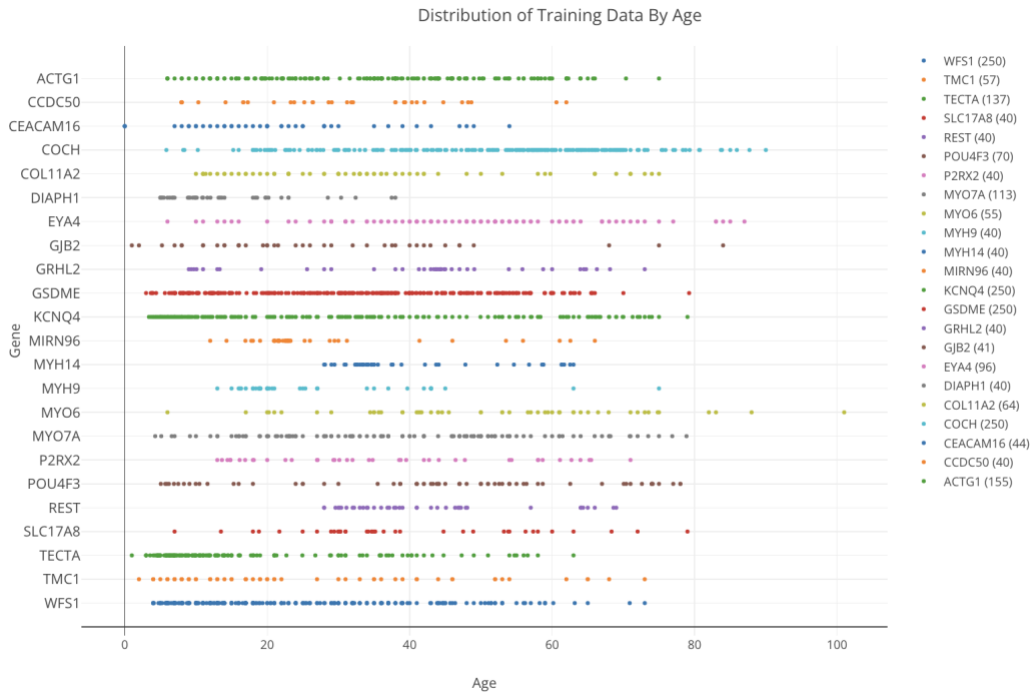

Figure 18. Distribution of the training data by age for each of the 23 genes.

3.2. Customization of AudioGene V9.1

The AGTD "dashboard" also allows users to customize AudioGene V9.1 to capture expert knowledge about their patient and the field to help guide the model to make more accurate predictions. The method incorporates the user's ability to filter the genes (by removing the data from selected genes) on which the model is trained. This customization tool will then rebuild the model with the new parameters (gene set) to update the model to the user's specifications. After the model is built, the dashboard will present the new prediction results using the custom AudioGene V9.1 model, assisting the user in making a diagnostic decision.

Due to technical limitations of the platform on which AudioGene V4 is built (Weka), it was not yet possible to support customizations in V4 at the time of writing this thesis. Therefore, in customizing the model, the user is limited to customizing V9 for single-instance patients.

3.2.1. Implementation of Customization

To implement the customization, we used the source code of AudioGene V9.1 and updated it based on the user specification. In training the model, the same interpolations and extrapolation methods were used as in the original training of the model [1]. For filtering the genes, we used Panda's library function `pandas.isin` to exclude any audiograms with a gene diagnosis not selected by the user [11].

3.3. Development of AGTD: AudioGene Translational Dashboard

The AudioGene Translational Dashboard is a web-based tool that allows clinical users and deafness researchers to explore the more mathematical and technical "backend" of AudioGene predictions through more intuitive and interactive graphical interfaces. AGTD also provides a "window" into the large dataset of audiograms used to train the AudioGene model and to help them gain insight into the decision-making processes of the AudioGene V4 and

AudioGene V9.1. AGTD was developed using the SERN stack (SQL [6], Express.js [8], React [15], and Node.js [10]), with Nginx [29], which was used for access control and security. All the data-intensive tasks of the tool, such as the model predictions and data analysis for visualizations, were developed in the widely used Python language and accessible through a Flask [14] server. All of the provided services are encapsulated in Docker containers. The overall software architecture of AGTD is shown in Figure 19.

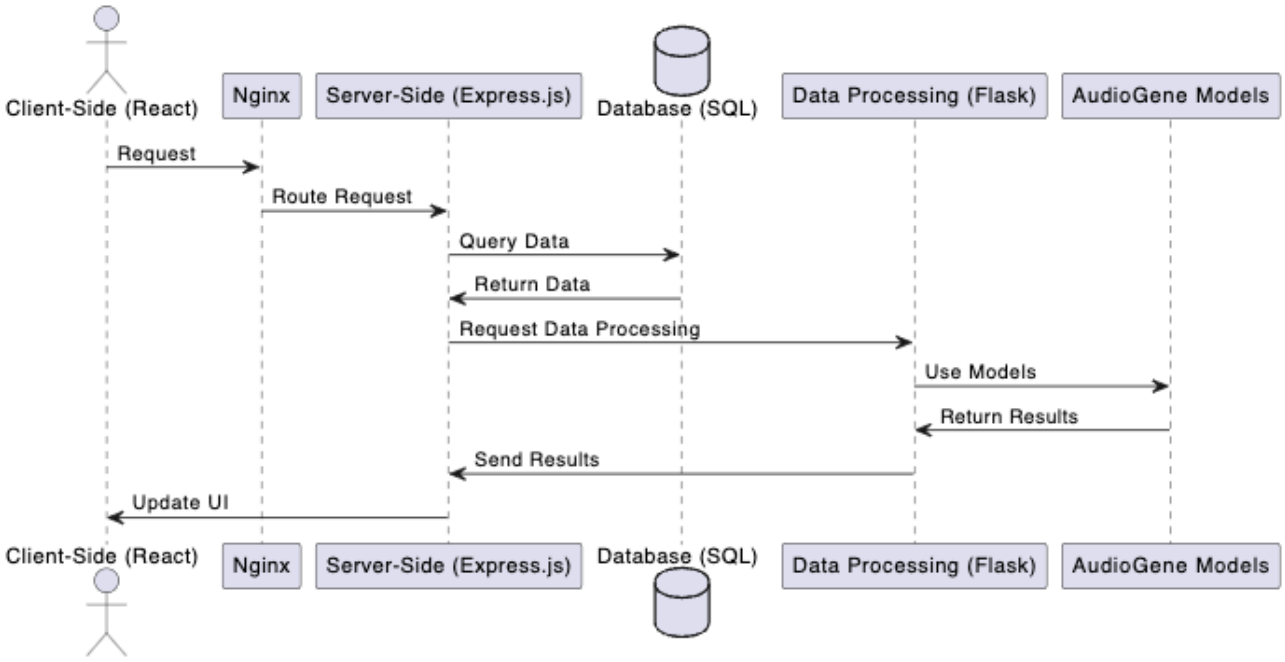

Figure 19. The Software Architecture of the AGTD shows client-to-server interactions and the server backend communications for data processing.

3.3.1. SQL Database

For the database backend, we chose to use SQL (Structured Query Language) as it is widely available and used in the development of modern web applications, as it facilitates efficient data storage and retrieval for dynamic content generation [6]. SQL is an open-source relational database management system (RDBMS) offered by Oracle™.

3.3.1.1. Database Structure and Schema

The AudioGene database schema was designed to suit the needs of the AudioGene platform (audiogene.eng.uiowa.edu). The underlying schema incorporates all the various entities and their interrelationships within the system, as shown in Figure 20.

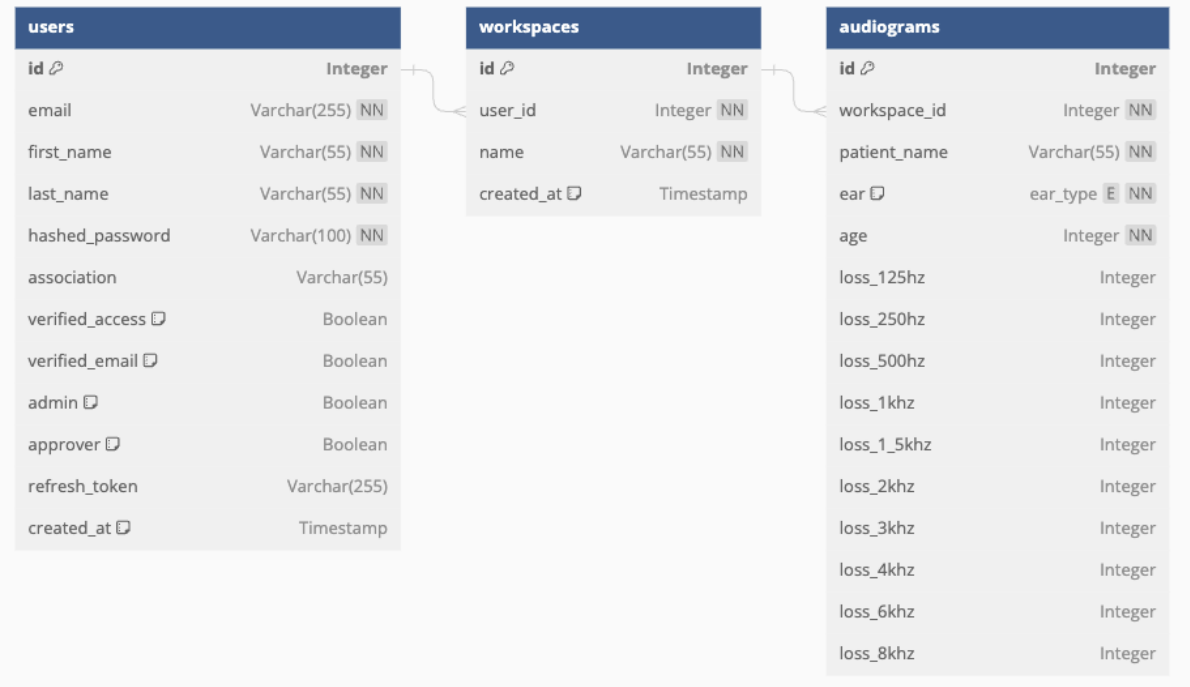

Figure 20. The schema of the AGTD SQL database shows the relationships among each of the tables.

3.3.1.1.1 Users Table

The `users` table is used for all user management within the platform. It holds the user's identification (name), authentication (password), and authorization details, including a unique email, encrypted password, and flags that denote permissions and roles within the application. Each user is associated with an ID, a unique identifier automatically incremented upon creating a new user record. This identifier acts as a foreign key (FK) in other tables, establishing relational integrity across the database.

3.3.1.1.2 Workspaces Table

To personalize the experience, the workspace table was introduced. Each workspace is linked to a user through the `user_id` FK and holds the name and creation timestamp. The `ON DELETE CASCADE` clause ensures that a user's deletion leads to removing their corresponding workspaces, thereby maintaining database consistency. This table captures all the user's runs within the model predictions.

3.3.1.1.3 Audiograms Table

A core function is in the audiograms table, which records detailed audiometric data. This includes patient identifiers, ear information, age, and hearing loss measurements across various frequencies. The workspace_id establishes a relationship with the workspaces table, allowing users to access and manage their patient’s audiograms within their dedicated workspaces.

3.3.2. Nginx Integration

Nginx acts as the gatekeeper for the AudioGene website, ensuring efficiency and security in handling web traffic to the site. It implements two important behavioral characteristics:

- Efficient Traffic Management As a reverse proxy, Nginx directs incoming traffic to the Express.js server configured for AudioGene [29]. It's configured to handle static content delivery and dynamic request routing, ensuring users experience fast load times and reliable access to the AudioGene functionalities.
- Secure Communication: The Secure Sockets Layer/Transport Layer Security (SSL/TLS) management capabilities of Nginx support encrypted data exchange with the AudioGene platform and safeguard user data from interception.

3.3.3. Express.js

Within the server-side architecture of AudioGene, Express.js serves as the backbone for handling HTTP requests and facilitating RESTful API development. It's a minimal and flexible Node.js web application framework that provides robust features for web and mobile applications [8]. AudioGene uses Express.js for various HTTP utility methods and middleware, as well as for its performance, as it provides only a thin layer of web application features without confusing Node.js features. It is also the connection layer between the front end and the more data-intensive tasks done in the Flask (backend) server.

3.3.3.1. Middleware Integration

Express.js integrates middleware that extends the application's capabilities [8]. For instance, the server's cookie parser is crucial in parsing cookie headers. This is particularly helpful for handling session data in the application. The server also uses express functions, `express.json`, and `express.urlencoded({ extended: true })` middleware is configured to parse incoming requests with JSON payloads and URL-encoded bodies so the server can process POST requests.

3.3.3.2. Session Management

AudioGene leverages `express-session` for managing sessions across different HTTP requests [8]. A unique session identifier, `userID`, maintains the session state, providing a secure mechanism for authenticating users throughout their interactions with the AudioGene platform. The session middleware's configuration includes session secret retrieval from environment variables, reinforcing the security of the user's sessions.

3.3.3.3. Static Routes and Dynamic API Endpoints

Express.js also manages static page routing, directing users to the appropriate views for different application parts. Dynamic API endpoints are constructed to handle various operations, such as user creation and authentication [8]. These endpoints cater to both public requests and those requiring token-based authentication, ensuring secure access to the application's functionalities. The endpoints can call further endpoints, such as the Flask server, all asynchronously.

3.3.3.4. Token Authentication

The server incorporates JWT (JSON Web Tokens) middleware for secure authentication [9]. This includes public endpoints that do not require user authentication and protected endpoints that utilize the `jwt.verify` function from the JSON web token library for validating access tokens, thereby securing routes that perform sensitive operations like user management.

3.3.4. React.js

Within the AudioGene tool, React.js is employed for its component-based architecture, which offers a modular approach to building the user interface (UI) [15]. Each component is a self-contained module that can maintain its state and logic, thereby promoting reusability. This architecture streamlines development by allowing individual components to be developed in isolation, enhancing the codebase's maintainability and allowing for rapid development.

3.3.5. Node.js

Node.js was chosen for the server environment due to its event-driven, non-blocking I/O model, which significantly enhances AudioGene's performance, particularly in handling concurrent data-intensive operations, such as calling the Flask server to predict the gene mutation

and processing the result of this operation. The application's back end, built with Node.js, is designed to handle multiple requests simultaneously with ease [10].

3.3.5.1. Event-Driven Architecture

The event-driven architecture of Node.js allows AudioGene to react to events in real-time, which is vital for the smoothness of our genetic data's interactive visualization. The server can process multiple streams of audiogram data, user interactions, and other events efficiently, providing a responsive experience to users.

3.3.5.2. Non-Blocking I/O

By utilizing non-blocking I/O operations, Node.js ensures that the server can handle I/O-bound tasks, such as database operations, without stalling the event loop [10]. This means that while a database query is being processed, Node.js can concurrently serve other requests, making the AudioGene server highly scalable.

3.3.5.3. Server Initialization

The application is configured to listen on a specified port, which can be defined in the environment variables, supporting both development and production environments, especially when deployed using containerization with Docker. Using this method allows for quick development without having to make many changes when switching from development to production.

3.3.6. Flask

Flask is a web framework that allows developers to build lightweight web applications quickly and easily with Flask Libraries and is one of the most popular Python-based web frameworks [14]. Our application used Flask’s capabilities to expose our Python functions as

web-accessible API, allowing for seamless interaction between client-side and server-side processes.

3.3.6.1. Framework Selection Rationale

The decision to use Flask was easy as it provided a simple framework with powerful capabilities. Flasks' ability to quickly set up routes to access our functions made it the ideal choice for our API development. It perfectly aligned with our need to perform complex data analytics, such as machine learning model building, data clustering, and preprocessing through API calls.

3.3.6.2. API Construction and Integration

Within our Dockerized environment, Flask is the intermediary between the computationally heavy backend and the web frontend. By defining routes, we can map specific HTTP requests to the Python functions that execute various tasks, such as running algorithms to process audiometric data or using AudioGene V4 or AudioGene V9.1 to make predictions.

For example, when a user initiates a request for data analysis, the call is first made to the Express.js server, which then routes it to the correct Flask route, which performs the analysis and sends the analyzed data back in JSON, which can then be dynamically rendered on the client side. This API-centric architecture ensures that AudioGene is responsive and interactive and separates data processing and user interface management.

3.3.7. Docker for Deployment

Docker, one of the numerous containerization platforms available, has quickly become an essential tool in the software development life cycle. It is known for its ability to encapsulate applications within containers to ensure consistent environments across different deployment

environments and systems [23]. By using Docker, developers can eliminate the “it only works on my machine” problem that so many face and provide a reproducible and isolated environment for each part of AudioGene.

In AudioGene, Docker containers were utilized to encapsulate and containerize various components within the application:

- AudioGene Website Container: This container orchestrates a seamless user experience by integrating the Express.js server with a pre-built React frontend. It manages client interactions, including authentication and session management. For model predictions and data analysis for visualizations, it delegates tasks to the Flask container designed for this purpose.
- Flask Container: The Flask server is held in this container and is responsible for performing computationally heavy tasks, including running AudioGene models, clustering, and preprocessing. Upon completion, it communicates results in JSON format to the Express.js server, which is then returned to the front end in the AudioGene Website Container.
- Database Container: An SQL database container maintains the persistence layer and stores user data.
- Nginx: An Nginx container acts as a reverse proxy, efficiently directing traffic between the client and.
- AudioGene Models Containers: AudioGene V4 and AudioGene 9.1 have separate containers with their specific run environment and are executable via Python commands.

- Preprocessor Containers: Containers for AudioGene V4 preprocessors (xlsToCSV converter, Perl preprocessor) encapsulate the specialized environments required to run this model.

The orchestration of these containers is managed by two Docker Compose files, which makes it easy to manage the services, networks, and volumes that make up AudioGene, and its tools [28]. The files are configured to use a shared Docker network, allowing the containers to communicate with each other seamlessly and securely using container IP addresses or container names. Illustrated in Figure 21 are the implemented containers, their interactions within each Docker Compose, and their shared network.

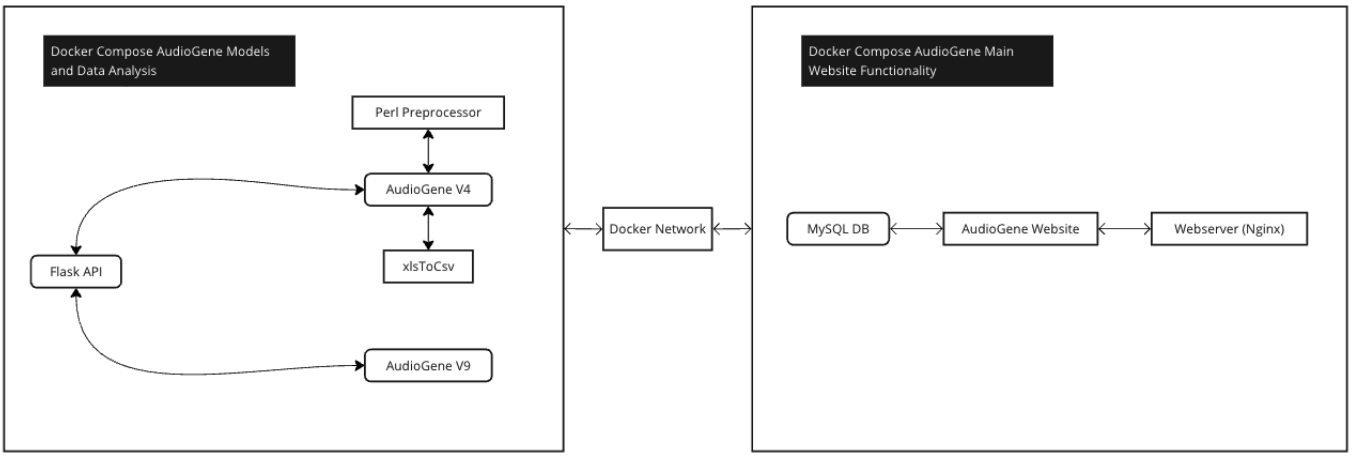

Figure 21. Showcases how the containers work together and communicate once each Docker Compose File is run.

CHAPTER 4: RESULTS AND DISCUSSION

Six case studies were evaluated, and they suggest that AGTD's new hybrid model, combining advanced visualizations and the AudioGene Models (AG4 and AG9.1), can assist in the process of determining a genetic diagnosis, as well as localization of mutations implicated as possible causes of ADNSHL, for a given patient.

4.1. Model Performance Analysis After Customization

The AGTD allows users to customize AG9.1 by filtering out genes known to be non-causative for the patient phenotype presented. It was observed that improved results could be obtained in the model's prediction accuracy. However, these improvements come at a cost of some other risks. If a user is not confident in the genes being filtered out, doing so could effectively eliminate the correct gene, causing the diagnosis to be incorrect and hindering AGTD's ability to increase the quality of diagnosis. However, if an expert user is confident about the filtering through analyzing the tools and visualizations AGDT provides, then the model's performance is likely to increase, and the likelihood of diagnosing the correct gene increases. For example, without any user-specified customization, the reported top-3 accuracy of the AG9.1 model is 77.8% from LOOCV [1]. If, for example, the user was confident in filtering out *GRHL2*, *POU4F3*, and *SLC17A8* after carefully evaluating the audio profiles, Spatial Analysis, Clustering Views, and Ethnicity Pie Chart, the top-3 accuracy is 80.7%, showing improved prediction capabilities using the filtered genes. This is just one such case of these results, and with 8,388,607 possible results ($C = 2^{23} - 1$), we did not perform tests for every combination. Through this example, we can see the power that this model customization could potentially bring to the user.

4.2. Case Studies and Clinical Implications

In evaluating the potential of AGTD in improving the diagnostic process of the genetic mutation underlying ADNSHL, we present 6 case studies; 5 of the cases show promising results when using AGTD, and one shows where AGTD can fail. We selected these case studies to illustrate both AudioGene V4 and AudioGene V9.1 (3 cases of each) to better illustrate what each model can provide. We also tried to include a variety of genes from each subclass (small, medium, and large) within our dataset. However, finding a small gene that served as a good example proved to be impossible, bringing home the point that there will always be limits to predicting in an imbalanced dataset. The same "appears" to be the case with human predictors (clinicians). The approach taken for each case study was designed to accomplish four things:

1. We evaluate the top three patient predictions, determine the subclasses for each (small, medium, or large), and assess our confidence in the predictions.
    a. If one of the top three predictions contained a small gene or a smaller medium gene, we logged this as out of the ordinary as well as something to be examined further.
        i. If this had been done in AudioGene V4, we would have given it more credence than AudioGene V9.1 because AudioGene V4 outperformed AudioGene V9.1 significantly in small subclass gene predictions.
    b. If not, we began with the assumption that this gene was in the top three predictions due to the better accuracy displayed in bigger genes by both models, as seen in Figure 22.

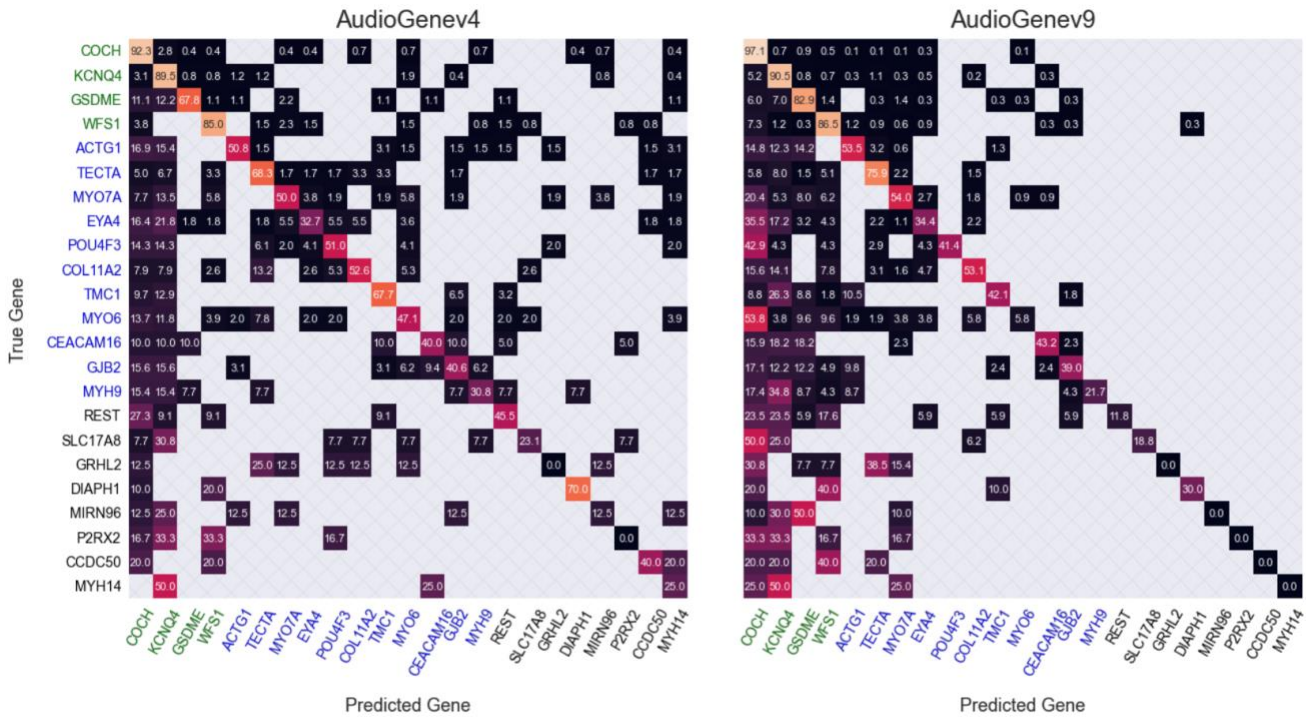

Figure 22. Comparative Analysis of Classification Accuracy in AudioGenev4 and AudioGenev9 using Leave-One-Out Cross-Validation. The figure illustrates the classification frequency of each gene as predicted accurately by both models. The frequency is normalized per row and scaled to a percentage with the darker colors being worse accuracy and the lighter being better. Absent values are indicated with hashes. Genes are color-coded by category for enhanced clarity. [1].

2. We examine the audio profiles for each of the top three predictions and analyze the aggregated audiogram trends for each age group compared to the selected patient.
    a. If the patterns displayed (up-sloping, down-sloping, or cookie-bite) of the aggregated audiograms compared to the patient are not followed in the top predictions (to a variable degree), then this suggests that the Spatial Analysis and Clustering tool needs to be examined further.
    b. In the case the aggregated audio profile pattern of one of the top three predicted genes shows great similarity to the patient's audiogram(s), we

would use this as supporting evidence toward choosing the gene mutation most likely causing the hearing loss.

3. We examine the spatial analysis and clustering tool to visually determine how well the patient's audiogram(s) is represented compared to nearby classified audiograms.
    a. In performing the analysis on a patient's audiograms, we first look at their cluster assignment and their closest genes as determined by KNN. We then visually examine all the genes in that cluster as well as their relative positions compared to our patient's audiogram. We would also examine the Greedy Clustering to determine whether there is any correspondence between the cluster assigned to the patient and the gene assigned to that cluster. If we find that one of the patient's audiogram's closest genes contains one of the top three predictions or that the surrounding audiograms are predominately one of the top three genes, we then use this as supporting evidence for that gene. However, if not, we use this as an unsupported case in which the correct gene is presumed to be within the top three predictions.
4. Finally, if the patient's ethnicity or geographical origin is available, we would use this data to possibly underscore the case that the correct gene is likely among the top three predictions.

4.2.1. Case Study 1 – MYO7A - Patient ID 5

In this first case study, we analyzed the results of a Patient (ID 5) diagnosed with *MYO7A* using two audiograms collected from the patient. AudioGene V4 predicted this case because the patient had more than one audiogram. The predictions for the patient (ID 5) can be seen below in Figure 23.

Figure 23. AudioGene V4 predictions for the patient (ID 5), with the top three genes being *MYO7A*, *EYA4*, and *WFS1*.

In examining the top three genes, we first considered the audio profiles of the top three genes and examined how they correlated with the patient's audiograms. In this analysis, we conclude three things. First, we found that *MYO7A* has a relatively similar profile to the patient's audiograms, matching similar dBLoss values for the first four frequencies. However, it tends to diverge into a down-sloping audiogram pattern in the higher frequencies instead of the up-sloping shape usually shown by *MYO7A* in that age range. Nevertheless, even with this apparent deviation, the difference is approximately only 10 dB loss on average, which can be seen in Figure 24, suggesting only a moderate difference that is not great enough to create doubt regarding *MYO7A* as the correct gene prediction.

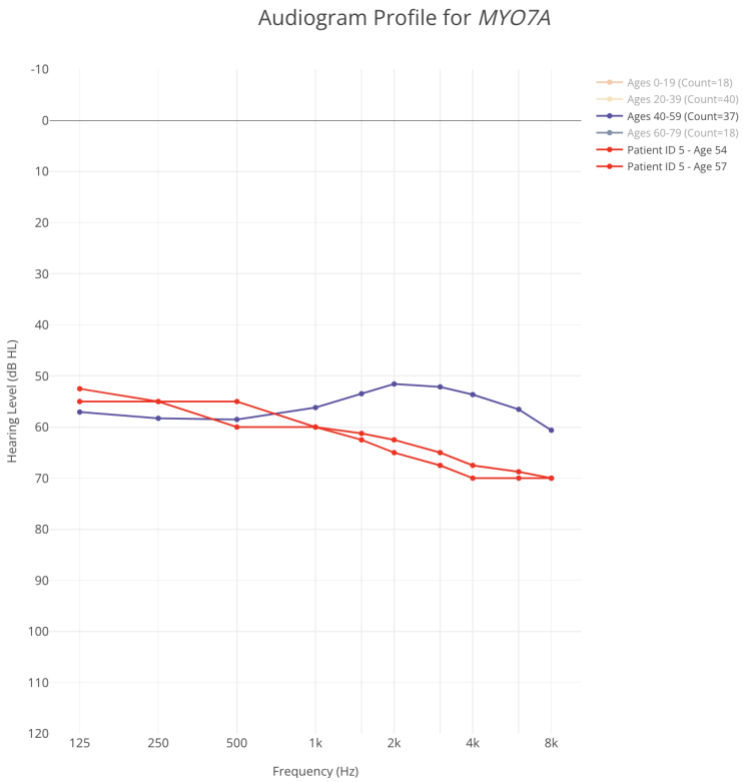

Figure 24. Audio Profile of *MYO7A* with the patient's (ID 5) audiograms in red, taken at 54 and 57 years of age, respectively.

Second, we find that *EYA4*, the second predicted gene, correlates more closely to the patient's audiogram, leading to initial suggestions that it may be *EYA4* instead of *MYO7A*. This correlation can be seen in the audio profile of *EYA4* in Figure 25.

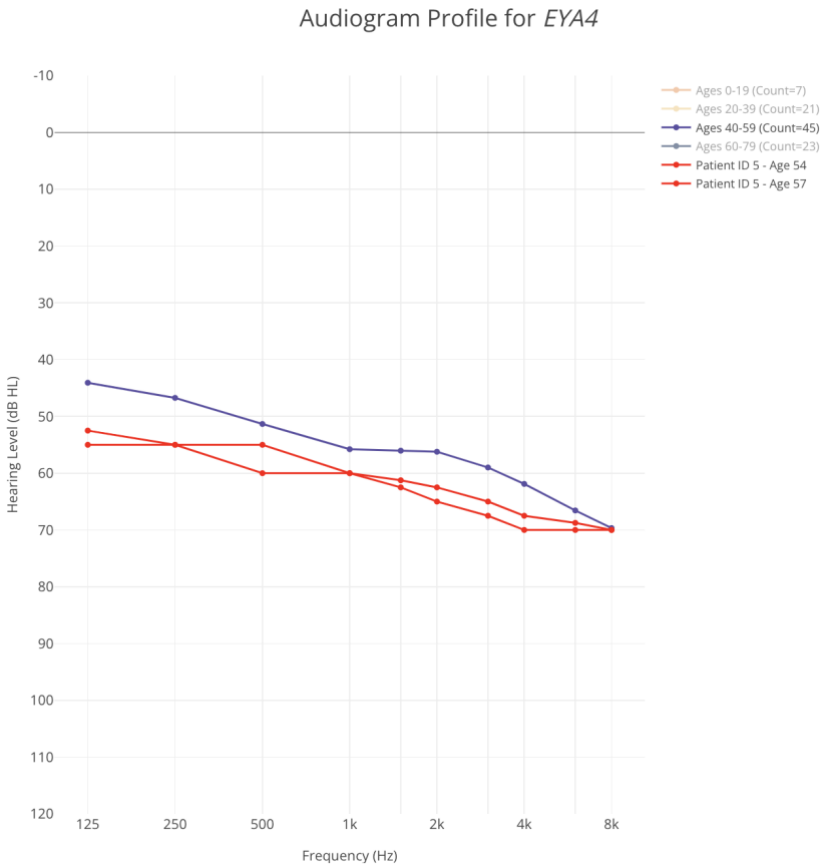

Figure 25. Audio Profile of *EYA4* with the patient's (ID 5) audiograms in red, taken at 54 and 57 years of age, respectively.

Finally, for the audio profile examination, we consider the last of the top three predictions, *WFS1*, where the audio profile tended to be more up-sloping and was visibly different from the patient's audiograms compared to the top two predicted genes, as shown in Figure 26. This suggests that *WFS1* would indeed be the least likely of the top three.

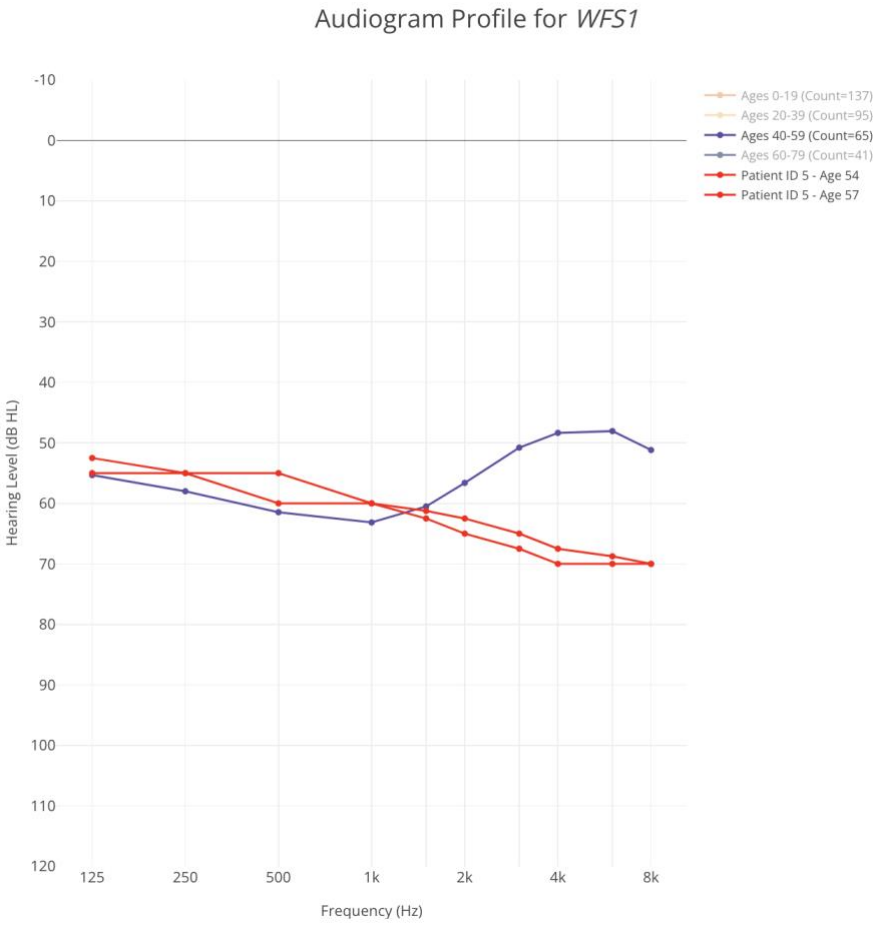

Figure 26. Audio Profile of *WFS1* with the patient's (ID 5) audiograms in red, taken at 54 and 57 years of age, respectively.

Continuing with our assumption that the correct gene is most likely either *MYO7A* or *EYA4*, we examine the clustering and see that our patient's (ID 5) audiogram is surrounded by mostly *MYO7A*, *EYA4*, *WFS1*, and *COCH*, as seen in Figure 27.

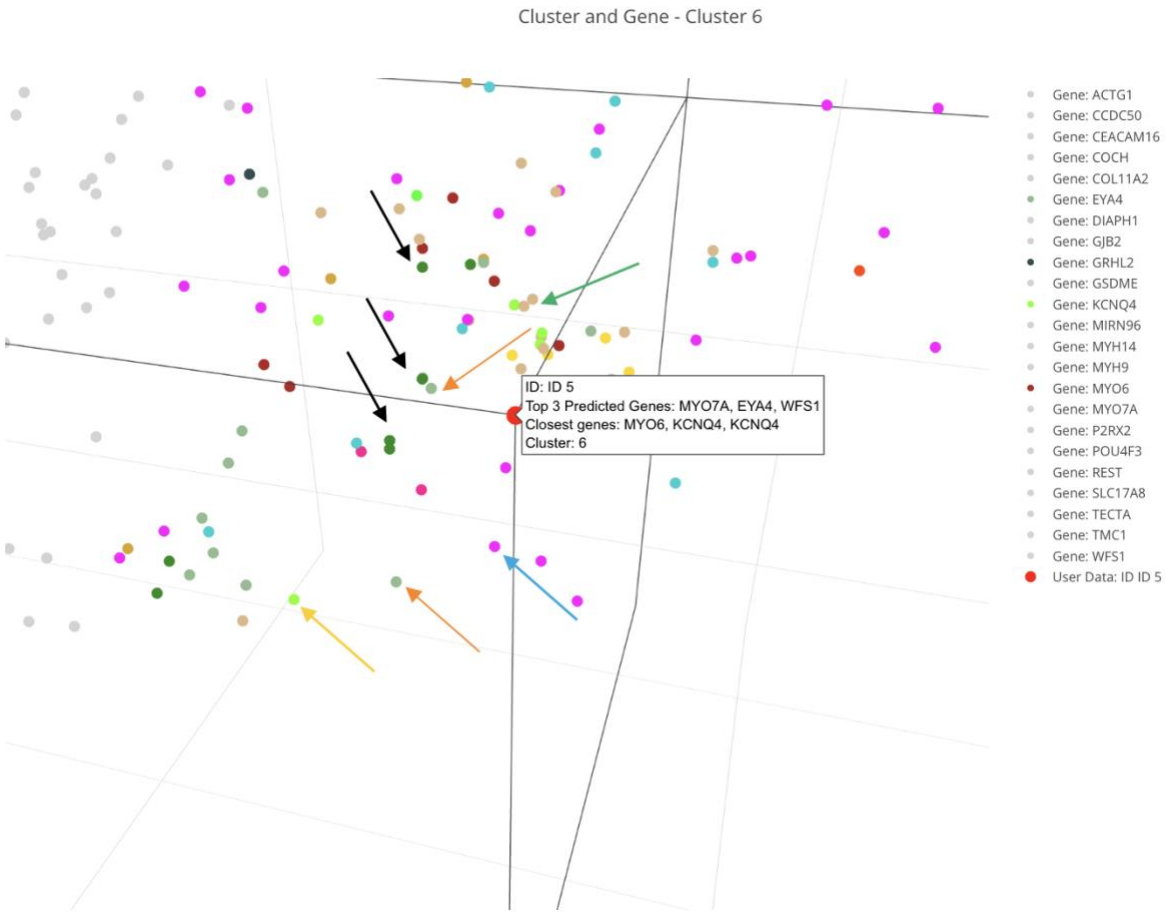

Figure 27. Three-dimensional plot of audiograms in the training set converted into 3 dimensions, with genes in cluster 6 colored (genes not in cluster are light grey). Patient (ID 5) is the red dot hovering over the displayed label. The black arrows point to *MYO7A* (green), the orange arrows point to *EYA4* (pale green), the green arrow points to *WFS1* (pale yellow), the blue arrow points to *COCH* (pink), and the yellow arrow points to *KCNQ4* (light green).

At first glance, the clustering may appear chaotic; however, upon further visual analysis of the surrounding genes and examining KNN distance calculations, we find that the genes nearest to our patient closely follow our top three predictions. We can see that the genes calculated to be the closest via the KNN metric are *MYO6* and *KCNQ4*; however, upon examining Figure 23, we can see that *MYO6* is the sixth prediction, with *KCNQ4* being the fourth. We want to examine this finding further to understand why *KCNQ4* is only the fourth prediction. To perform this analysis, we examine the audio profiles and find that *KCNQ4* has a

steep down-sloping characteristic and does not correspond well with our patient's audiogram, suggesting that *KCNQ4* should be ruled out. Next, looking at the top three predictions and their proximity to the patient's audiogram, we find that while *WFS1* does have some points in proximity to our patient, its audio profile does not follow the same trend, suggesting that it is likely not the correct gene. Looking next at *EYA4*, we can see some points near the audiogram; however, instances of *MYO7A* appear to outnumber them by a modest degree.[1] Due to *MYO7A* being the top prediction and slightly outnumbering *EYA4* in terms of points closest to the patient's audiogram, as well as its audio profile being reasonably close to the patient's audiograms, this suggests that *MYO7A* is most likely the correct prediction, as *EYA4* lacked sufficient evidence to overturn the decision. This use case shows the power of AGTD in helping clinicians to decide between the top three genes and to give them greater confidence in their diagnosis.

4.2.2. Case Study 2 – TECTA – Patient ID 14

For this second case, we examined the results of Patient (ID 14), diagnosed with *TECTA*, using the two audiograms collected. Due to the patient's multiple audiograms collected, AudioGene V4 was used for the predictions. The predictions for this patient are shown in Figure 28 below.

[1] We note here that at the time of this thesis's publication, it was impossible to normalize the display of points in the 3D plot. The consequence is that in each case where a conclusion or inference is drawn from counting the number of points near the patient point, bias is possible due to more data points for some genes than others. In future work, this deficiency should be addressed.

ID 14	TECTA	KCNQ4	EYA4	MYO7A	COL11A2	ACTG1	CEACAM16	GRHL2

Figure 28. Gene Predictions were made using AudioGene V4 for the patient (ID 14), with the top three genes being *TECTA*, *KCNQ4*, and *EYA4*.

In examining the top three genes predicted by AudioGene V4, we first considered each gene's audio profile. Looking at *TECTA* first, the audio profile shown in Figure 29 shows that the patient's audiograms are visibly similar to the cookie-bite shape as typically presented in *TECTA*, making us more confident that *TECTA* is a possibility for the correct gene diagnosis.

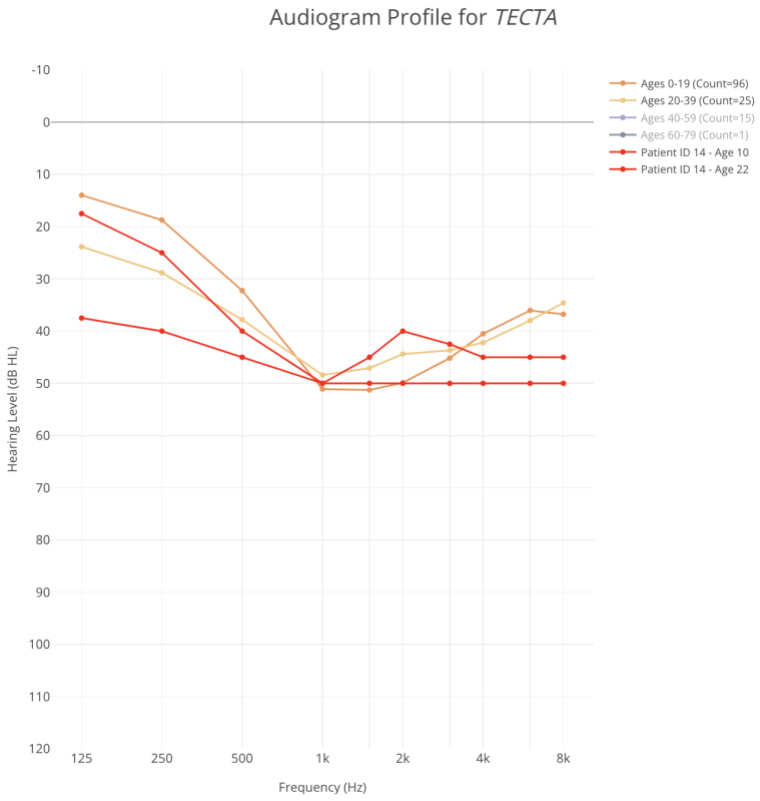

Figure 29. Audio Profile of *TECTA* with the patient's (ID 14) audiograms in red, taken at 10 and 22 years of age, respectively.

Next, we examined the audio profile of *KCNQ4* and found that while its shape follows the down-sloping character of the patient's audiograms presented, it is significantly steeper, as

shown in Figure 30. This suggests that *KCNQ4* is perhaps less likely to be the correct gene than *TECTA*.

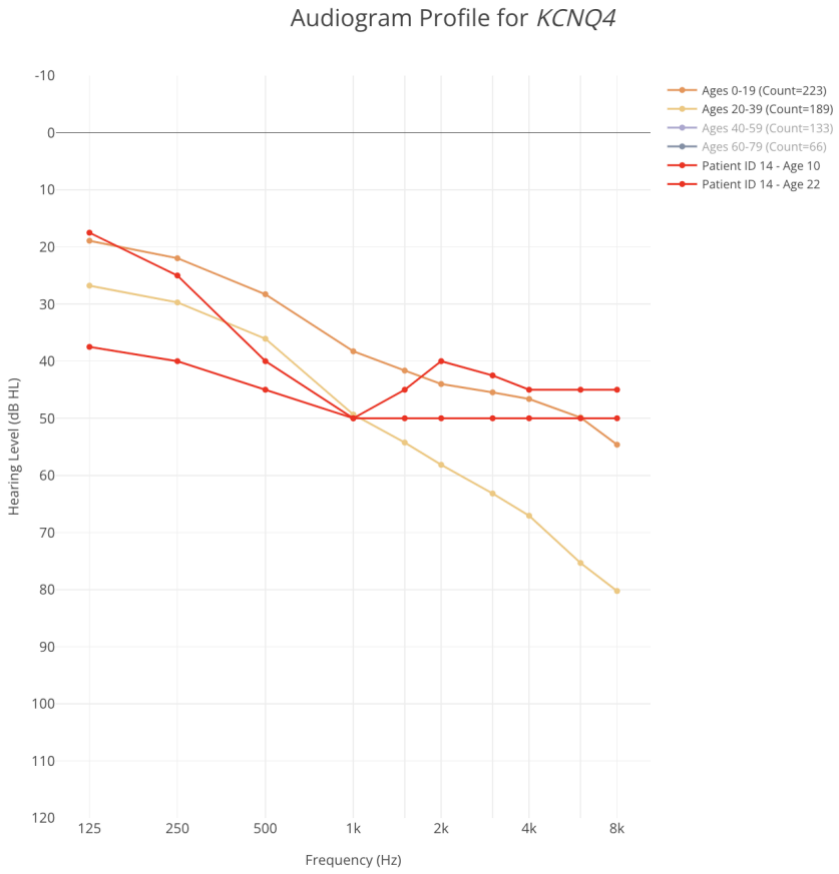

Figure 30. The audio profile of *KCNQ4* with the patient's (ID 14) audiograms is in red, taken at 10 and 22 years of age, respectively.

Finally, for the audio profile examination, we considered the third prediction, *EYA4*, where we found that while it does have a similar downward slope initially, there is a noticeable difference around 1K HZ, where the patient's audiograms no longer continue to downslope, whereas *KCNQ4* does. This can be seen in Figure 31. The difference is not enough to rule out *KCNQ4*, but it does place it behind *TECTA* and before *KCNQ4* in the subjective likelihood ranking. This leads us to believe that *TECTA* is the most likely gene prior to considering the clustering analysis.

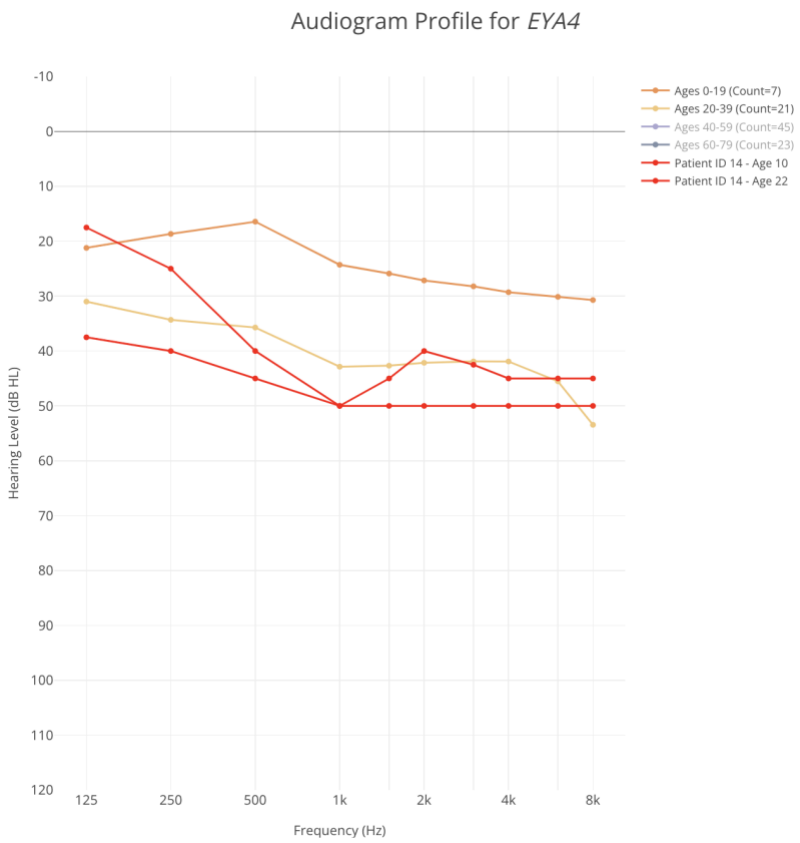

Figure 31. Audio Profile of *EYA4* with the patient's (ID 14) audiograms in red, taken at 10 and 22 years of age, respectively.

Continuing our assumption that *TECTA* is the correct gene, we further analyzed the clustering, as shown in Figure 32.

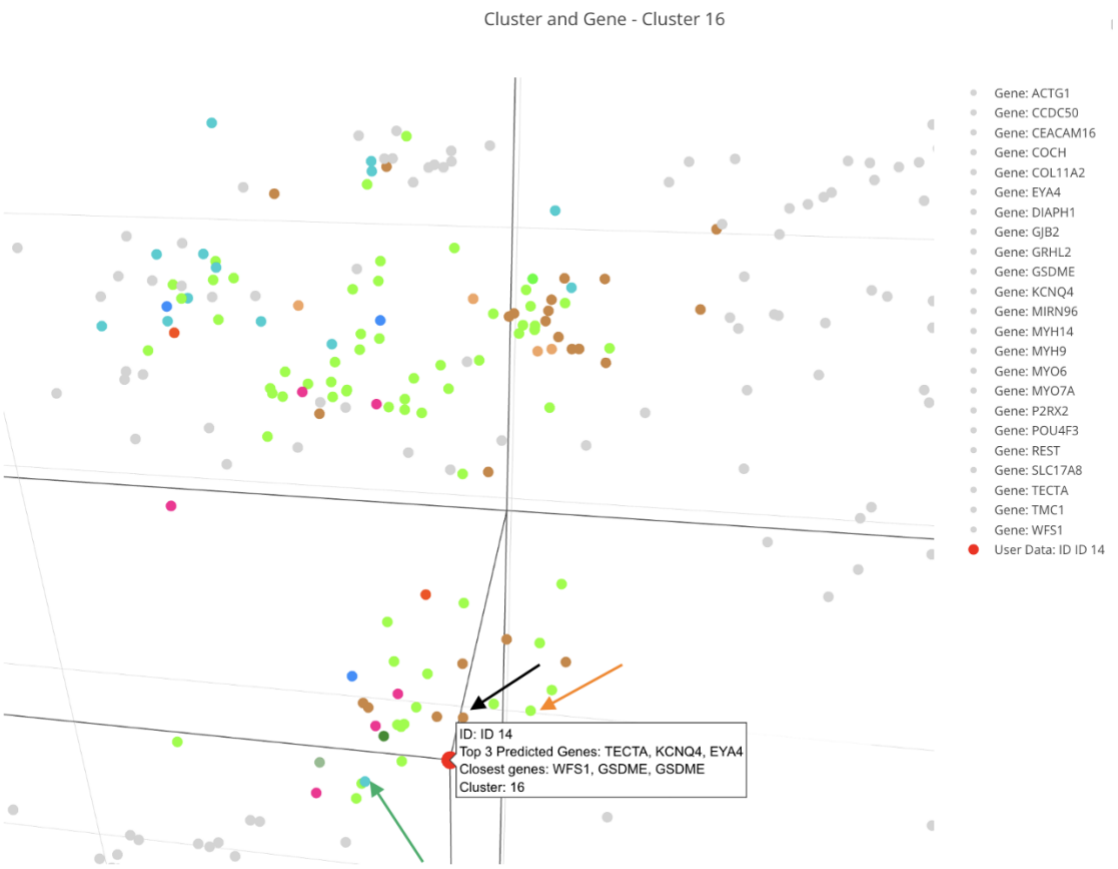

Figure 32. Three-dimensional plot of audiograms in the training set converted into three dimensions, with genes in cluster 16 colored (genes not in cluster are light grey). The patient (ID 14) is the red dot hovering over the displayed label. The black arrows point to *TECTA* (brown), the orange arrows point to *KCNQ4* (bright green), and the green arrow points to *GSDME* (light blue).

Looking at the clustering, we first see that the closest genes to our patient's (ID 14) audiogram, via the KNN distance calculations, are *WFS1* and *GSDME*. However, glancing at Figure 28, we can rule these genes out, as they are outside the top eight predictions. Next, we visually examined the cluster, as seen in Figure 32. Looking at the patient's audiogram, we can see that it is surrounded mainly by *TECTA* (brown) and *KCNQ4* (bright green). Considering our earlier analysis of the audio profile of *KCNQ4*, having the least similar shape to the patient's audiograms and *TECTA* having the most similar, we concluded that *TECTA* was likely the correct genetic diagnosis, which is the actual genetic diagnosis. This case is another indicator of

the power that AGDT can provide clinicians by distinguishing top predictions and enhancing confidence in the diagnosis.

4.2.3. Case Study 3 – GJB2 – Patient ID 43

For this third case study, we analyzed the results of the Patient (ID 43), who was diagnosed with a *GJB2* genetic mutation. The patient's data includes only one audiogram; hence, AudioGene V9.1 was used to make the predictions, with the top predictions shown below in Figure 33.

Figure 33. Gene Predictions were made using AudioGene V9.1 for the patient (ID 43), with the top three genes being *GJB2*, *TMC1*, and *GSDME*.

In examining the top three genes predicted by AudioGene V9.1, we first analyzed the audio profiles of each of the top three genes: *GJB2*, *TMC1*, and *GSDME*. Initially, we looked at the audio profile for the top predicted gene, *GJB2*, compared to the patient's (red line) audiogram, as shown in Figure 34. Visually, we can see that the audiogram tends to follow a similar pattern of moderate down-sloping as presented in *GJB2*. This was a good indicator that this gene is a good candidate and should be considered further.

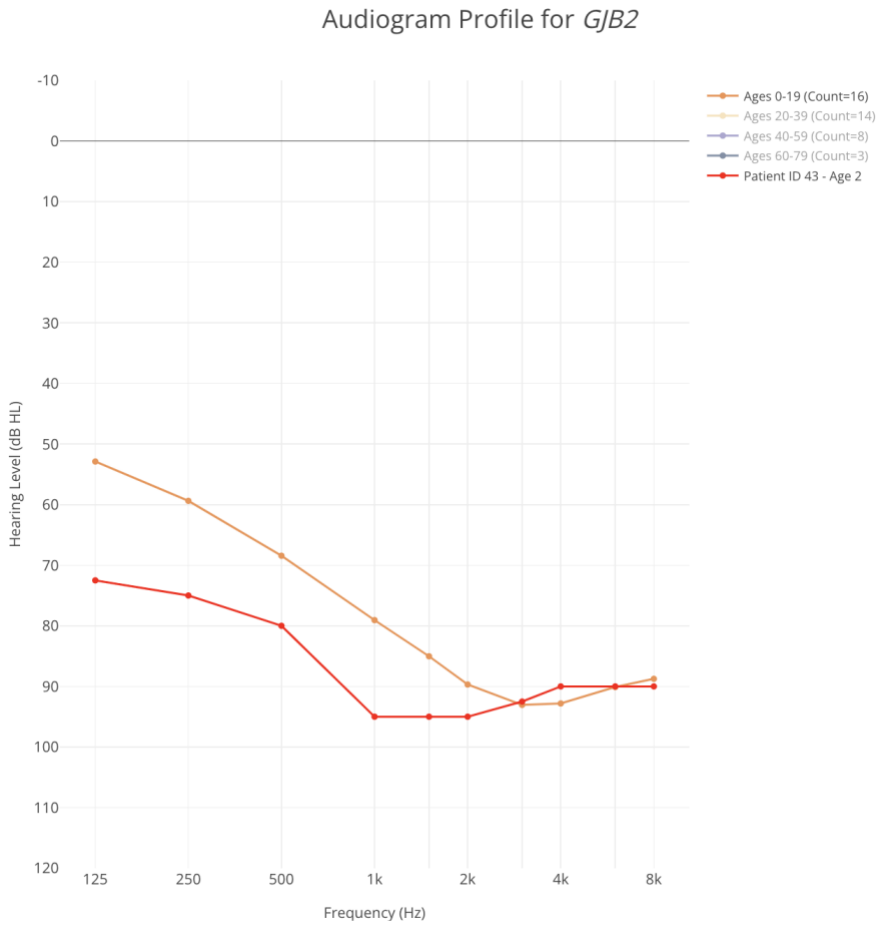

Figure 34. Audio Profile of *GJB2* with the patient's (ID 43) audiogram in red, taken at two years of age.

Next, we looked at the audio profile of the second top predicted gene, *TMC1*, as shown in Figure 35. Looking at its audio profile, we see that *TMC1* has a more pronounced down-sloping trend than the patient's audiogram, as well as there being a clear distinction in the dBLoss, specifically at lower frequencies, thereby casting doubt upon the hypothesis of *TMC1* being the correct gene.

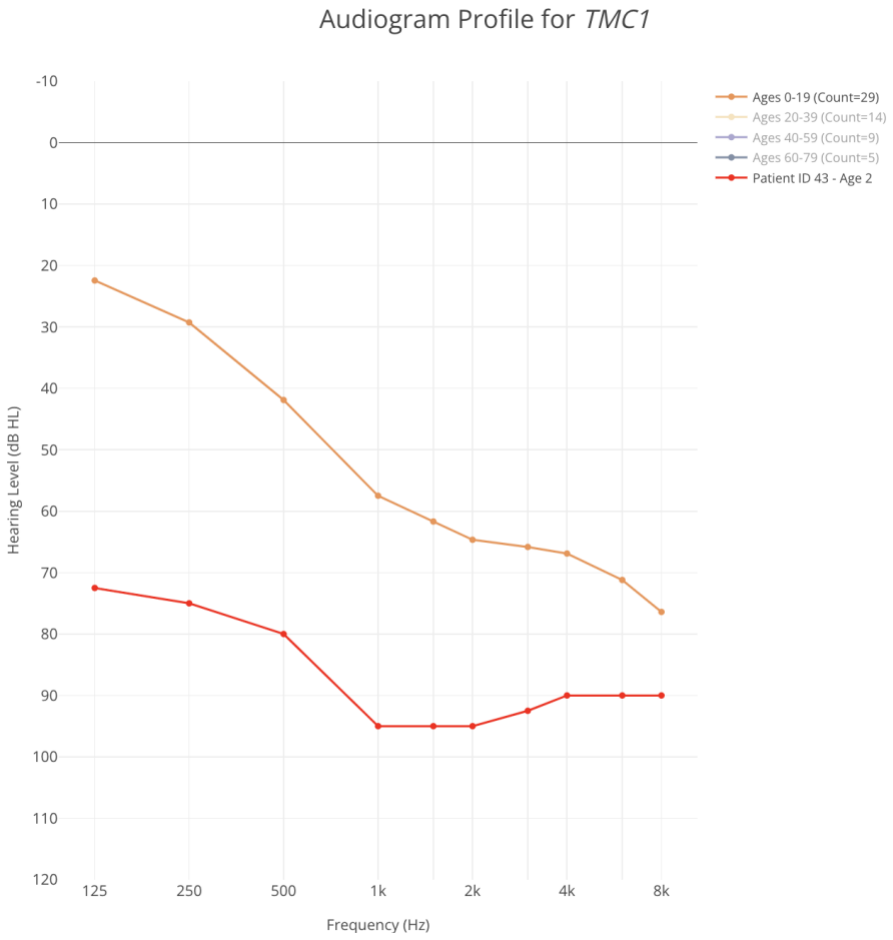

Figure 35. Audio Profile of *TMC1* with the patient's (ID 43) audiogram in red, taken at two years of age.

Finally, for the audio profile examination step, we inspect the audio profile for the third predicted gene, *GSDME*, which can be seen in Figure 36. The first evident observation is the significant separation in dBLoss along all frequencies. While it does have a similar shape at lower frequencies compared to the patient's audiogram, this does not appear significant enough to suggest that *GSDME* would be in the correct gene. Therefore, this implies that *GJB2* is the most likely candidate prior to the clustering analysis.

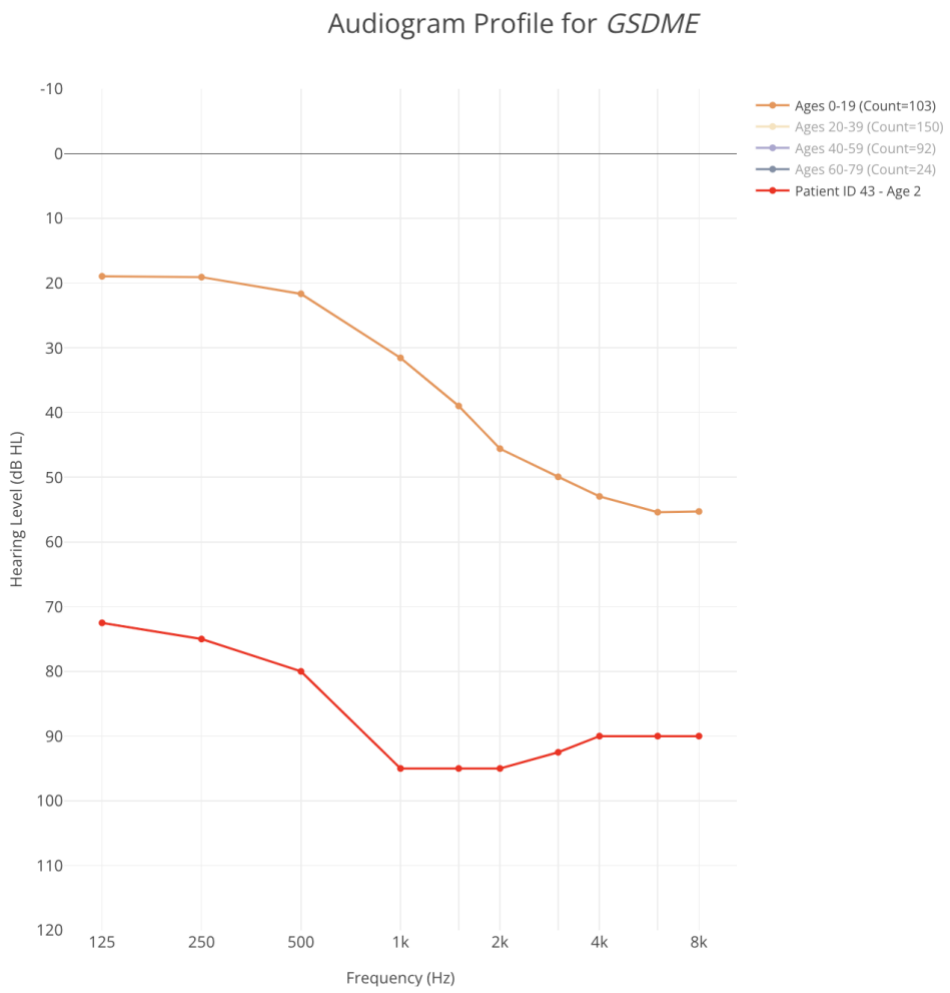

Figure 36. Audio Profile of *GSDME* with the patient's (ID 43) audiogram in red, taken at two years of age.

Continuing our assumption that *GJB2* was the correct gene, we performed further analysis using the clustering shown in Figure 37.

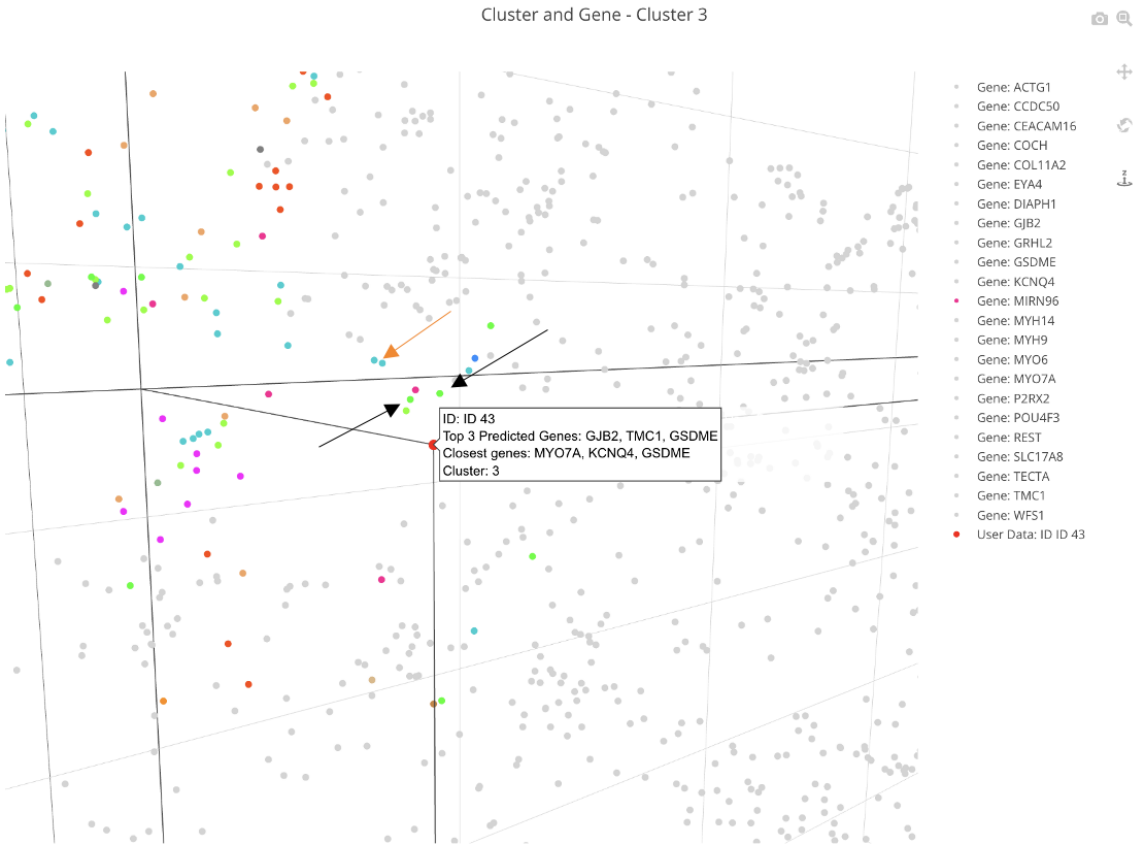

Figure 37. Three-dimensional plot of audiograms in the training set converted into three dimensions, with genes in cluster 3 colored (genes not in cluster are light grey). The patient (ID 43) is the red dot hovering over the displayed label. The black arrows point to *GJB2* (bright green), and the orange arrows point to *GSDME* (light blue).

Upon examination of the clustering interface, we can see that the closest genes from the distance calculation using KNN are *MYO7A*, *KCNQ4*, and *GSDME*. Glancing at Figure 33, we can see that *MYO7A* and *KCNQ4* were outside the top six predictions, and their audio profiles did not correlate to the patient's (ID 43). Therefore, we can most likely rule them out, as it is unlikely that they are the correct prediction. Next, we examined the cluster visually and determined that the closest genes are *GJB2* (bright green) and *GSDME*. As previously stated, the audio profile of *GSDME* did not correlate to that of the patient (ID 43), but *GJB2* did. From this analysis, AGTD suggests that *GJB2* should be the correct gene, therefore supporting our initial

hypothesis and diagnosing the correct gene, highlighting the ability of AGTD to suggest the correct gene accurately, even when it is not the first overall prediction.

4.2.4. Case Study 4 – WFS1 – Patient ID 13

In our fourth case, we explored the results of a Patient (ID 13) diagnosed with a *WFS1* genetic mutation. The patient had three audiograms collected; therefore, AudioGene V4 was used to make the predictions shown in Figure 38.

Figure 38. Gene Predictions were made using AudioGene V4 for the patient (ID 13), with the top three genes being *TECTA*, *WFS1*, and *COL11A2*.

In examining the top three genes predicted by AudioGene V4, we first analyzed the audio profiles of each of the top three genes: *TECTA*, *WFS1*, and *COL11A2*. In our examination, we first looked at the top predicted gene, *TECTA*. The audiogram for our patient showed a similar pattern to *TECTA*, as they both have a moderate downslope until the middle frequencies and upslope shortly after, as shown in Figure 39. This comparison presents a promising supportive case for *TECTA* being the correct gene.

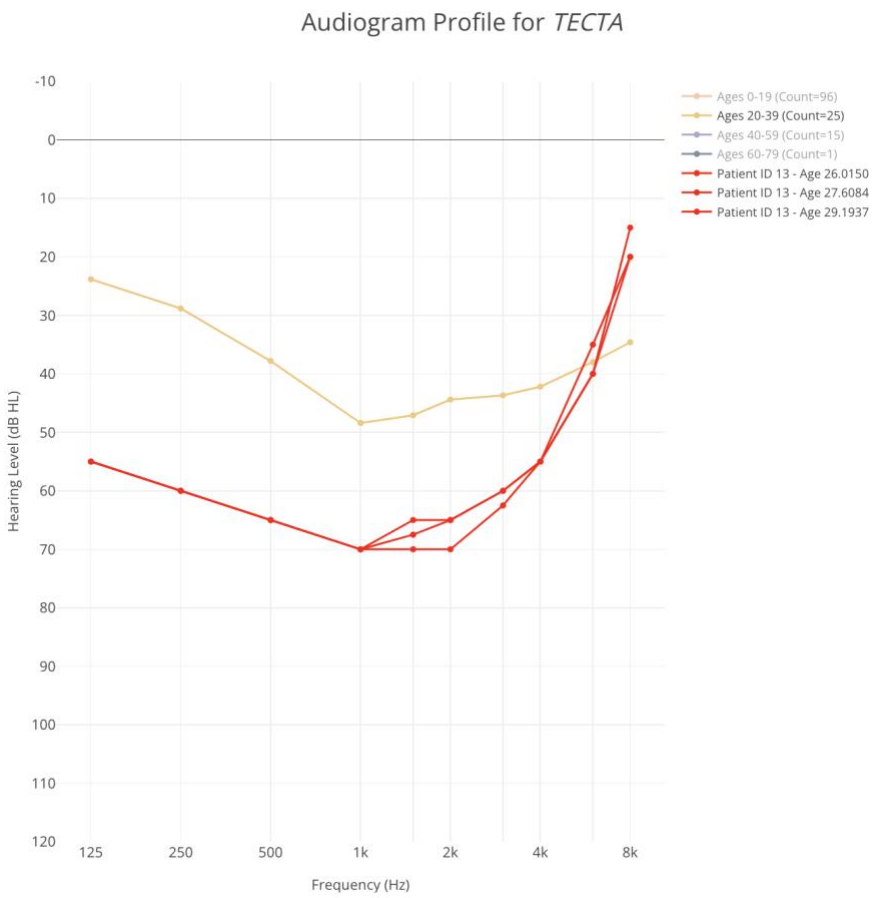

Figure 39. Audio Profile of *TECTA* with the patient's (ID 13) audiograms in red, taken at 26, 27, and 29 years of age, respectively.

Next, we looked at the audio profile of the second predicted gene, *WFS1*, showcased in Figure 40. In our examination, we can quickly discern that the shape for *WFS1* appears to follow a similar shape to our patient's audiograms, where it is down-sloping until the middle frequencies and then up-sloping afterward. In contrast to *TECTA*, *WFS1* appears to have a better correspondence to the patient's audiogram, providing evidence to suggest that the gene could be either one and needs further investigation.

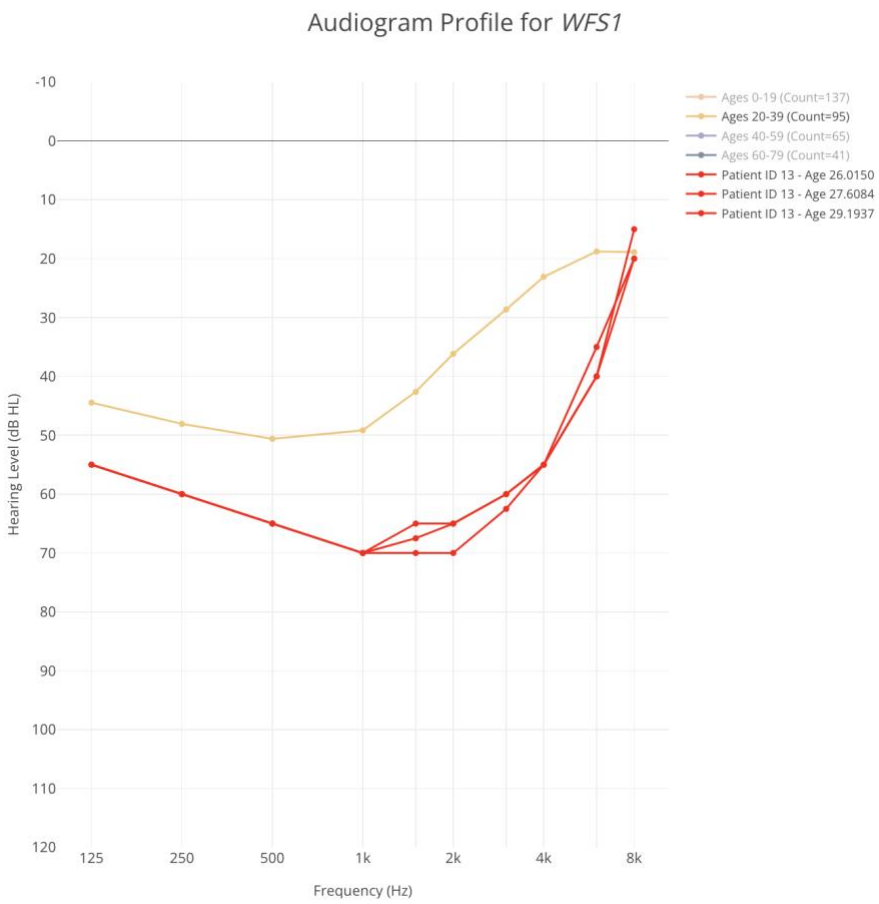

Figure 40. Audio Profile of *WFS1* with the patient's (ID 13) audiograms in red, taken at 26, 27, and 29 years of age, respectively.

Finally, for the audio profile examination, we looked at the audio profile of the last of the top three predicted genes, *COL11A2*, as shown in Figure 41. The audio profile shows that *COL11A2's* shape is cookie-bite, a contrast to the patient's audiograms. Therefore, considering the last two audio profiles for the other top three genes, we determined *COL11A2* to be the least likely.

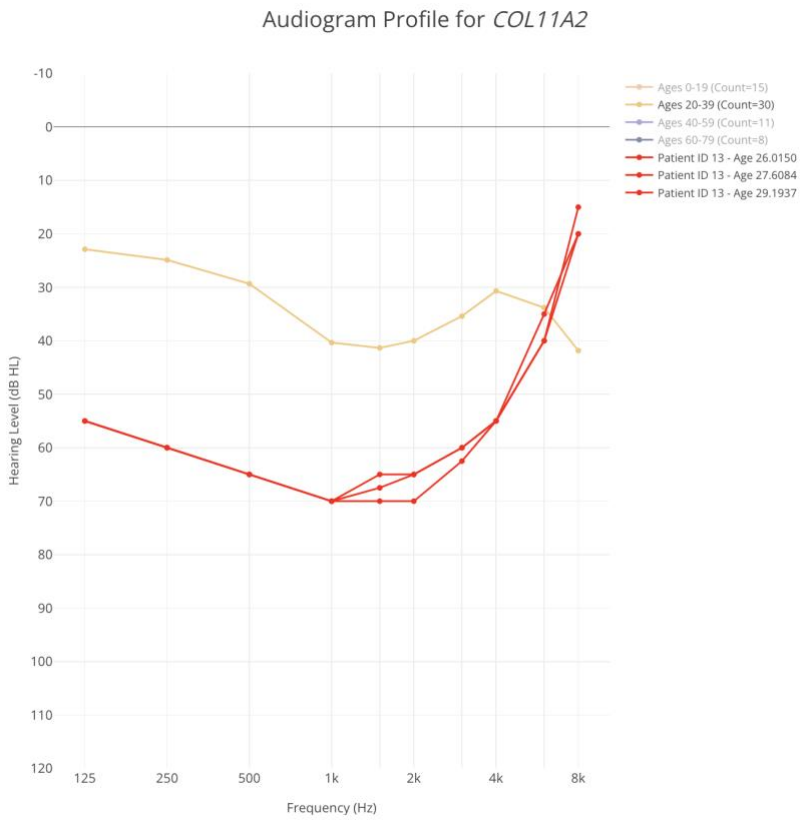

Figure 41. Audio Profile of *COL11A2* with the patient's (ID 13) audiograms in red, taken at 26, 27, and 29 years of age, respectively.

Continuing our assumption that either *TECTA* or *WFS1* is the correct gene, we further analyzed the clustering shown in Figure 42.

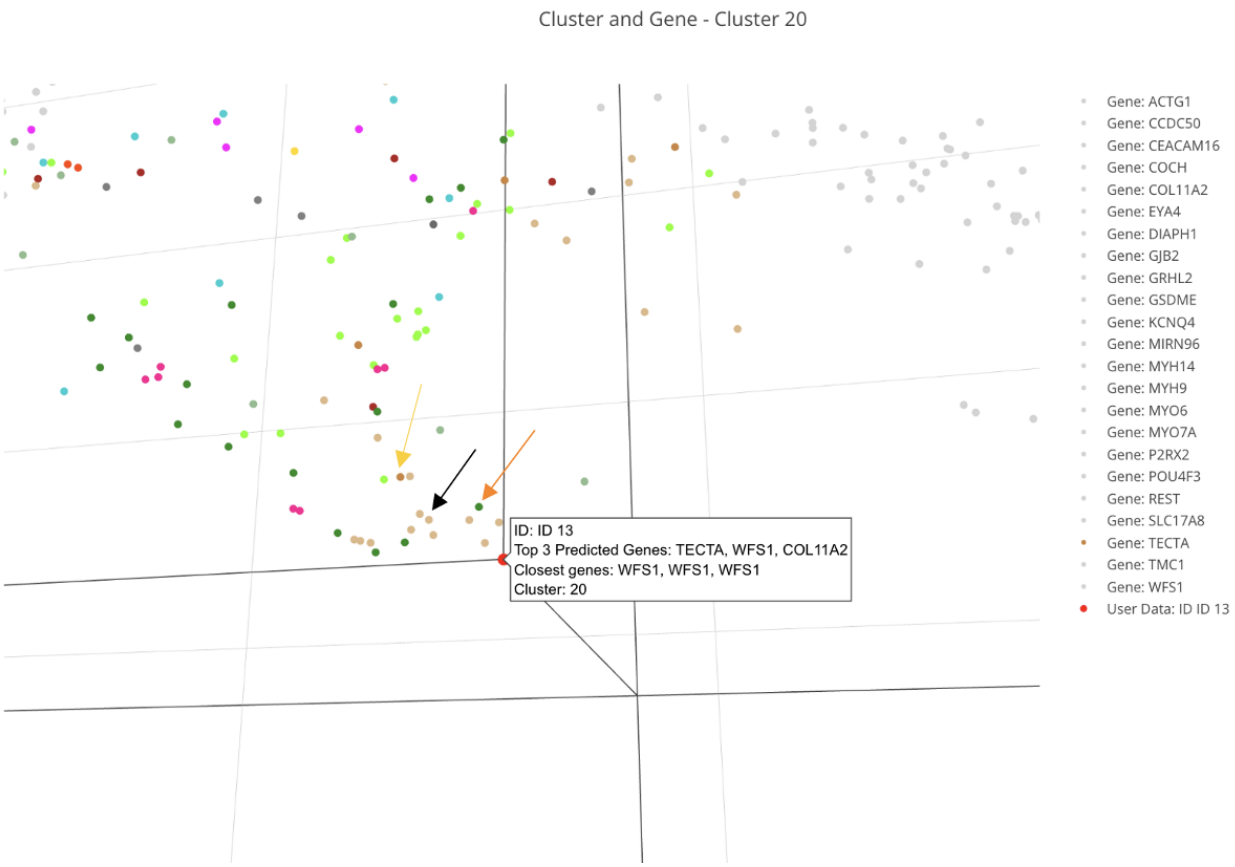

Figure 42. Three-dimensional plot of audiograms in the training set converted into three dimensions, with genes in cluster 20 colored (genes not in cluster are light grey). Patient (ID 13) is the red dot hovering over the displayed label. The black arrow points to *WFS1* (light brown), the orange arrow points to *MYO7A* (green), and the yellow arrow points to *TECTA* (brown).

Upon examination of the clustering interface, we see that the visually closest and KNN closest genes are all *WFS1*. The only other gene present in the immediate area is *MYO7A*, which, in examining its audio profile, we found to have a cookie-bite shape, a contrast to the shape of all of our patients' audiograms. Furthermore, looking at Figure 38, we can see that *MYO7A* is the eighth predicted gene, which suggests that *MYO7A* would not be the correct diagnosis. From this analysis, we also reconsidered *TECTA*, as although its audio profile was promising, it only appears to be a far neighbor of the patient's audiogram, whereas *WFS1* surrounds it. Therefore, using AGTD, we would suggest that the correct diagnosis would be *WFS1*, which would be

accurate. This case showed AGTD's improvement in diagnostic confidence, especially in cases where the correct gene is not the first prediction.

4.2.5. Case Study 5 – COCH – Patient ID 12

For this fifth case study, we examined a Patient (ID 12) diagnosed with *COCH*, our most significant gene and 1/3 of all our training data. The patient had only one audiogram collected, so AudioGene V9.1 was used for the predictions shown in Figure 43.

Figure 43. Gene Predictions were made using AudioGene V9.1 for the patient (ID 12), with the top three genes being *COCH*, *EYA4*, and *MYO7A*.

In examining the top three genes predicted by AudioGene V9.l, we first analyzed the audio profiles of each of the top three genes: *COCH*, *EYA4*, and *MYO7A*. We looked at *COCH*, the top predicted gene for the first audio profile, as shown in Figure 44. Upon visually examining the audio profile, we find the moderate down-sloping shape of *COCH* followed the patient's down-sloping audiogram, similarly giving *COCH* a solid initial case for being the correct gene.

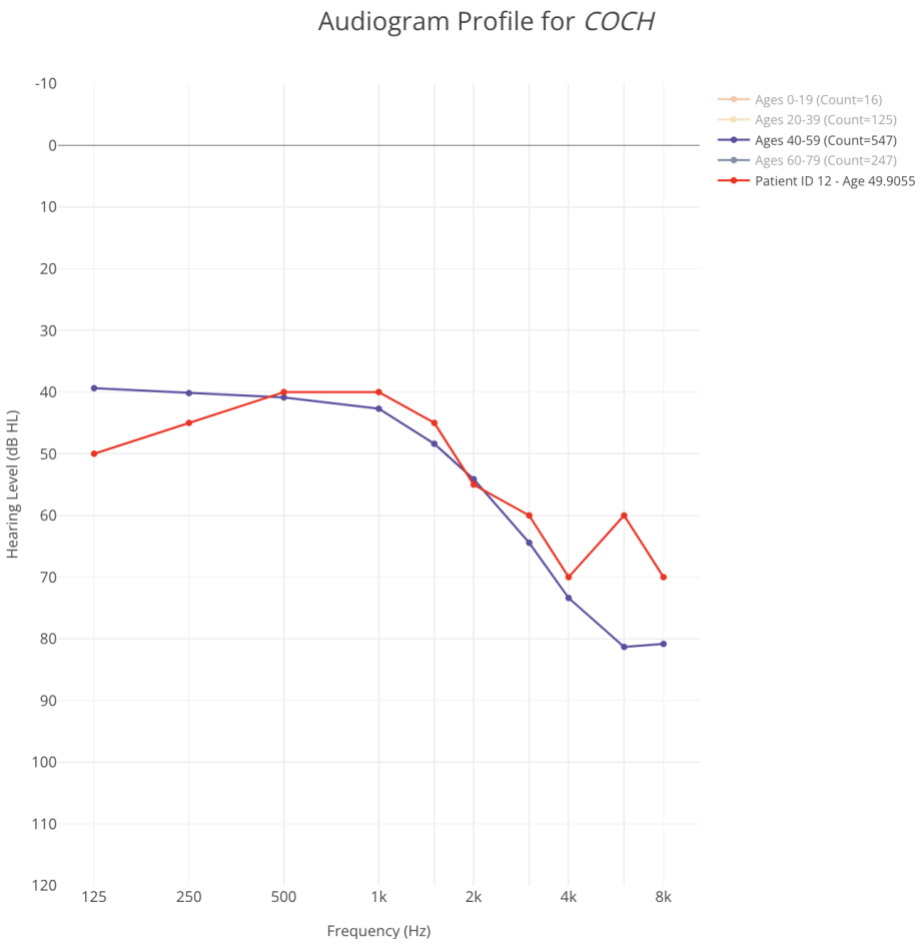

Figure 44. Audio Profile of *COCH* with the patient's (ID 12) audiogram in red, taken at 49 years of age.

Next, we visually examined the audio profile of the second predicted gene, *EYA4*, shown in Figure 45. In this audio profile, we see that the shape of *EYA4* is strictly down-sloping, whereas the patient's audiogram upslopes slightly for the first three frequencies and then drastically downslopes. While the range of dBLoss between the patient's audiogram and the audio profile of *EYA4* for the patient's age range is similar, it does not exhibit as close of similarities as *COCH*, suggesting *COCH* is still the most promising gene.

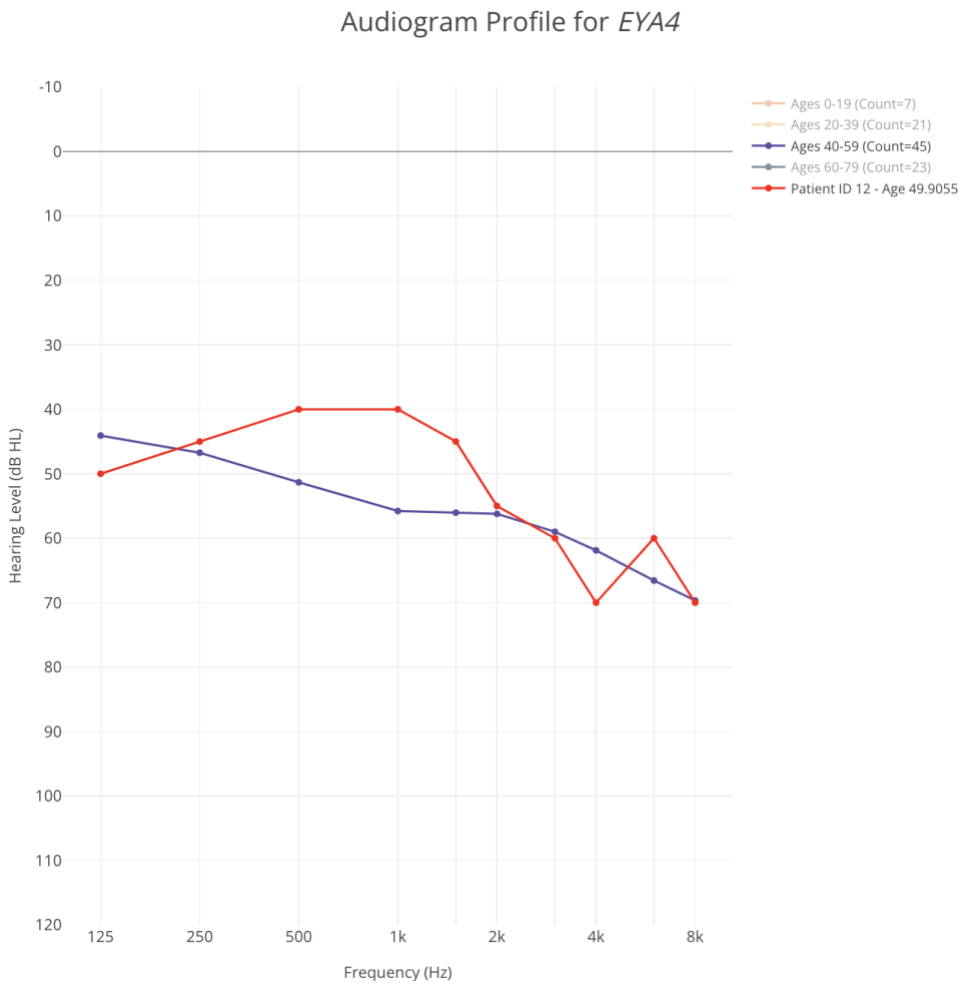

Figure 45. Audio Profile of *EYA4* with the patient's (ID 12) audiogram in red, taken at 49 years of age.

Finally, for the audio profile examination, we looked at the audio profile of the last of the top three predicted genes, *MYO7A*, as shown in Figure 46. Looking at the audio profile, we can see that the shape of *MYO7A* upslopes moderately up until 2000 Hz and then downslopes, similar to the patient's audiogram. However, there are still marginal discrepancies in dBLoss, especially compared to *COCH*, leading us to believe that *COCH* is still the most likely gene.

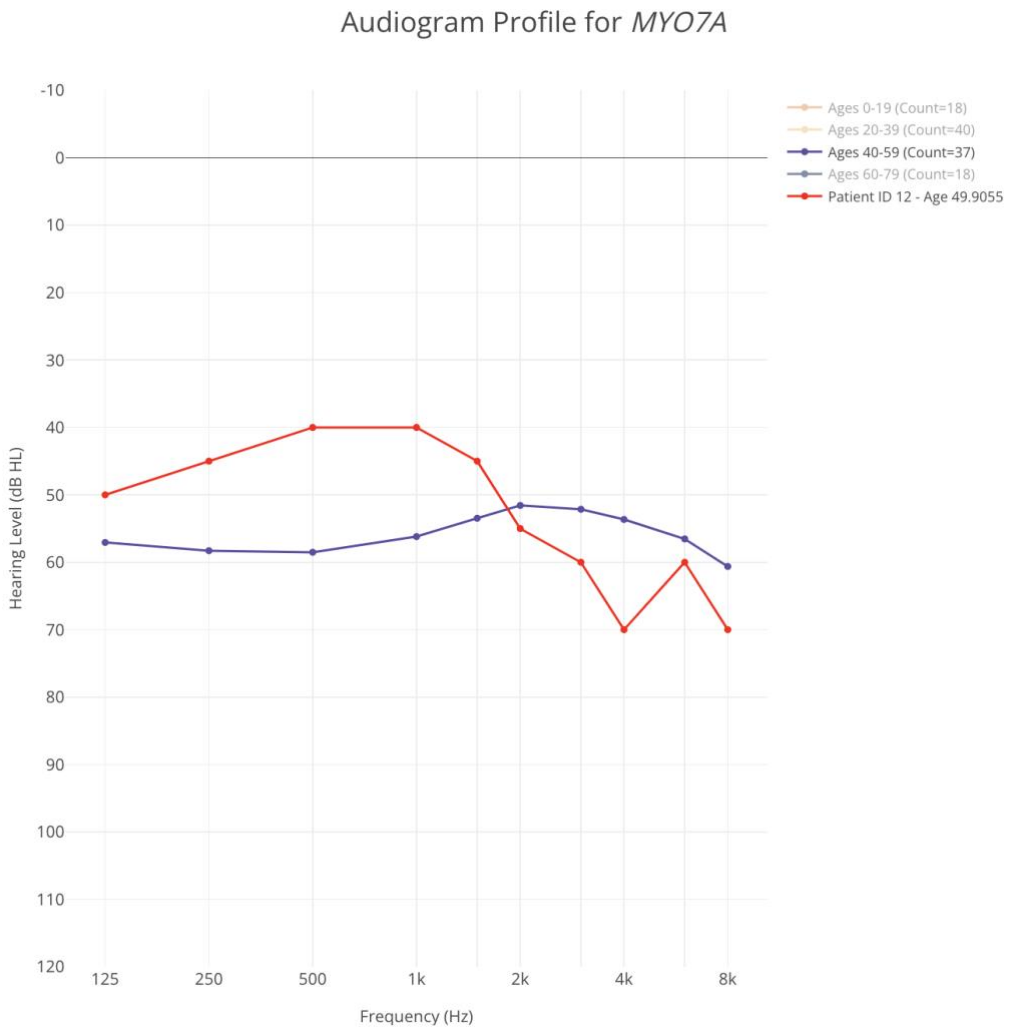

Figure 46. Audio Profile of *MYO7A* with the patient's (ID 12) audiogram in red, taken at 49 years of age.

Continuing our assumption that *COCH* is the correct gene, we further analyzed the clustering shown in Figure 47.

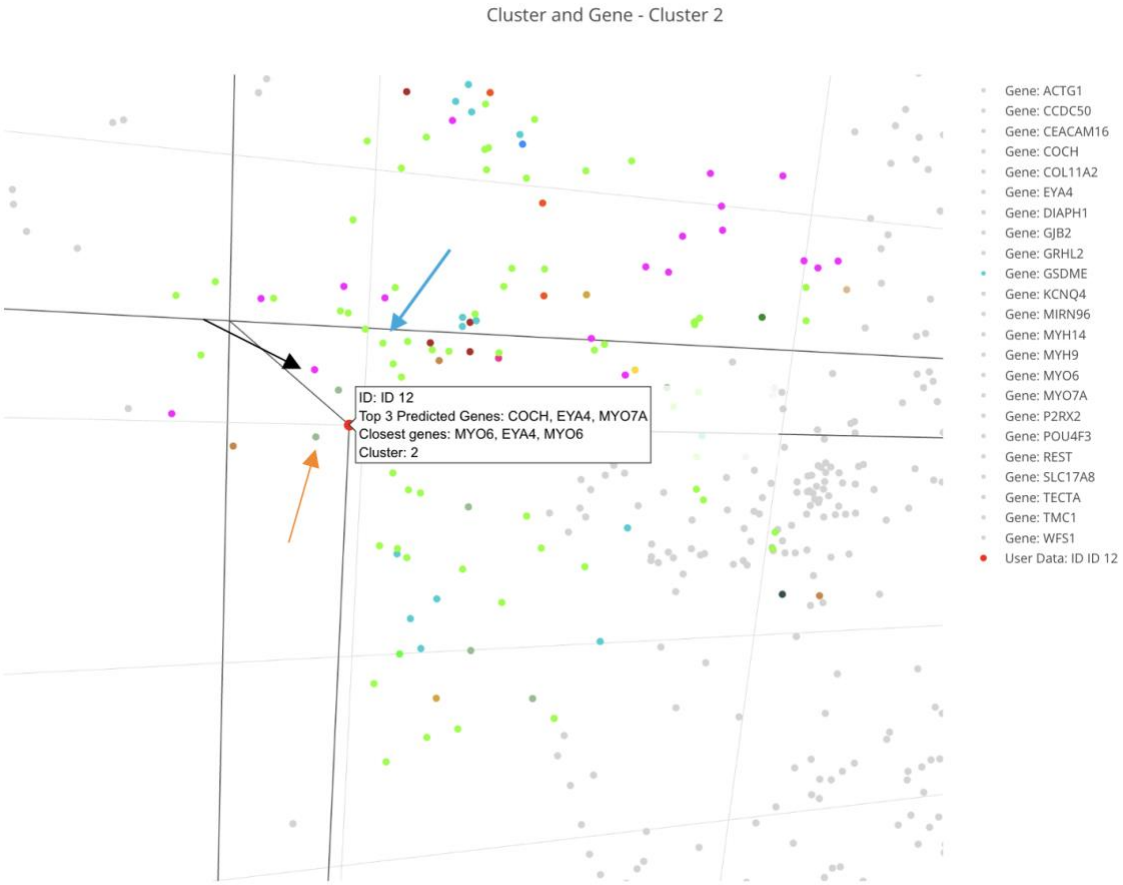

Figure 47. Three-dimensional plot of audiograms in the training set converted into three dimensions, with genes in cluster 2 colored (genes not in the cluster are light grey). Patient (ID 12) is the red dot hovering over the displayed label. The black arrow points to *COCH* (pink), the orange arrow points to *EYA4* (pale green), and the blue arrow points to *KCNQ4* (bright green).

From our examination, we can see that various genes surround the patient's audiogram. We first look at *EYA4* as it is among the top predictions and is visually close to the audiogram and distance-wise, as characterized by the closest neighbors classified by KNN distances. We also saw earlier how our analysis of the audio profile for *EYA4* showed that it had a similar shape compared to the patient's audiogram, therefore putting doubt into our assumption that *COCH* was possibly the correct gene. However, looking at Figure 22, we can see that AudioGene V9.1 has a true positive rate of 97.1% on *COCH*, and the audio profile shape of *COCH* is more closely aligned with the patient's audiogram. Therefore, *EYA4* does not provide enough evidence to

change our original assumption. Looking at the next closest gene via KNN distances, *MYO6*, we find that while it is the fifth predicted gene, its audio profile does not have as similar a correlation to the patient's audiogram compared to *COCH*. Lastly, we looked again visually at the clustering interface and saw that *KCNQ4* (bright green) was all around the patient's audiogram. Looking at Figure 43, we see that it is the sixth prediction. However, upon examination of the audio profile for *KCNQ4*, we saw a steep downslope, which starkly contrasted with the patient's up-sloping, then down-sloping audiogram. Therefore, using AGTD, the analysis suggests that *COCH* was the correct diagnosis for the patient, which is correct. This case study shows that while there can be conflicting findings within the clustering results, narrowing down more prominent genes becomes more accessible due to the accuracy that AudioGene V9.1 gives with the large classes.

4.2.6. Case Study 6 – DIAPH1 – Patient ID 9

For this last case study, we analyzed the results of the Patient (ID 9), who was diagnosed with a *DIAPH1* genetic mutation. The patient only had one audiogram collected. Therefore, AudioGene V9.1 was used for the predictions, shown in Figure 48.

Figure 48. Gene Predictions were made using AudioGene V9.1 for the patient (ID 9), with the top three genes being *KCNQ4*, *TMC1*, and *TECTA*.

In examining the top three genes predicted by AudioGene V9.l, we first analyzed the audio profiles of each of the top three genes: *KCNQ4*, *TMC1*, and *TECTA*. For the first audio profile, we looked at *KCNQ4*, which can be seen in Figure 49. Looking at the audio profile, we can see that the patient's audiogram follows a moderately similar line to *KCNQ4*. This suggested that *KCNQ4* had a strong supportive case for being the correct gene.

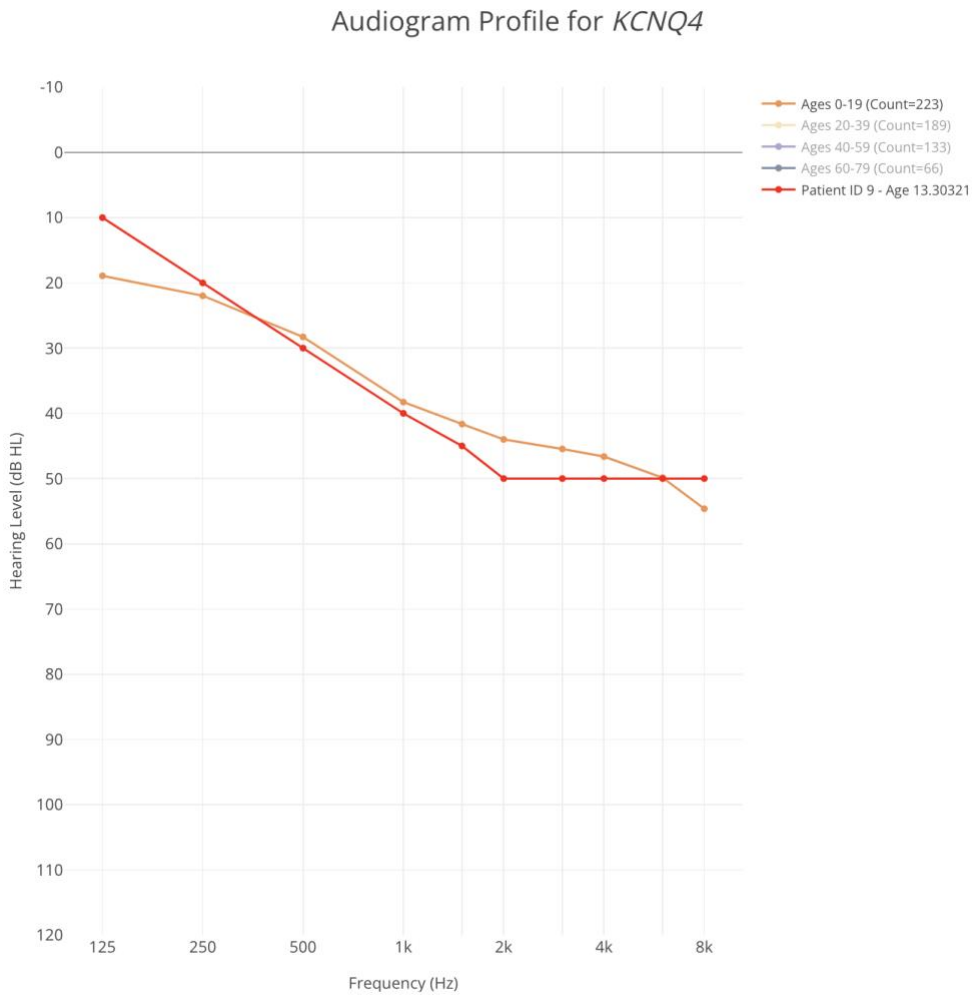

Figure 49. Audio Profile of *KCNQ4* with the patient's (ID 9) audiogram in red, taken at 13 years of age.

Next, we examined the audio profile for the second predicted gene, *TMC1*, shown in Figure 50. Looking at the audio profile, we can see that the sharp down-sloping shape of *TMC1* followed the patient's down-sloping audiogram until 2000 Hz when the patient's audiogram flattened out. This audio profile suggests that while *TMC1* has a case for being the correct gene, the similarities in *KCNQ4* compared to the patient are too great to suggest that it is *TMC1* instead of *KCNQ4*.

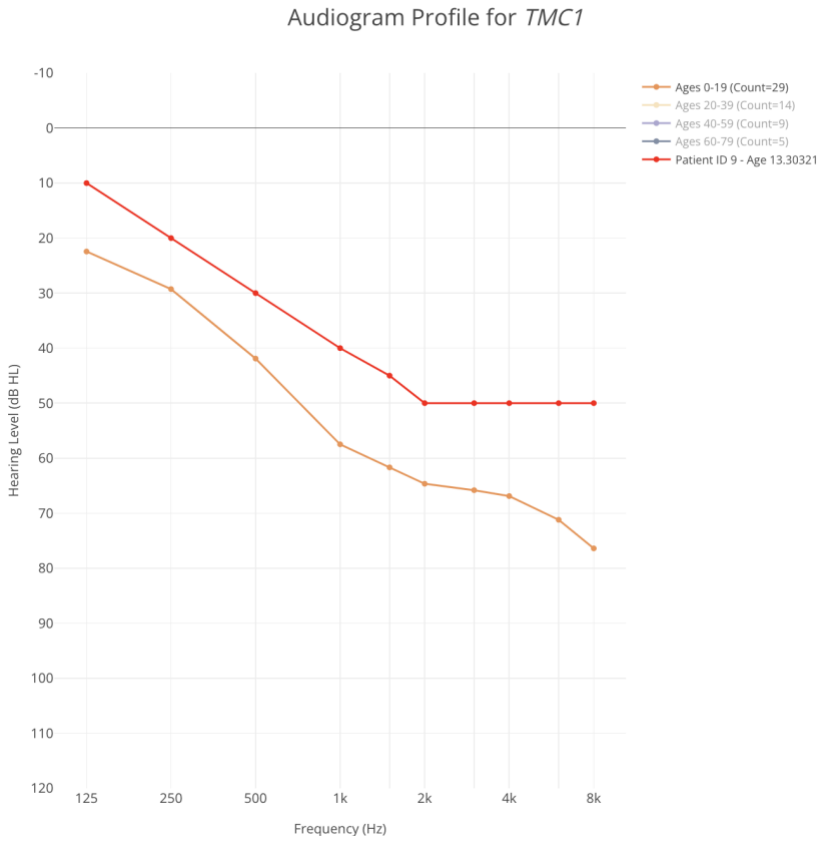

Figure 50. Audio Profile of *TMC1* with the patient's (ID 9) audiogram in red, taken at 13 years of age.

Finally, for the audio profile examination, we looked at the audio profile of the last of the top three predicted genes, *TECTA*, as shown in Figure 51. Visually, we see that the audio profile of *TECTA* presents a slightly similar shape to the patient's audiogram. However, it does not come close to the similarities displayed in Figures 49 and 50 for *KCNQ4* and *TMC1*, respectively. Therefore, we determined that, at this point, *TMC1* is the least likely gene out of the top three.

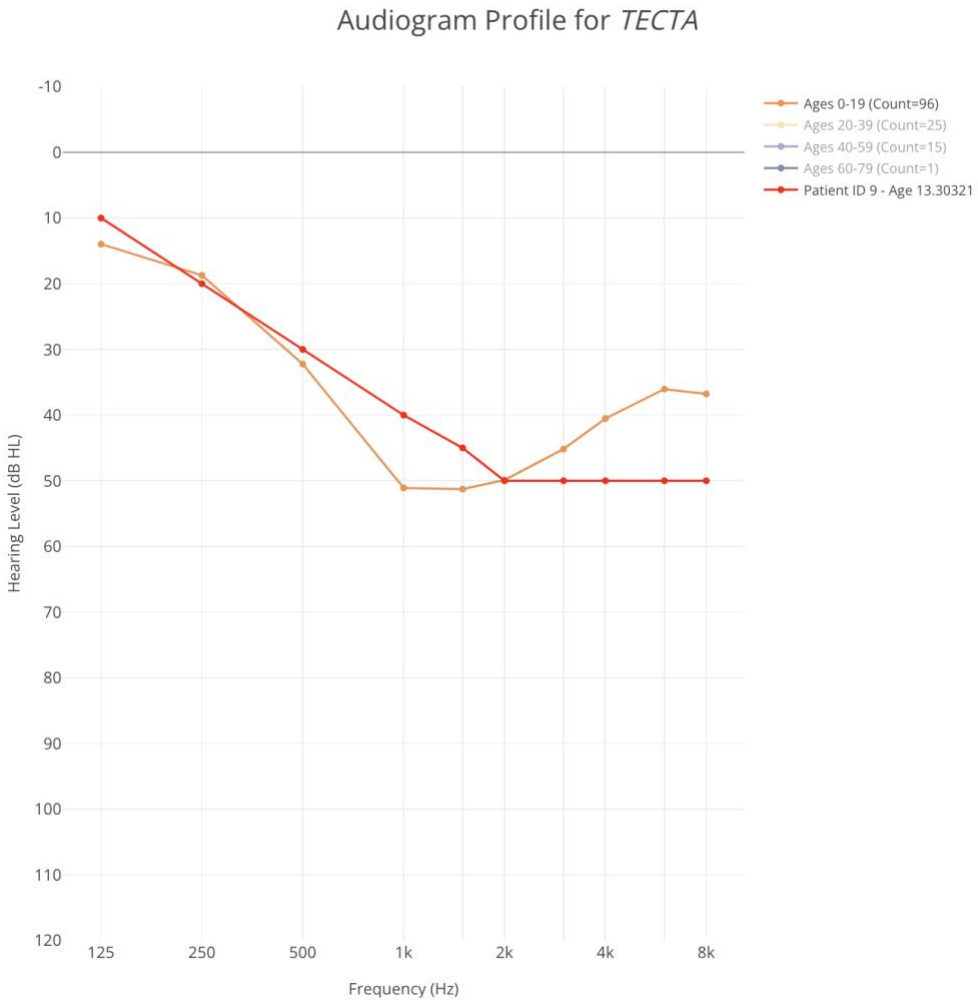

Figure 51. Audio Profile of *TECTA* with the patient's (ID 9) audiogram in red, taken at 13 years of age.

Continuing our assumption that *KCNQ4* was the correct gene, we further analyzed the clustering shown in Figure 52.

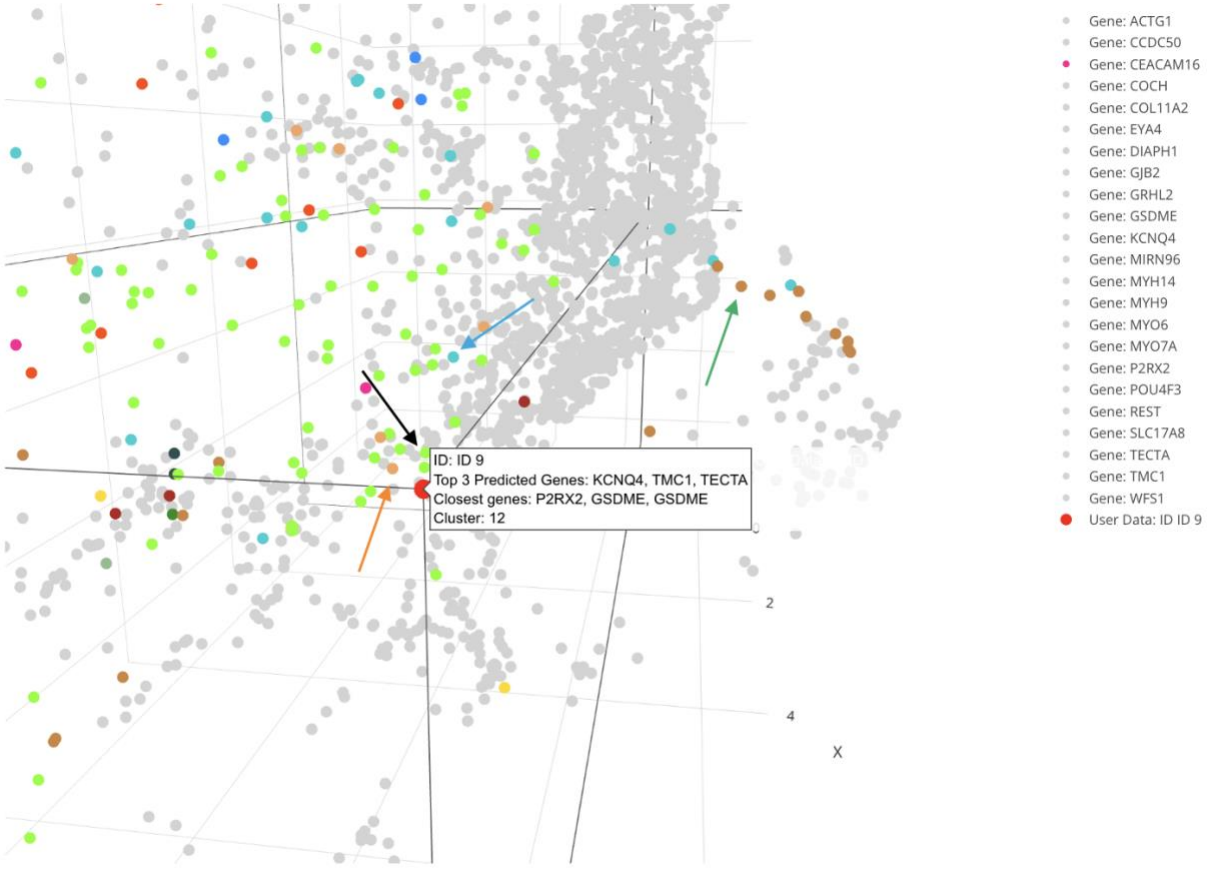

Figure 52. Three-dimensional plot of audiograms in the training set converted into three dimensions, with genes in cluster 12 colored (genes not in the cluster are light grey). Patient (ID 9) is the red dot hovering over the displayed label. The black arrow points to *KCNQ4* (bright green), the orange arrow points to *TMC1* (light brown), the green arrow points to *TECTA* (brown), and the blue arrow points to *GSDME* (light blue).

Upon examining the clustering tool, we see that the closest genes from the distance calculation using KNN are *GSDME* and *P2RX2*. Looking at *GSDME* in the prediction order, found in Figure 48, we can see that it is the fourth predicted gene, giving reason to investigate further through the audio profile. Upon looking at the audio profile, we saw that the patient's audiogram closely followed the *GSDME* shape, suggesting that *KCNQ4* may not be the correct gene. However, upon further examination of the clustering, we can see that there are far more points of *KCNQ4* (bright green) than *GDSME* points (light blue). Looking at the other closest gene via the KNN distances, *P2RX2*, we see that it is not within even the top eight predictions, as

shown in Figure 48. Therefore, this led us to believe this gene was unlikely to be correct. Next, we looked visually around the cluster and saw that *TMC1* was close, visually, as well; however, as stated in our audio profile examination of *TMC1* above, we determined that it was not as close as *KCNQ4*.

Along with this, the number of *KCN4* points completely overwhelms the points of *TMC1*. Given this information, AGTD suggests that the correct diagnosis would be *KCNQ4*, even though the correct diagnosis was *DIAPH1*. This incorrect diagnosis could stem from various reasons, including the dissimilarity of the audio profile of the correct gene, as shown in Figure 53, and the gene being the 20th predicted gene by AudioGene V9.1. However, we believe this is due to this patient's audiogram being predicted by AudioGene V9.1, where there is only a 30% top-one accuracy (shown in Figure 22), along with the training data containing 10 instances for this gene. Therefore, this leads us to conclude that this case was unlikely to perform well in AGTD, displaying the flaws and shortcomings of AGTD and showcasing that while the tool may be powerful, it is imperfect.

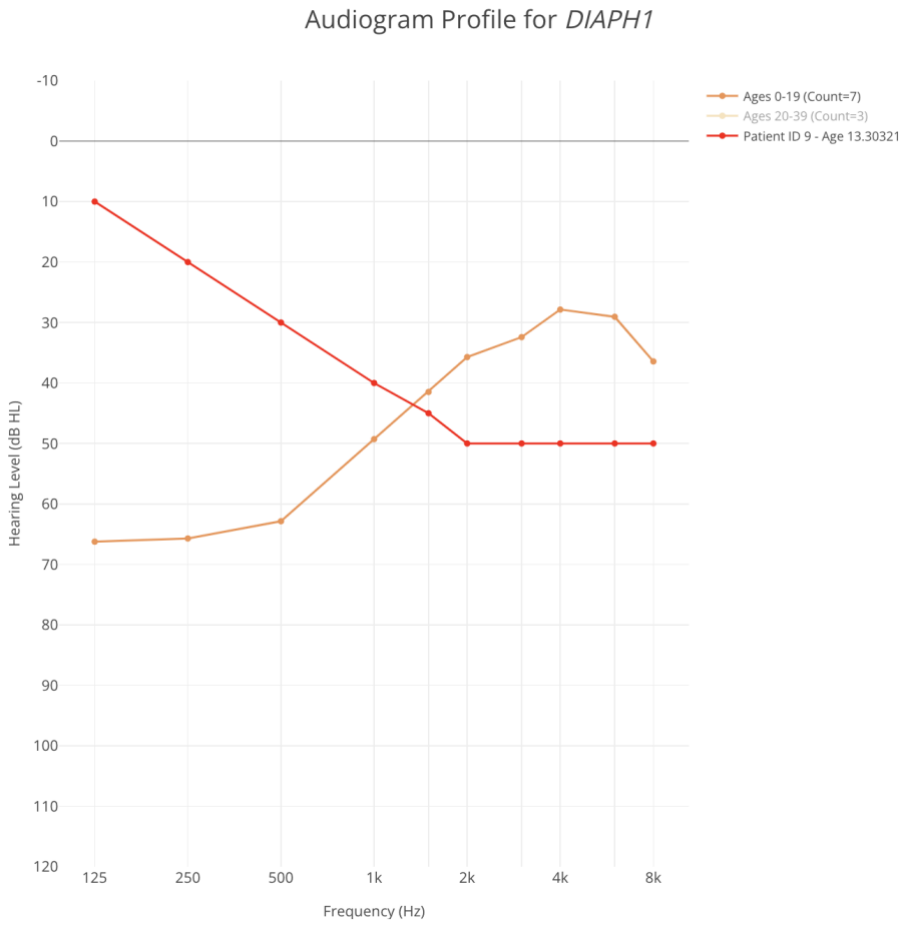

Figure 53. Audio Profile of *DIAPH1* with the patient's (ID 9) audiogram in red, taken at 13 years of age.

4.3 Discussion of Findings

In our search for illuminating case studies, we found a variety of applications for AGTD, and these cases suggest that using the dashboard can help clinicians improve the fidelity of diagnosis for patients – whether through increasing or decreasing confidence in a set of predictions. Through the case studies, AGTD demonstrated its utility in bolstering confidence in accurate diagnoses while appropriately tempering assurance in potentially incorrect predictions. In essence, AGTD was capable at distinguishing the true positives from false positives, improving diagnostic precision. However, it also revealed instances where AGTD could inadvertently diminish confidence in a correct diagnosis, particularly in genes less represented

within the training dataset. Despite this, a key strength of AGTD lies in its capacity to discern—with a significant degree of certainty—whether a diagnosis falls within the 73% average accuracy our models nominally achieve. It also appears to be valuable in affirming correct diagnoses and, crucially, signaling when a prediction might belong to the 27% error margin, guiding users towards or away from specific diagnoses with a more informed perspective. In conclusion, AGTD appears to be a promising tool with various use cases that build a foundation for further research and diagnostic methods.

CHAPTER 5: CONCLUSION AND FUTURE WORK

In conclusion, this thesis describes the use of the AGDT dashboard, which aims to solve the challenge of improving both the accuracy and confidence of ADNSHL diagnoses through the hybrid model of machine learning models combined with advanced visualizations. Through six case studies analyzed, we illustrated the pros and cons of using AGTD. However, we concluded that AGTD is a powerful tool that can improve the accuracy of diagnostics and enhance understanding of the predictions of the AudioGene models. Thus, we suggest that AGDT is a leap forward in the AudioGene platform's predictive capabilities, forming a firm foundation on which experts can form and validate hypothesized diagnoses of ADNSHL cases—implying that AGTD will be a valuable tool moving forward in the field of targeted treatments for patients experiencing hearing loss.

5.1. Recommendations for Future Research

For future work with AGTD, a couple of improvements can be made to the tool to enhance the overall user experience and guide the user in making better diagnoses. The first change would be how we interpret multiple audiograms for a patient in the three-dimensional cluster plot. Suggestions for this would be looking into the method employed by AG4 for bagging audiograms and finding the best representative or displaying to the user multiple options to show the patient, such as: “Show All Audiograms,” “Show Oldest Audiogram,” “Show Youngest Audiogram,” and so one. The following change to be made would be to add in a resampling technique method, where there are three options provided to the user: Random Resampling for Up-Sampling and Down-Sampling, SMOTE for Up-Sampling, and NearMiss for Down-Sampling. The user could either select Random Sampling and input the total number of samples they desired or a combination of SMOTE and NearMiss and input the minimum and

maximum samples they desired. The last change would be to normalize the display points in the 3D plot so that there could be a correct inference drawn from the surrounding points near the patient, eliminating bias based on counts alone, as bias is possible due to more data points for some genes than others.

5.2. Summary

AGDT is not a perfect diagnostic tool. However, through case studies, experimentation, and clinician suggestions for improvements, we discovered a method that allows clinicians to understand the AudioGene models better, helping to further the advancements of ADSNHL targeted treatment. As of the publication of this thesis, the platform is available on the AudioGene website (audiogene.eng.uiowa.edu) and is open to public use.

REFERENCES

1. Ryan, S.R. (2023). Machine Learning Prediction of Genetic Hearing Loss via Selective Intra-ensemble Data Partitioning. (Unpublished master's thesis). University of Iowa, Iowa City, IA.
2. Nwakama, C. (2021). AudioGene 9.0: novel ensemble machine learning classification of 23 classes of autosomal non-syndromic hearing loss (deafness). (Unpublished master's thesis). University of Iowa, Iowa City, IA.
3. Taylor, K.R., Smith, R.J.H., & Hood, L.J. (2013). AudioGene: predicting hearing loss genotypes from phenotypes to guide genetic screening. *Hum Mutat*, **34**(4), 539-545.
4. Taylor, K.R., Smith, R.J.H., & Hood, L.J. (2016). Audioprofile Surfaces: The 21st Century Audiogram. *Annals of Otology, Rhinology & Laryngology*, **125**(5), 361-368.
5. Weininger, O., Smith, R.J.H., & Hood, L.J. (2019). Computational analysis based on audioprofiles: A new possibility for patient stratification in office-based otology. *Audiology Research*, **9**(2), 230.
6. Enterprise DNA Experts. (n.d.). What is SQL used for? 7 top uses. https://blog.enterprisedna.co/what-is-sql-used-for
7. Oracle. (n.d.). What is a relational database? https://www.oracle.com/database/what-is-a-relational-database/
8. Express. (n.d.). Node.js web application framework. http://expressjs.com/
9. JWT. (n.d.). Introduction to JSON web tokens. https://jwt.io/introduction
10. Node.Js®. (n.d.). About Node.Js. http://nodejs.org/en/about
11. Pandas. (n.d.). API reference - Pandas 1.5.2 documentation. https://pandas.pydata.org/pandas-docs/version/1.5/reference/index.html
12. Plotly Graphing Libraries. (n.d.). Plotly javascript graphing library in JavaScript. http://plotly.com/javascript/
13. Hildebrand, M. S., et al. (2008). Audioprofile-directed screening identifies novel mutations in KCNQ4 causing hearing loss at the DFNA2 locus. *Genet Med*, **10**(11), 797-804.
14. GeeksforGeeks. (n.d.). Flask tutorial. http://www.geeksforgeeks.org/flask-tutorial/
15. React. (n.d.). React – A JavaScript library for building user interfaces. http://legacy.reactjs.org/
16. MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, **1**(14), 281-297.
17. Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, **31**(8), 651-666.
18. Hamerly, Greg; Elkan, Charles (2002). Alternatives to the k-means algorithm that find better clusterings. *Proceedings of the eleventh international conference on information and knowledge management (CIKM)*.

19. Jolliffe, I. T. (2002). Principal Component Analysis. Springer-Verlag.
20. Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, **13**(1), 21-27.
21. Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, **46**(3), 175-185.
22. Deza, E., & Deza, M.M. (2009). Encyclopedia of Distances. Berlin: Springer.
23. Merkel, D. (2014). Docker: Lightweight Linux Containers for Consistent Development and Deployment. *Linux Journal*. https://www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment
24. Bader, K., et al. (2021). Test-retest reliability of distortion-product thresholds compared to behavioral auditory thresholds. *Hear Res*, **406**, 108232.
25. Schmuziger, N., Probst, R., & Smurzynski, J. (2004). Test-retest reliability of pure-tone thresholds from 0.5 to 16 kHz using Sennheiser HDA 200 and Etymotic Research ER-2 earphones. *Ear Hear*, **25**(2), 127-32.
26. Bessen, S. Y., et al. (2023). Test-retest repeatability of automated threshold audiometry in Nicaraguan schoolchildren. *International Journal of Audiology*, **62**(3), 209-216.
27. Smith, R. J., Bale, J. F., Jr., & White, K. R. (2005). Sensorineural hearing loss in children. *Lancet*, **365**(9462), 879-90.
28. Docker Documentation. (n.d.). Docker compose overview. http://docs.docker.com/compose/
29. NGINX. (n.d.). What is a reverse proxy vs. load balancer? https://www.nginx.com/resources/glossary/reverse-proxy-vs-load-balancer/
30. Vijay, H. (n.d.). Dimensionality reduction: PCA, Tsne, Umap. https://aurigait.com/blog/blog-easy-explanation-of-dimensionality-reduction-and-techniques/
31. Coenen, A., & Pearce, A. (n.d.). Understanding Umap. https://pair-code.github.io/understanding-umap/
32. McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv:1802.03426 [stat.ML].
33. Albarrak, A. M. (2023). Improving the trustworthiness of interactive visualization tools for healthcare data through a medical fuzzy expert system. *Diagnostics*, **13**(10):1733. https://doi.org/10.3390/diagnostics13101733
34. Krzywinski, M., et al. (2009). Circos: An information aesthetic for comparative genomics. *Genome Res*, **19**:1639-1645. https://genome.cshlp.org/content/early/2009/06/15/gr.092759.109.abstract
35. Xu, W., Zhong, Q., Lin, D., et al. (2021). CoolBox: A flexible toolkit for visual analysis of genomics data. *BMC Bioinformatics*, **22**, 489. https://doi.org/10.1186/s12859-021-04408-w

36. Moreno-Cabrera, J. M., del Valle, J., Castellanos, E., et al. (2020). Evaluation of CNV detection tools for NGS panel data in genetic diagnostics. *Eur J Hum Genet*, **28**, 1645–1655. https://doi.org/10.1038/s41431-020-0675-z
37. Katsanis, S., & Katsanis, N. (2013). Molecular genetic testing and the future of clinical genomics. *Nat Rev Genet*, **14**, 415–426. https://doi.org/10.1038/nrg3493
38. Thorvaldsdóttir, H., Robinson, J. T., & Mesirov, J. P. (2013). Integrative genomics viewer (IGV): High-performance genomics data visualization and exploration. *Briefings in Bioinformatics*, **14**, 178-192. https://academic.oup.com/bib/article/14/2/178/208453
39. Venkatesh, M. D., Moorchung, N., & Puri, B. (2015). Genetics of non syndromic hearing loss. Medical Journal, *Armed Forces India*, **71**(4), 363–368. https://doi.org/10.1016/j.mjafi.2015.07.003
40. Wu, T.-F., Lin, C.-J., & Weng, R. C. (2003). Probability estimates for multi-class classification by pairwise coupling. In *Advances in Neural Information Processing Systems 14* (NIPS 2003). http://www.csie.ntu.edu.tw/~cjlin/papers/svmprob/svmprob.pdf