Fall 2016

Variant-curation and database instantiation (Variant-CADI): an integrated software system for the automation of collection, annotation and management of variations in clinical genetic testing

Andrea Rae Hallier
University of Iowa

VARIANT-CURATION AND DATABASE INSTANTIATION (VARIANT-CADI): AN INTEGRATED SOFTWARE SYSTEM FOR THE AUTOMATION OF COLLECTION, ANNOTATION AND MANAGEMENT OF VARIATIONS IN CLINICAL GENETIC TESTING

by

Andrea Rae Hallier

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

December 2016

Thesis Supervisor: Associate Professor Terry A. Braun

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

_____

MASTER'S THESIS

_____

This is to certify that the Master's thesis of

Andrea Rae Hallier

has been approved by the Examining Committee for
the thesis requirement for the Master of Science degree
in Biomedical Engineering at the December 2016 graduation.

Thesis Committee: _____
Terry A. Braun, Thesis Supervisor

_____
Thomas L. Casavant

_____
Michael J. Schnieders

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Dr. Terry Braun, for his support and expertise guiding me through not only this thesis, but also my undergraduate and graduate career. It would not have been the same without his assistance. The time he devoted did not go unnoticed and I am deeply grateful for Dr. Braun's time.

To my mentor, Sean Ephraim, his kind and encouraging leadership through this process has left a lasting impression. I would like to thank him for sharing his experiences and wisdom with me. It would have been a greater struggle without Sean.

ABSTRACT

One of the tools a clinician has in disease diagnosis and treatment is genetic testing. To generate value in genetic testing, the link between genetic variants and disease must be discovered, documented, and shared within the community. Working with two existing genomic variation tools, Kafeen and Cordova, a new set of features referred to as Variant-Curation and Database Instantiation (Variant-CADI) was identified, designed, implemented and integrated into the existing Cordova system to unite data collection, management and distribution into one cohesive tool accessible through user interfaces. This eliminates the user needing specialized knowledge of the underlying implementation, data pipeline or data management to collect desired disease specific genetic variations. Using this tool, new disease-specific variation database instances have been initialized and created as demonstrations of the utility of these applications.

PUBLIC ABSTRACT

One of the tools a clinician has in disease diagnosis and treatment is the evaluation of DNA sequences for changes that may cause disease. Changes in DNA sequence can be valuable for the diagnosis of disease, consideration of treatment, and understanding the pathophysiology of disease. Tools to share the link between disease phenotypes and observed DNA changes can be valuable to the research and clinical communities. This thesis describes extensions to software tools (Cordova and Kafeen) that are used to aggregate data for the evaluation of pathogenicity of DNA sequence variants. Kafeen is a data collection pipeline that searches public repositories for both pathogenic and non-pathogenic variations. It assembles this data and calculates an overall pathogenicity prediction for each variation found. Cordova is a web template suited to management of genetic variation data. Cordova is available to the research community for the deployment of disease-specific genetic variation databases. New functionality was identified, designed, implemented and integrated along with Kafeen into the Cordova system to automate collection, management and distribution of genetic variation data within the Cordova software system. Using this tool, new disease-specific variation database instances have been initialized and created as demonstrations of the utility of these applications.

TABLE OF CONTENTS

LIST OF TABLES vi

LIST OF FIGURES vii

CHAPTER 1 - INTRODUCTION 1

CHAPTER 2 - BACKGROUND 6

2.1 Cordova 6

2.2 Kafeen 27

CHAPTER 3 – METHODS 29

CHAPTER 4 - VARIANT-CADI 30

4.1 Gene Upload 33

4.2 Variant Collection 35

4.3 Normalize Nomenclature 37

4.4 Expert Curation 40

4.5 Review and Release Changes 43

4.6 Database Updates 46

CHAPTER 5 - OUTCOMES 47

CHAPTER 6 – FUTURE ENHANCEMENTS 53

REFERENCES 55

LIST OF TABLES

Table 1. System Requirements 32

Table 2. Head and Neck Cancer Genes 50

Table 3. Renal Genes 50

Table 4. Cleft Genes 50

Table 5. Vision Genes 51

LIST OF FIGURES

Figure 1. Data Collection Flow Before Variant-CADI 4
Figure 2. Data Collection Flow After Variant-CADI 5
Figure 3. Cordova Interface Schema 7
Figure 4. Public Gene View 8
Figure 5. Public Gene View Expanded 9
Figure 6. Public Variant View 10
Figure 7. Public Variant View Pop-up 11
Figure 8. Secure Login Page 13
Figure 9. Admin Create User Interface 14
Figure 10. Admin View Users Interface 15
Figure 11. User Profile Interface 16
Figure 12. Admin View Groups Interface 17
Figure 13. Admin Create Group Interface 17
Figure 14. Edit Variant Find Variant By Gene 18
Figure 15. Edit Variant Find Variant Interface 19
Figure 16. Edit Variant Interface 20
Figure 17. Add Variant Interface 21
Figure 18. Release Changes Interface 22
Figure 19. Review Changes Variant Description Table 23
Figure 20. Release Changes Edit Unreleased Variant 24
Figure 21. Activity Logs 25
Figure 22. Cordova Database Schema 26
Figure 23. Kafeen Pipeline 28
Figure 24. Variant-CADI Overview 31
Figure 25. Cordova System with Variant-CADI Functionality 32
Figure 26. Upload Genes Data Flow 33
Figure 27. Upload Genes Interface 34

Figure 28. Variant Collection Data Flow 35
Figure 29. Setup Database Data Flow 36
Figure 30. Variant Collection 36
Figure 31. Normalize Nomenclature Data Flow 38
Figure 32. Normalize Nomenclature Interface 39
Figure 33. Expert Curation Data Flow 41
Figure 34. Expert Curation Interface 42
Figure 35. Release Changes Data Flow 44
Figure 36. Release Changes Interface After Variant-CADI 45

CHAPTER 1 - INTRODUCTION

One of the tools a clinician has in disease diagnosis and treatment is genetic testing. The value to patients in genetic testing comes from the ability to i) better diagnose disease, ii) develop new therapies and drug targets, and iii) to better understand the pathophysiology of disease. There is additional value to both research and patients when phenotypes and variants can be shared with the research and clinical communities. Developing accessible software is vital to sharing this disease specific genetic information. Many of the challenges in this field include collecting, managing, organizing and distributing relevant disease-specific genetic variation data. The software described here incorporates and expands upon existing genetic variation tools to automate the collection, review and distribution of disease-specific genetic variants as they relate to pathogenicity. This new software and functionality is referred to as Variant-Curation and Database Instantiation (Variant-CADI). Variant-CADI is an expansion to the existing and feature-rich software system, Cordova [1], and a complementary analysis pipeline called Kafeen [2]. Variant-CADI incorporates data collection, annotation, and management that is required to instantiate a Cordova instance. The additional functionality provided by Variant-CADI decreases the data management and installation burden on Cordova users minimizing many of the hurdles posed by data management and sharing in genetic variation research. To demonstrate the utility of this software, several new variation databases have been instantiated using the new Variant-CADI functionality to collect, insert and manage variation data, each site pertaining to a unique disease.

The Cordova system and complementary data pipeline, Kafeen, acquires and integrates genomic data (genes), variant data (variants, allele frequencies, and pathogenicity predictions), and phenotypes (benign, pathogenic) and generates displays for this data [3].

The Kafeen pipeline acquires data from public databases to identify relevant variants (in genes) and their annotation (benign or pathogenic). The results from clinical genetic testing, or research-based genetic testing, of variants in genes may support, or contradict, previous categorization of variations as pathogenic or benign. Local experts then have the ability to curate data to further support or dispute the Kafeen assigned disease status, benign or pathogenic. Ideally, clinicians and investigators would review this data for consensus prior to public release of the data. Given the substantial amount of data (1000s of genes, millions of variants) and the complexity of the Cordova system, deploying and instantiating a new instance of Cordova requires intimate knowledge of the data and the system architecture.

Cordova does not provide methods to collect or curate this data. In addition, Cordova lacks functionality to specify gene lists for batch uploads, interfaces for data review, entry, editing and logging. This thesis describes new interfaces and software system changes to implement these lacking features. Figure 1 and Figure 2 highlight the functionality that has been implemented through Variant-CADI and described in this thesis. These interfaces automate much of the manual and configuration steps currently required to instantiate a Cordova system instance and integrates the Kafeen pipeline with Cordova.

Both Cordova (https://github.com/clcg/cordova) and Kafeen (https://github.com/clcg/kafeen) are open source and are available to the research community [1][2]. Cordova is lacking the necessary tools and interfaces for data management by users that are not intimately familiar with the database and software architecture (i.e., the software authors). The addition of Variant-CADI allows acquisition and manual curation of data not only in initializing a Cordova instance, but to manage and version data throughout the life of the site.

Installing an instance of Cordova on a server requires fewer steps compared to the task of collecting, formatting, curating and loading all of the data used to fill the database tables to describe the variants. To collect this specific set of data requires several steps. The data sets that fill a Cordova instance are substantial (millions of variants, MAFs and pathogenicity scores). The objective of this work is to make it possible for a non-software-author of Cordova and Kafeen to be able to acquire, curate, manage and track changes in data in a Cordova instance through integrated user interfaces.

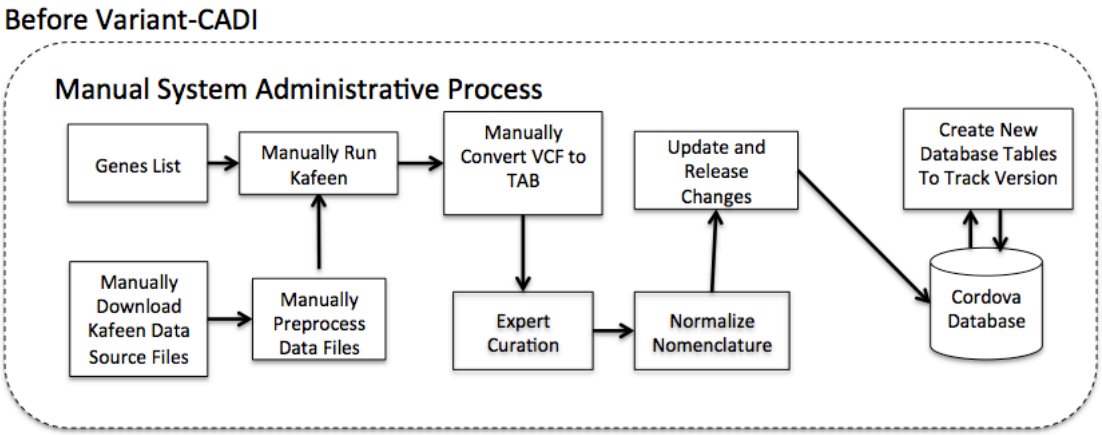


Figure 1. Data Collection Flow Before Variant-CADI

To collect features needed to instantiate a Cordova instance before the introduction of Variant-CADI required intimate knowledge of both Kafeen, Cordova and local scripts to execute the flow. First users needed to acquire data source files for Kafeen and these files needed to be pre-processed. After these files have been collected and processed, the Kafeen configuration file can be edited to a user's preferences. Now Kafeen can be run manually with a gene list input. The output of Kafeen is a VCF file, this must again be managed with another local script to convert the required data to tab format. Now this data is in tab format, the user can add expert curations and normalize the nomenclature manually with additional local scripts. Now that this file is in it's final form, it can be inserted into the Cordova Database, either the queue for staging and review or the public version. Not only does a user need to update the variations table, but a handful of other database tables that require information about variation and gene counts. The version table must also be updated. To use the site's versioning system, the tables must be duplicated, renamed and the data inserted into the new tables.

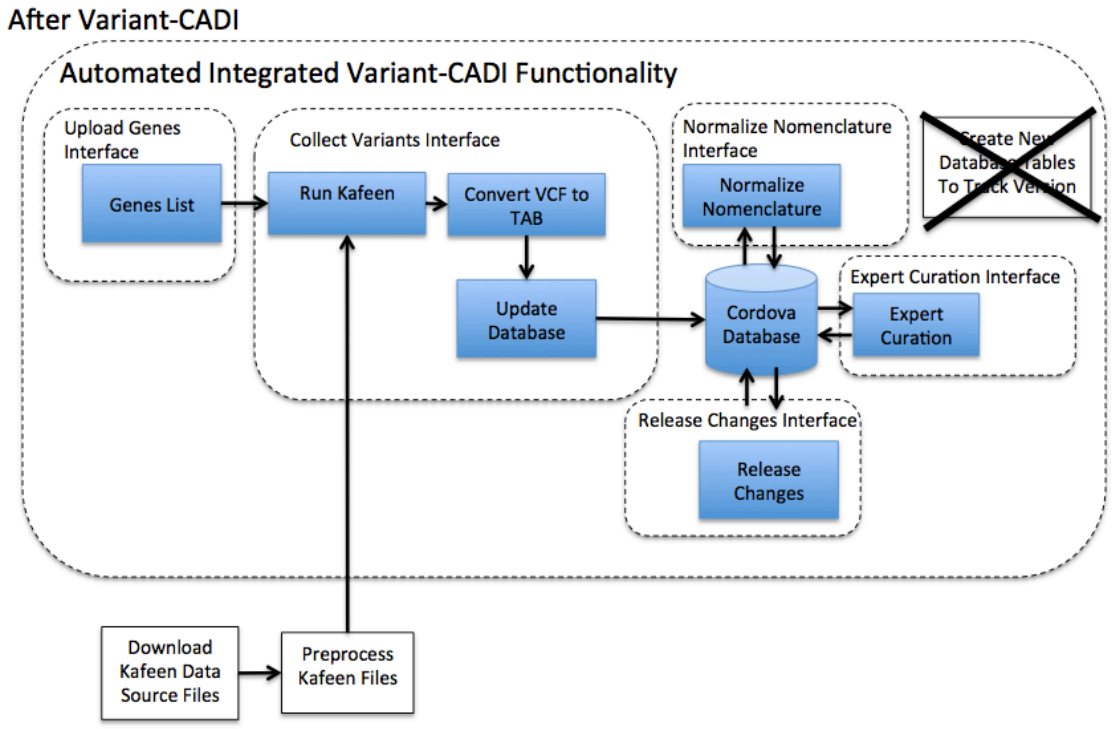


Figure 2. Data Collection Flow After Variant-CADI

After the implementation of Variant-CADI, the data flow looks much different. The responsibility of setting up Kafeen is still on the user, however all other steps have been encapsulated into user interfaces that are incorporated into Cordova. The user first enters a list of genes in the genes interface. Then the user is asked to initiate gene collection. With the press of a button, Kafeen is executed, the output converted and the collected data inserted into the Cordova database queue table for review. Upon data collection completion, the user receives an email with information on the execution of Kafeen. Now that the data is in the queue, the user can use the Expert Curation and Normalize Nomenclature interfaces to apply desired curations to the collected data. Finally, the user can review and release these collected and curated variants in the Release Changes interface. The versioning system was modified and the data flow now resembles a "database update" procedure, where before it was a "database delete and re-create procedure." This eliminates the need to recreate and rename database tables.

The existing software systems, Cordova and Kafeen, are described in Chapter 2. Chapter 3 presents the methods used to implement new Variant-CADI software. Chapter 4 describes the functionality of the implemented Variant-CADI software. The results in Chapter 5 show the use of Variant-CADI to instantiate additional disease-specific variant databases. Finally, Chapter 6 concludes with future enhancements.

CHAPTER 2 - BACKGROUND

2.1 Cordova

Cordova is a software system implemented to provide support for local genetic screening projects [4]. An instance of Cordova is used to deploy an on-line genetic variation database for the management of disease-specific genetic variations. Cordova is designed to represent data for a specific disease domain. Cordova is written in PHP [5]. PHP has native libraries and functionality for interaction with MySQL [6] databases that are used to organize the underlying data. The PHP CodeIgniter framework [7] used to develop Cordova fosters the use of a Model View Controller (MVC) architecture in the software design. This architecture design allows for easy adaption of new features by separation and reusability of functionality in the code base.

Described below in Figure 3 are two main divisions of Cordova, "Public" and "Secure." The “Public” interfaces are available to anyone with or without an account on the Cordova instance. The “Secure” interfaces are available to any Cordova user with an account. These users would typically be the curators of the data provided on the public interfaces.

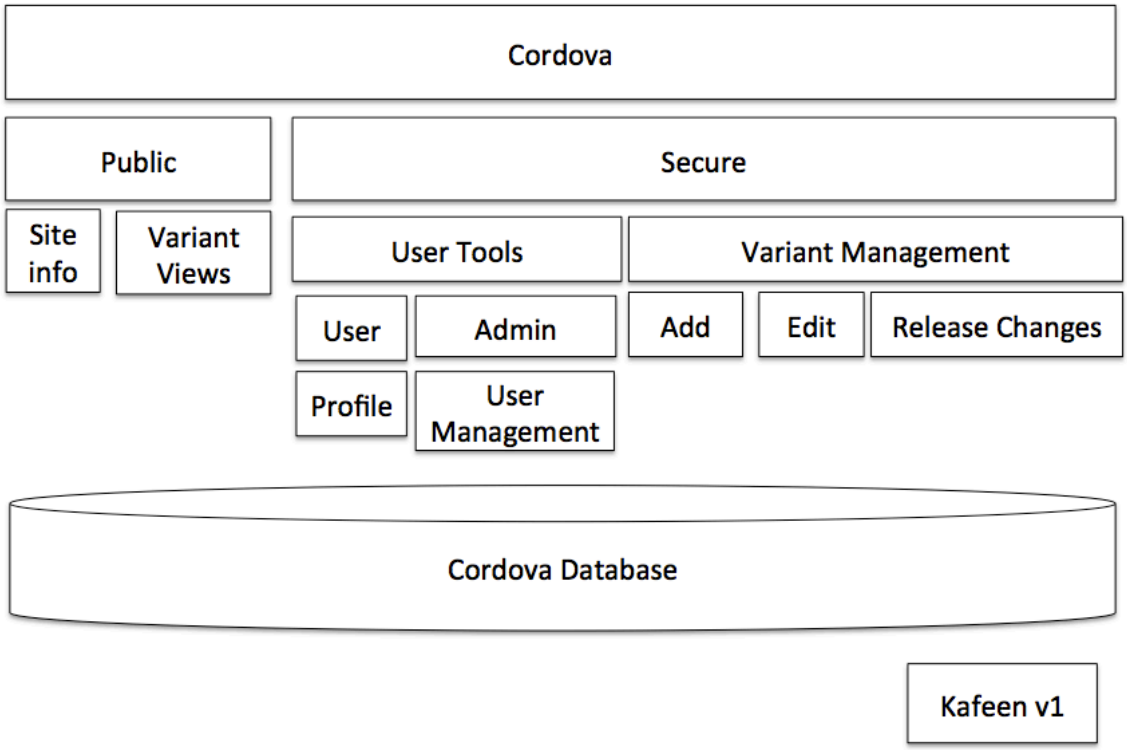


Figure 3. Cordova Interface Schema

The Cordova software application prior to implementation of Variant-CADI consisted of public and secure interfaces. The public interfaces were available to users without login credentials and provided site information and views of the variants. The users with login credentials had access to User Tools including user profile and as a site administrator user management tools. All users had access to variant management tools including adding a single variant (with annotations that relied on a previous version of Kafeen), editing a single variant and releasing staged variants upon approval to the public interfaces.

Public facing interfaces allow browsing of data by gene letter, described in Figure 4, by gene name, described in Figure 5, and by variant, described in Figure 6 and Figure 7.

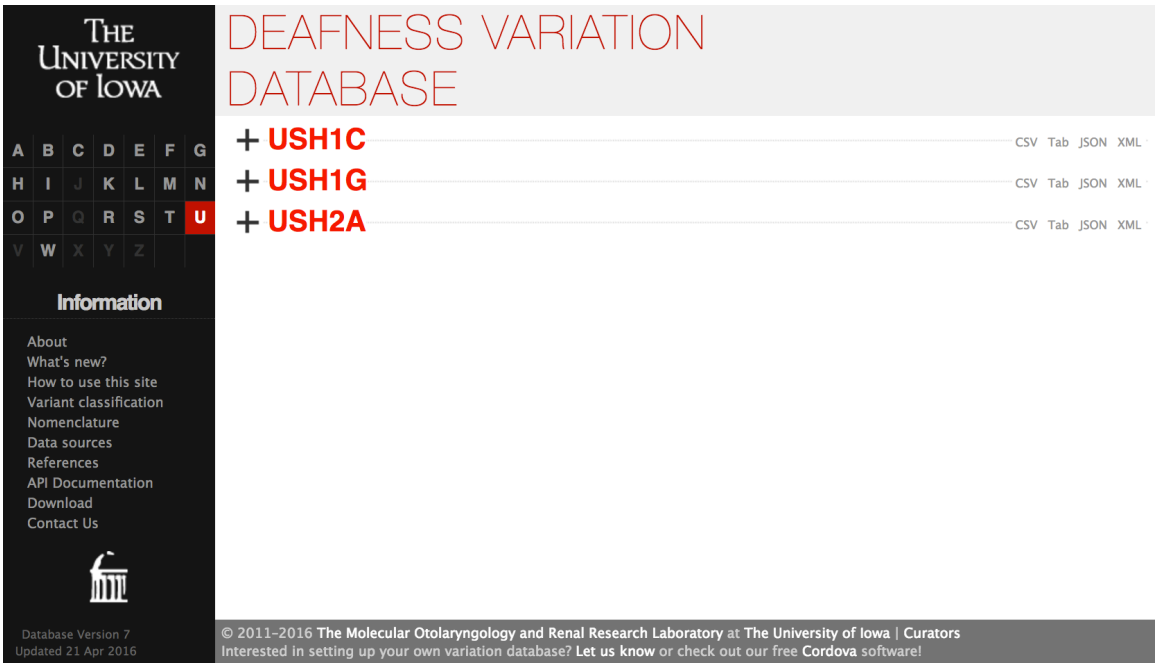


Figure 4. Public Gene View

This figure describes the gene view of the variants from a Cordova instance, the Deafness Variation Database. This interface offers selection of a gene to view grouped by first letter of the gene name. On the right hand side of the page are links to download formatted files of all the variant data for each gene.

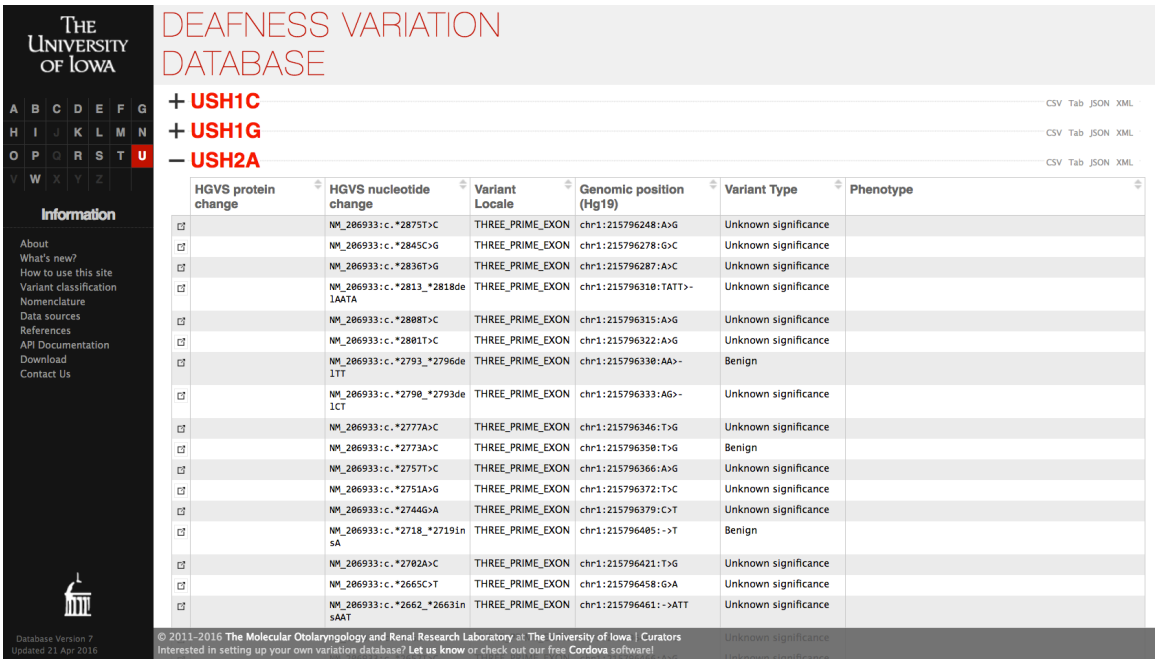


Figure 5. Public Gene View Expanded

With a gene expanded, a list of all of the variants associated with that gene can be viewed in a table along with a few select attributes displayed for each.

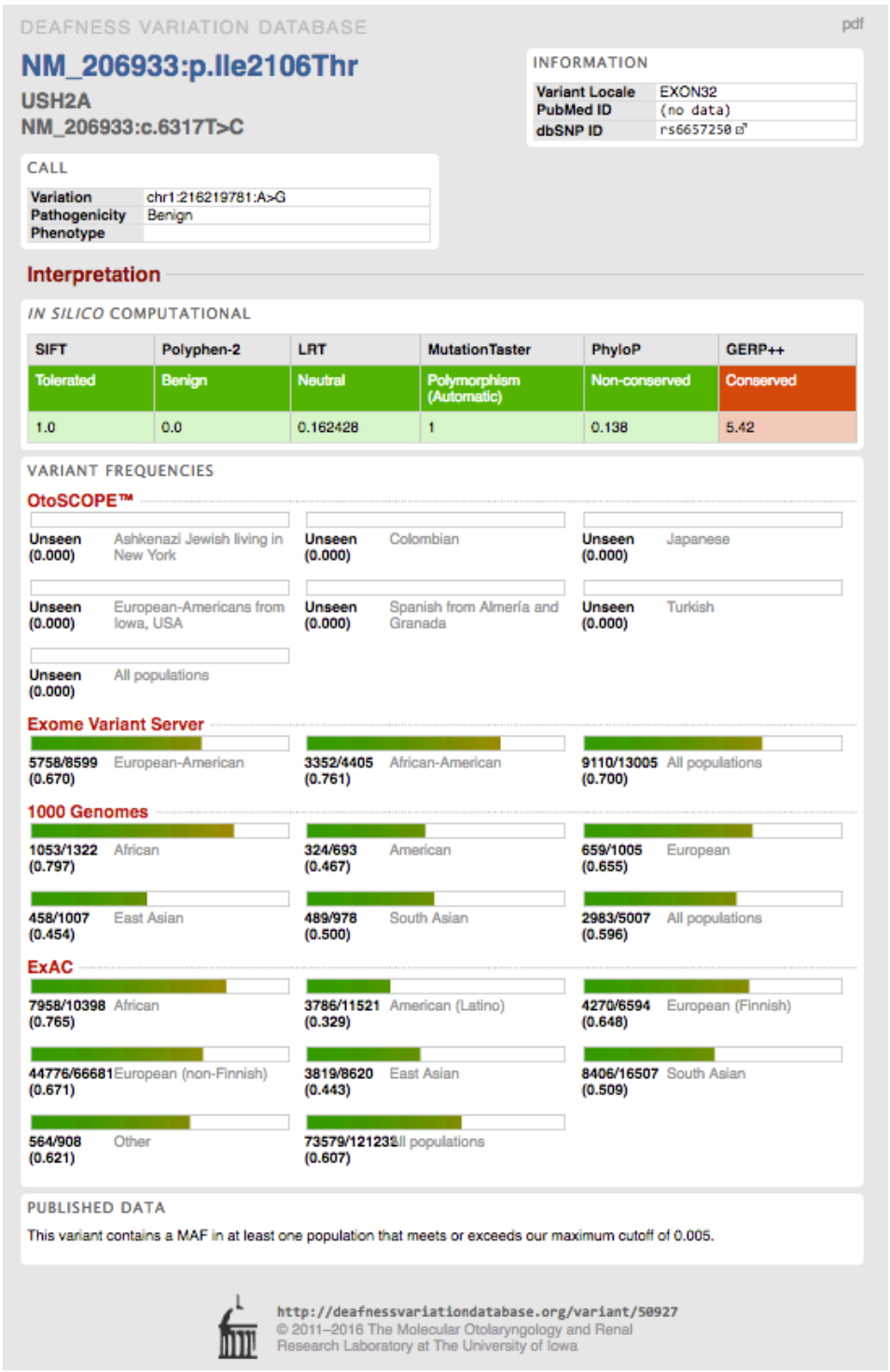


Figure 6. Public Variant View

This is a variant view for each of the variants in the Cordova database that describes all the attributes associated with that variant in the above format and allows the download of the page as a PDF file.

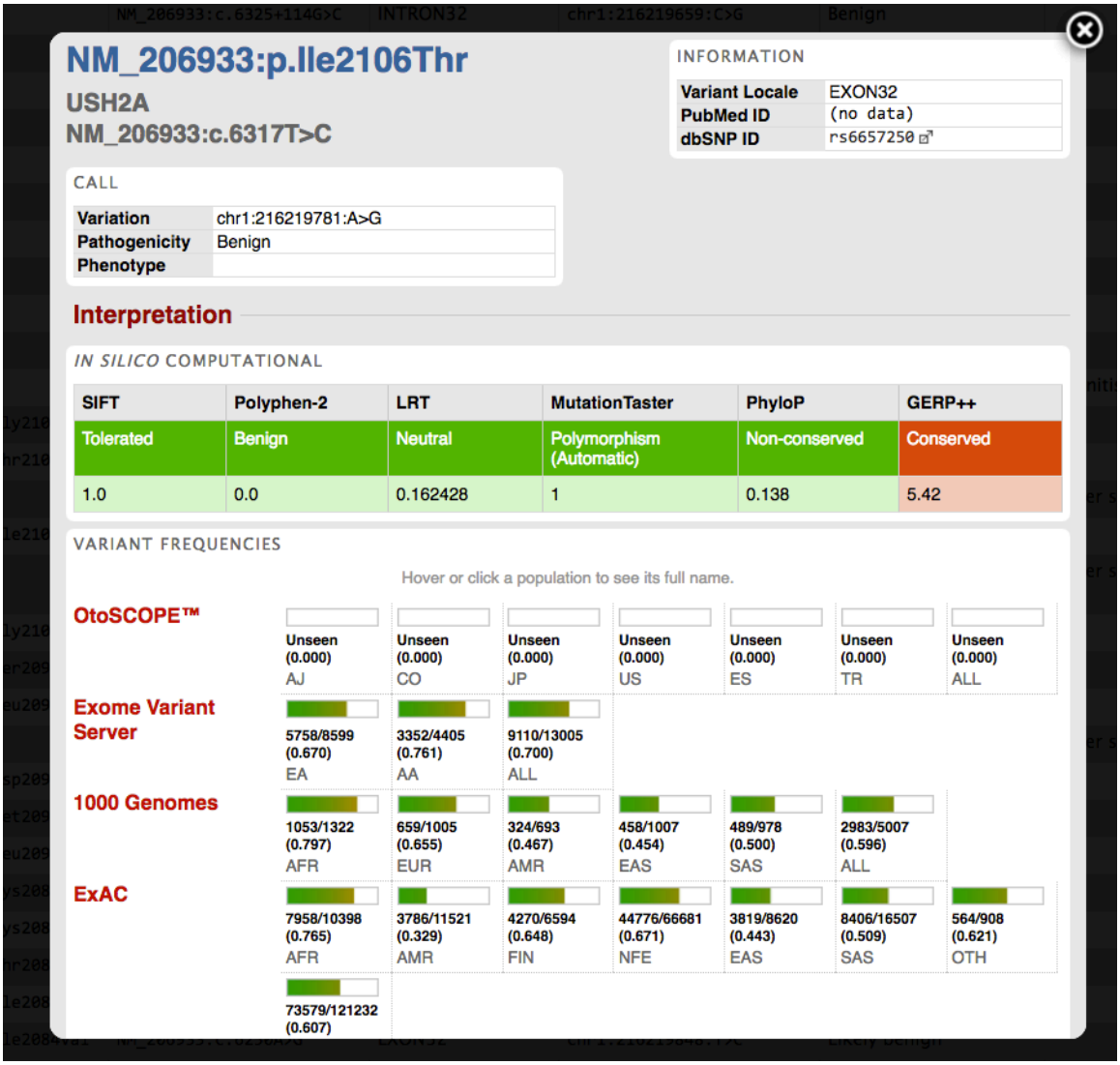


Figure 7. Public Variant View Pop-up

This pop–up window describes the same information as in the variant view, however in a scrollable pop up window to eliminate navigation away from the underlying page and page reloads.

The variant level view in Figure 5 describes each variant's gene name, description of the genetic variant using HGVS [8] nomenclature, description of the protein using HGVS nomenclature, and the pathogenicity classification. This pathogenicity classification is generated from the local analysis Kafeen pipeline, that implements an algorithm to summarize the following: public prediction scores; phenotypic data; and variant locale

(intron, exon, genomic coordinates). In Figure 6 and 7 are links to relevant pub-med articles; dbSNP [9] identifier; color coded prediction summary of six prediction algorithms including SIFT [10], PolyPhen 2 [11], LRT [12], MutationTaster [13], PhyloP [14] and GERP++ [15]; a table of minor allele frequencies from OtoScope [16], ExomeVariantServer [17], 1000 Genomes [18], and ExAC [19] (each describe up to seven different populations); and finally any notes for the variant are displayed on the interface. At the gene level described in Figure 3 and Figure 4, Cordova allows the user to download the variant data for that gene formatted in several ways including CSV, tab, JSON, and XML.

The secure interfaces of Cordova provide a data and user management system to handle administrative duties described in Figure 8-21. The admin functionality includes the ability to create a user, create a user group, view and edit users and view and edit user groups. Any and all users with accounts that have logged in have the ability to add a variation without annotation, edit a released variant, review changes that are staged for release, edit variants that are not yet released, edit and view their user profile and view an activity log of the site that tracks what users are doing on the site including releasing new variants, editing variants or adding variants to the unreleased variants queue.

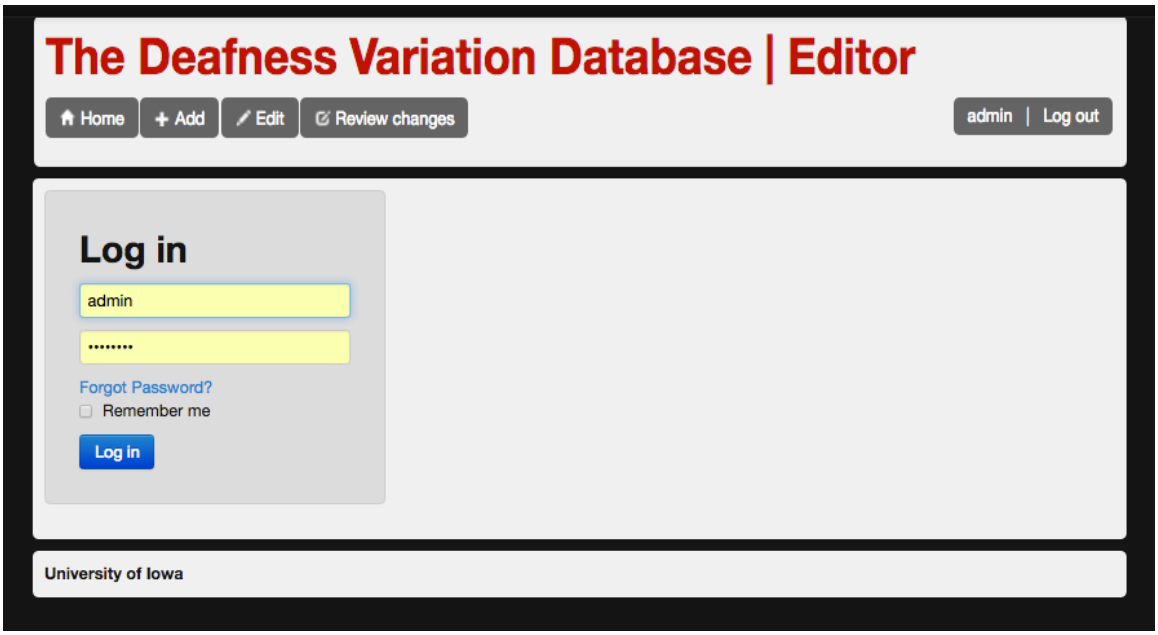


Figure 8. Secure Login Page

The Secure Login Page allows users that have accounts on a Cordova instance to login and interact with the secure functionality.

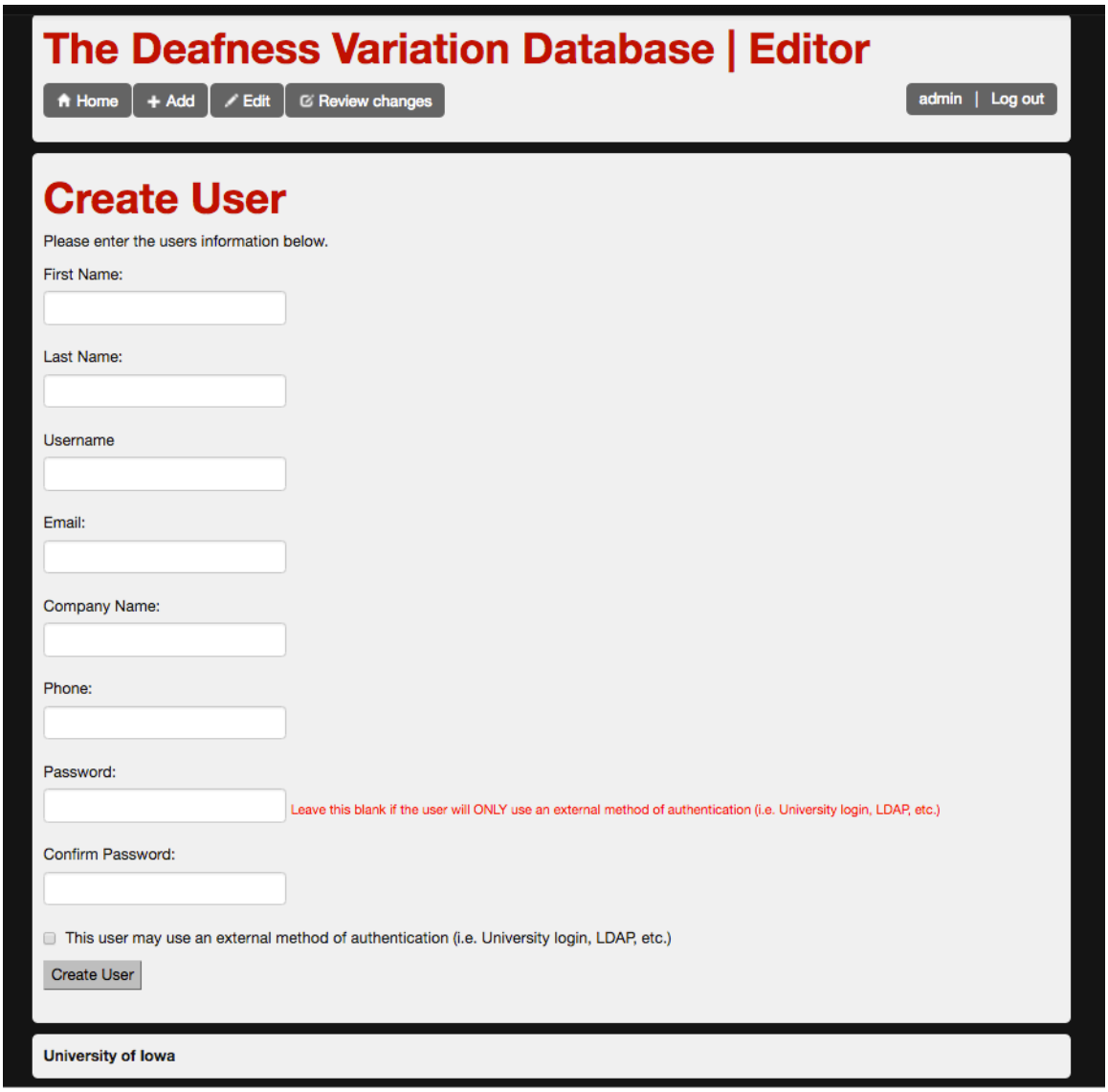


Figure 9. Admin Create User Interface

The Admin Create User Interface allows the Cordova instance administrator to add users to the Cordova instance.

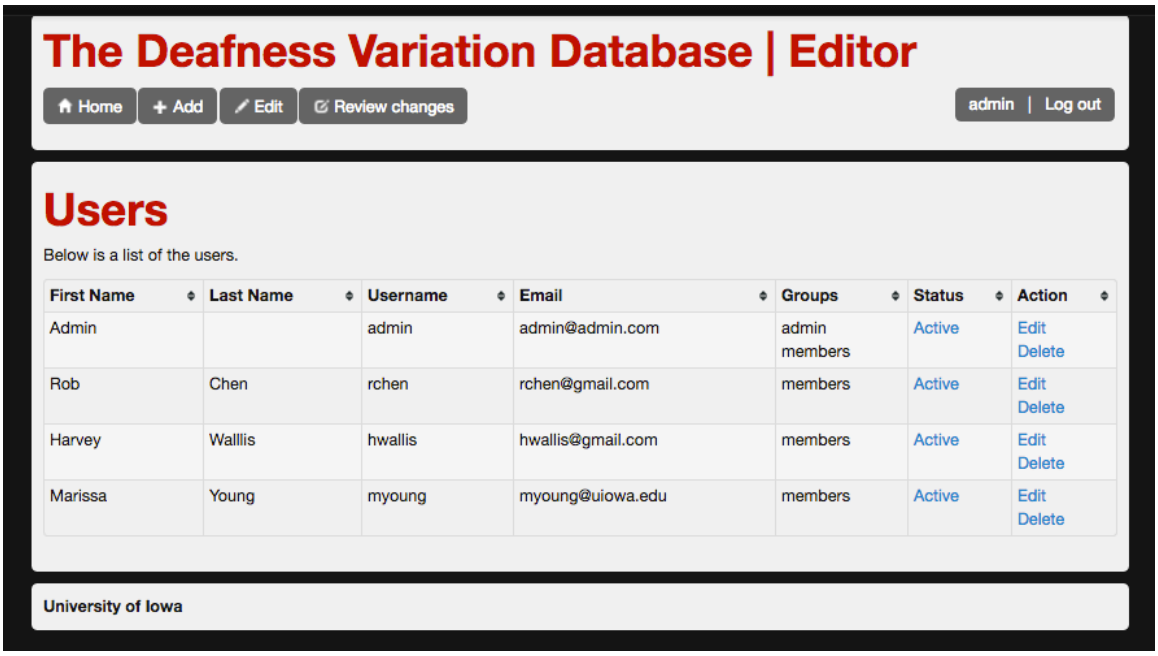


Figure 10. Admin View Users Interface

The Admin View Users Interface allows a Cordova instance administrator to view and manage current users of the Cordova instance by editing or deleting the users accounts.

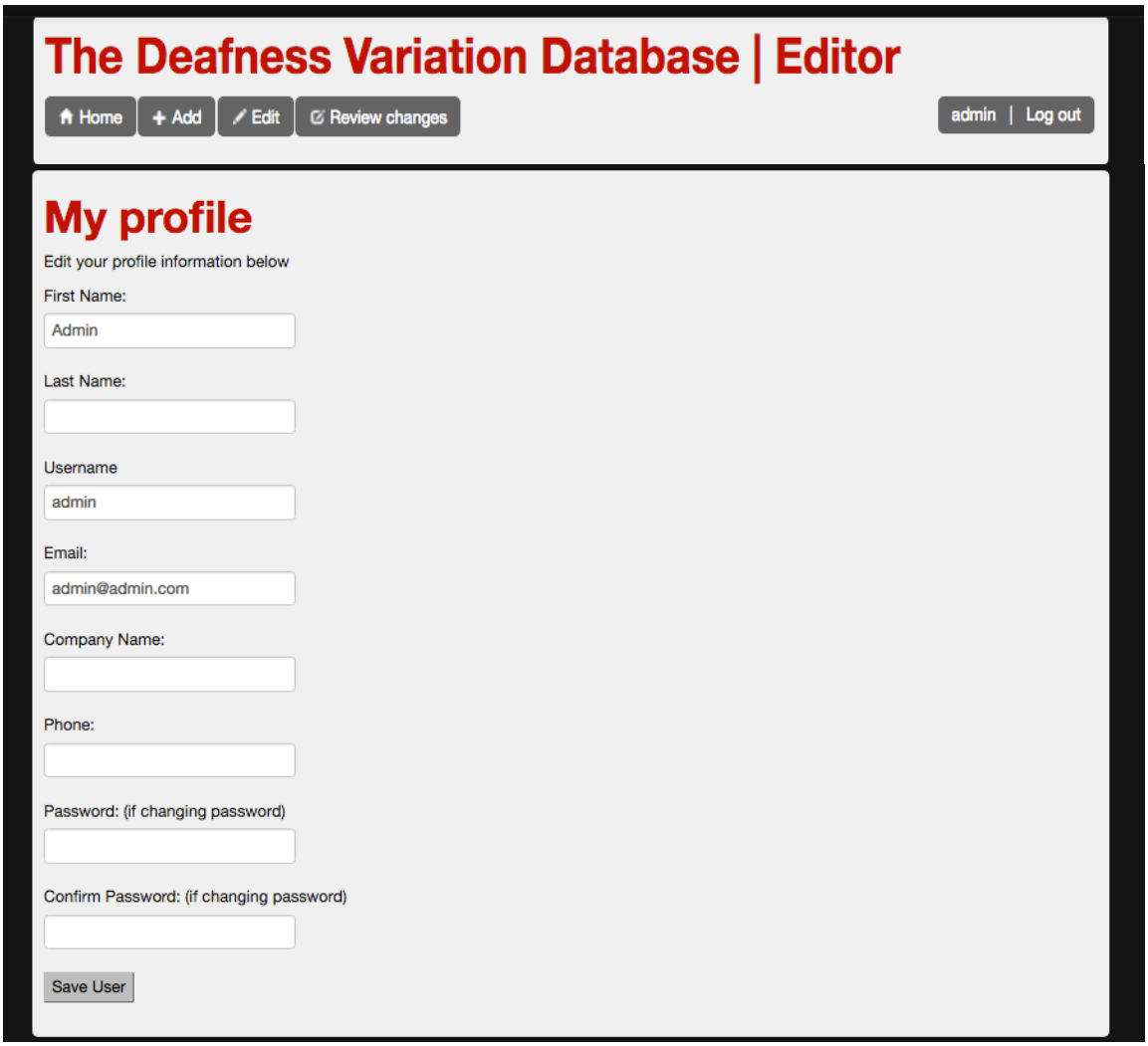


Figure 11. User Profile Interface

The User Profile Interface allows any user to view and edit their profile information including updating passwords.

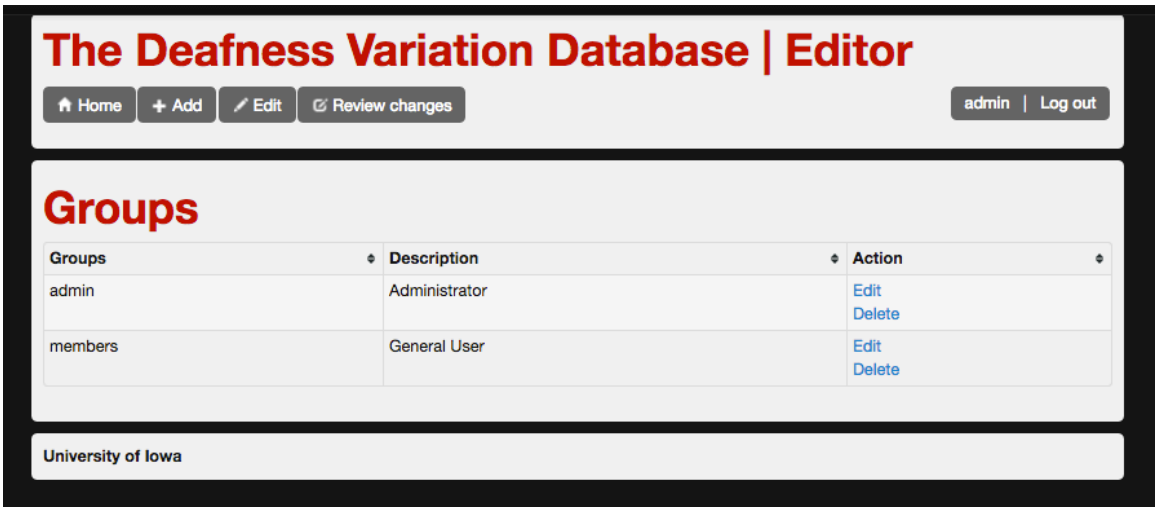


Figure 12. Admin View Groups Interface

The Admin View Groups Interface allows a Cordova instance administrator to view and mange current groups by editing or deleting a group.

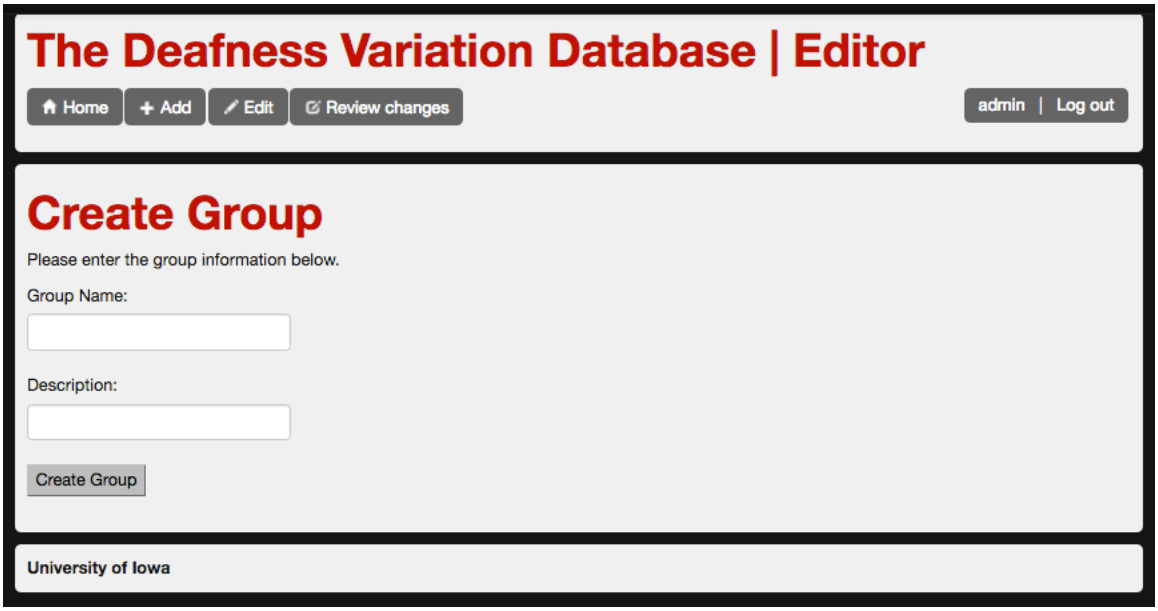


Figure 13. Admin Create Group Interface

The Admin Create Group Interface allows the Cordova instance administrator to add a user group to the Cordova instance.

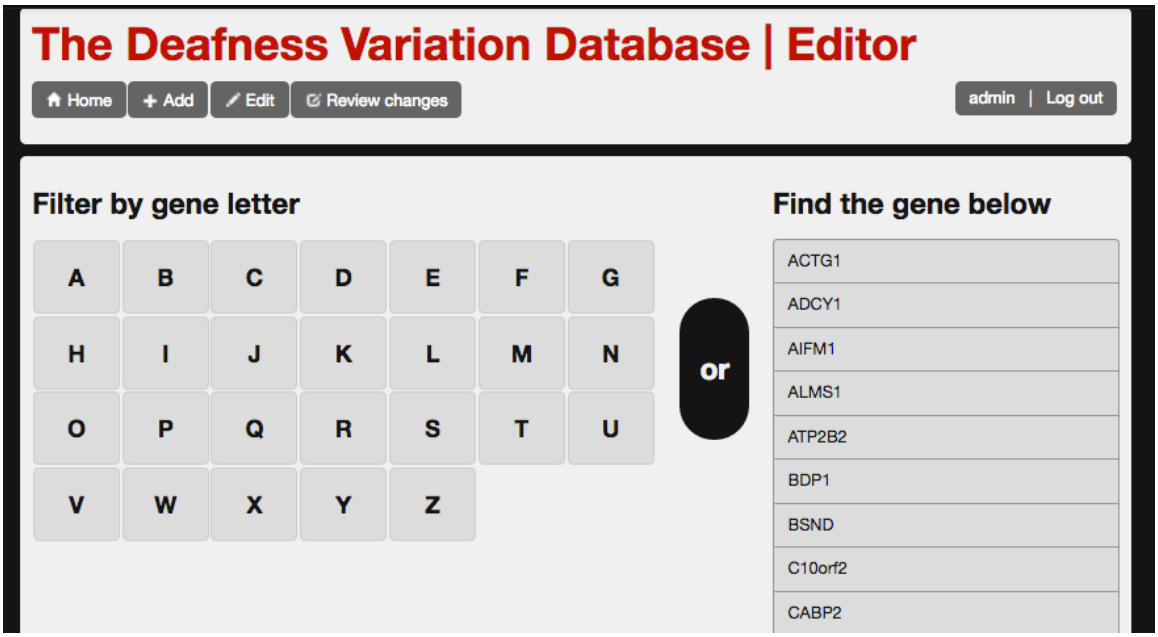


Figure 14. Edit Variant Find Variant By Gene

The Edit Variant Find Variant By Gene Interface allows users logged into a Cordova instance to find a variant for modification in a gene.

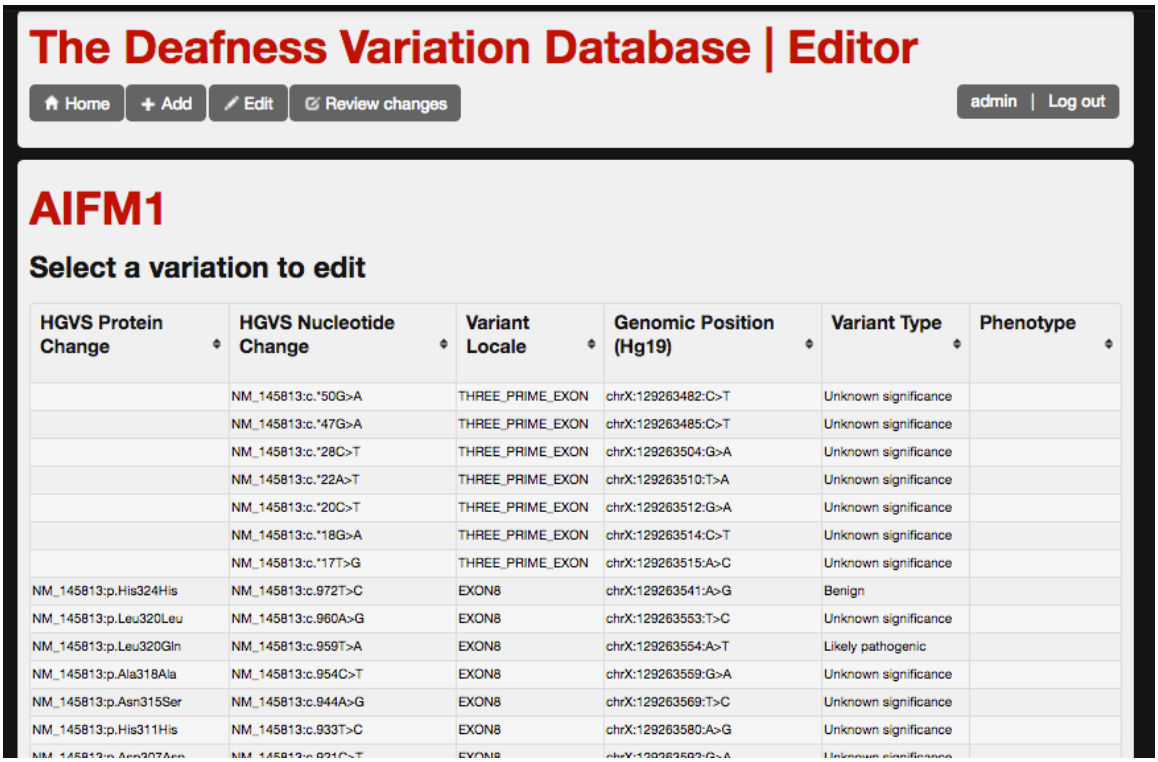


Figure 15. Edit Variant Find Variant Interface

The Edit Variant Find Variant Interface is accessible through the Edit Variant Find Variant By Gene Interface. By selecting a gene on the previous interface the user is redirected here to see the variants in that gene that can be edited. By selecting a variant to edit here the user is redirected to the Edit Variant Interface.

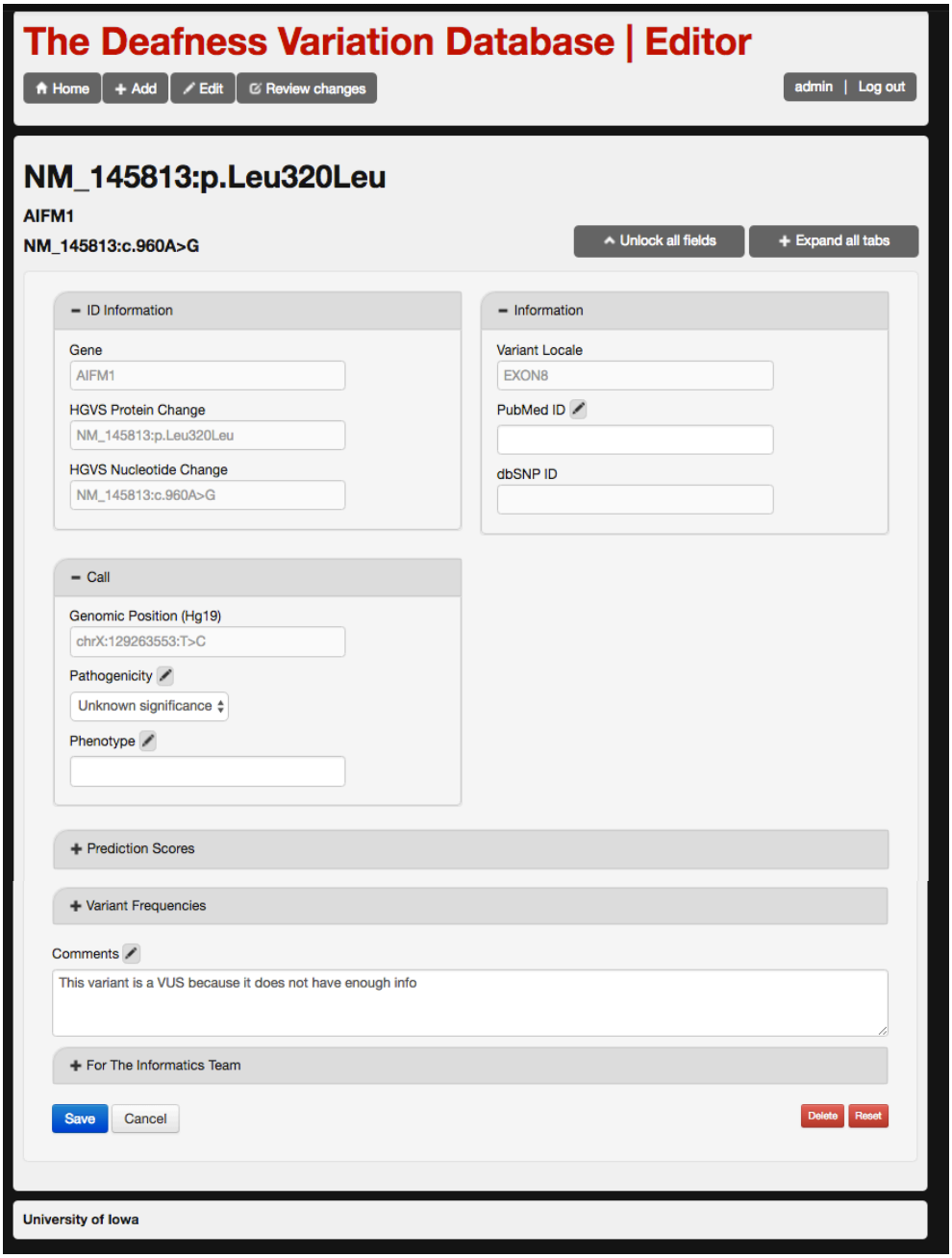


Figure 16. Edit Variant Interface

The Edit Variant Interface is arrived at through the Edit Variant Find Variant By Gene Interface and the Edit Variant Find Variant Interface. Here the user can modify any of the data of a variant that is currently in the variations database table, these are variants that have been released to the public interfaces.

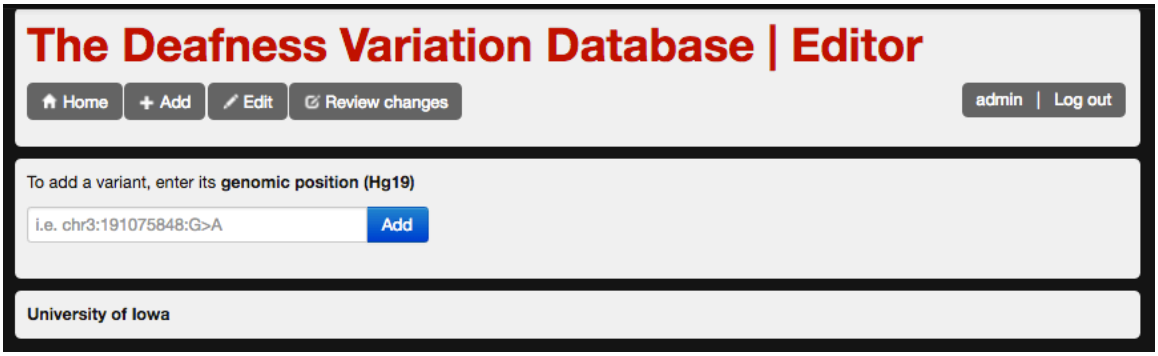


Figure 17. Add Variant Interface

The Add Variant Interface allows a user that is logged in to add a variant to the variations_queue table without any annotations. This variant, being in the variations_queue table will be available for viewing and editing prior to release in the Release Changes Interface. This interface does not add a variation to the live site without it first being confirmed and released.

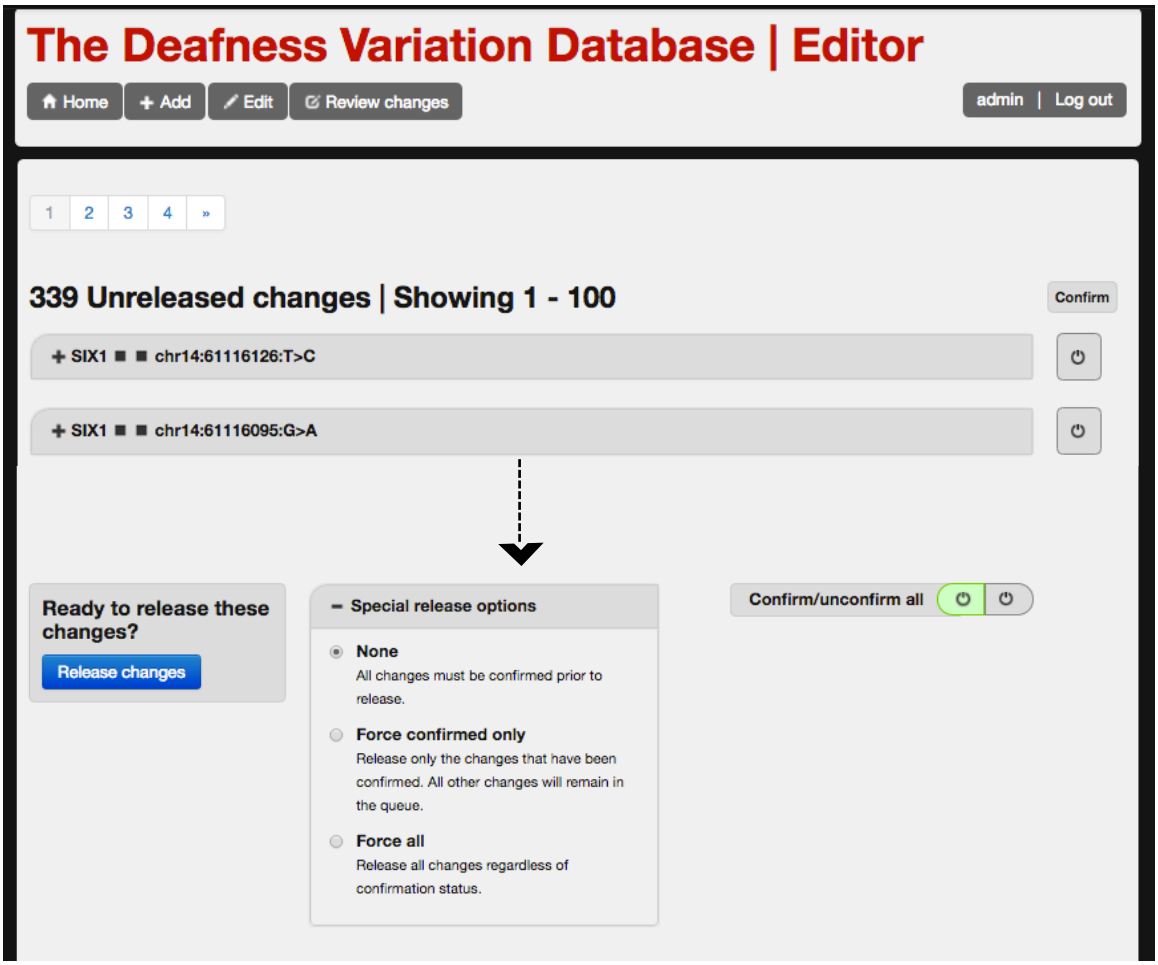


Figure 18. Release Changes Interface

The Release Changes Interface displays all of the variations in the variations_queue table that are staged for release to the variations table where they will be accessible to the public. Here, users can confirm variations for release and make any final edits to the staged variants before release.

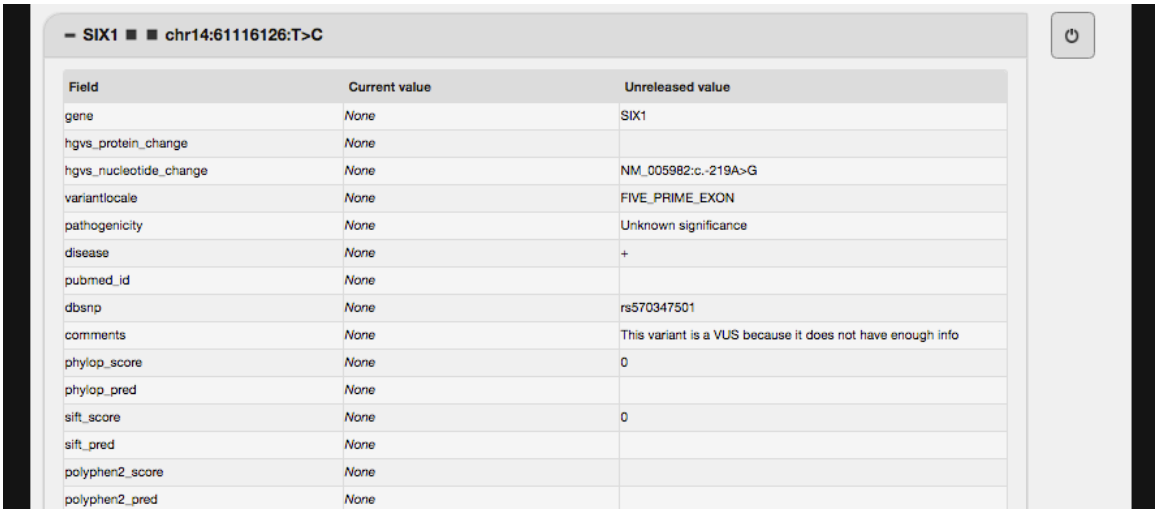


Figure 19. Review Changes Variant Description Table

On the Release Changes interface, a user can expand any listed variation to see current values in the public variations table for that variant and the unreleased values side by side. The variant described above is a new addition and there is no current value for the variant in the public site. The user can access additional information on this variant through the Release Changes Edit Unreleased Variant Interface.

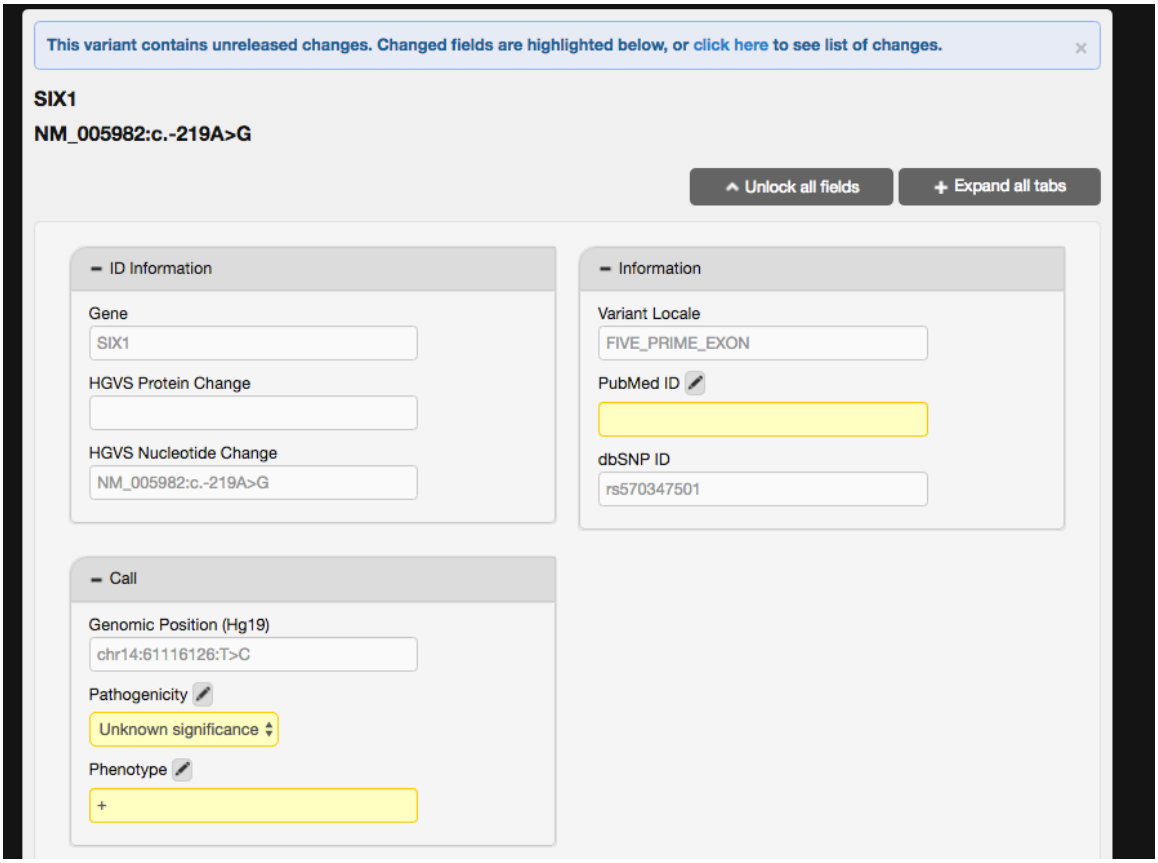
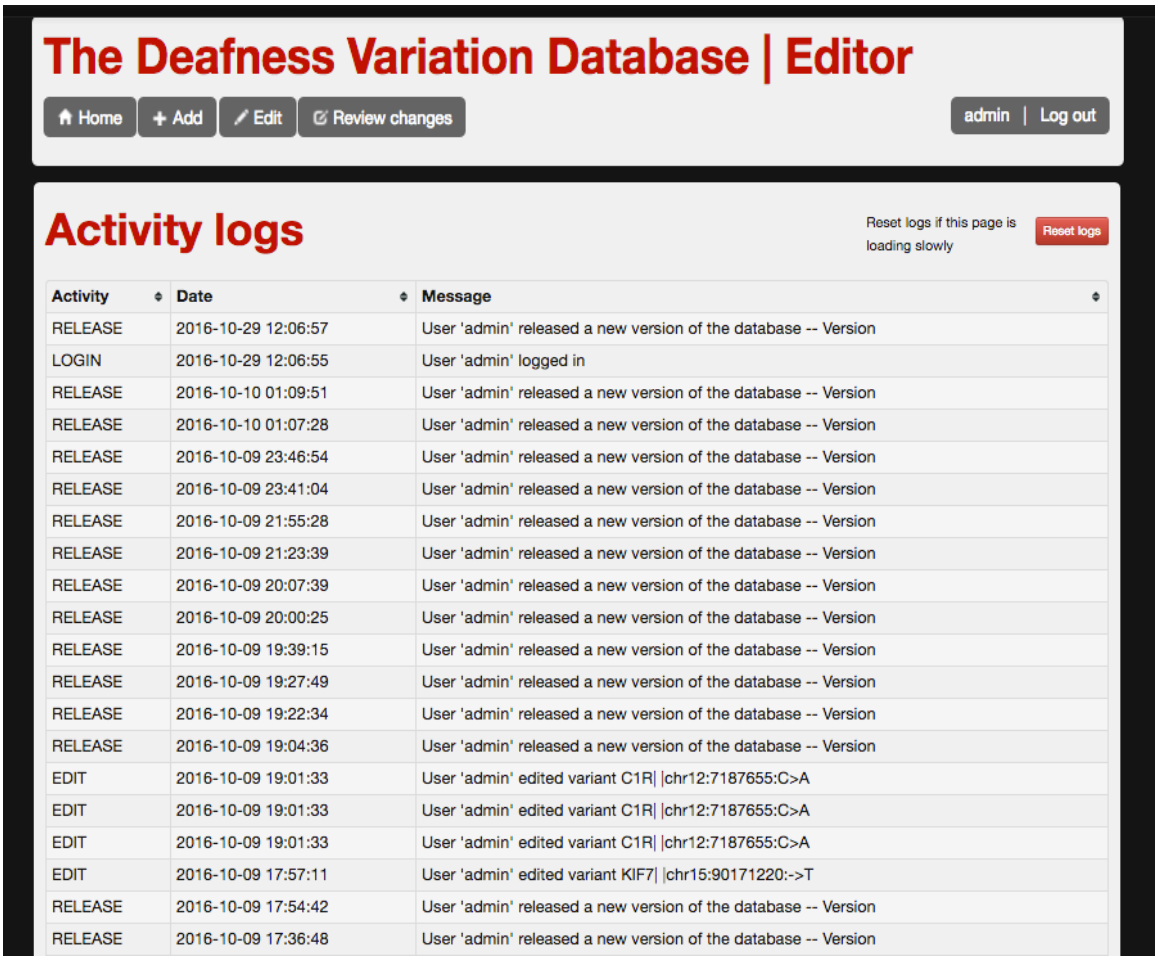


Figure 20. Release Changes Edit Unreleased Variant

The Release Changes Edit Unreleased Variant Interface allows a user to make edits to a variant that is staged for release in the variations_queue.



Activity	Date	Message
RELEASE	2016-10-29 12:06:57	User 'admin' released a new version of the database -- Version
LOGIN	2016-10-29 12:06:55	User 'admin' logged in
RELEASE	2016-10-10 01:09:51	User 'admin' released a new version of the database -- Version
RELEASE	2016-10-10 01:07:28	User 'admin' released a new version of the database -- Version
RELEASE	2016-10-09 23:46:54	User 'admin' released a new version of the database -- Version
RELEASE	2016-10-09 23:41:04	User 'admin' released a new version of the database -- Version
RELEASE	2016-10-09 21:55:28	User 'admin' released a new version of the database -- Version
RELEASE	2016-10-09 21:23:39	User 'admin' released a new version of the database -- Version
RELEASE	2016-10-09 20:07:39	User 'admin' released a new version of the database -- Version
RELEASE	2016-10-09 20:00:25	User 'admin' released a new version of the database -- Version
RELEASE	2016-10-09 19:39:15	User 'admin' released a new version of the database -- Version
RELEASE	2016-10-09 19:27:49	User 'admin' released a new version of the database -- Version
RELEASE	2016-10-09 19:22:34	User 'admin' released a new version of the database -- Version
RELEASE	2016-10-09 19:04:36	User 'admin' released a new version of the database -- Version
EDIT	2016-10-09 19:01:33	User 'admin' edited variant C1R| |chr12:7187655:C>A
EDIT	2016-10-09 19:01:33	User 'admin' edited variant C1R| |chr12:7187655:C>A
EDIT	2016-10-09 19:01:33	User 'admin' edited variant C1R| |chr12:7187655:C>A
EDIT	2016-10-09 17:57:11	User 'admin' edited variant KIF7| |chr15:90171220:->T
RELEASE	2016-10-09 17:54:42	User 'admin' released a new version of the database -- Version
RELEASE	2016-10-09 17:36:48	User 'admin' released a new version of the database -- Version

Figure 21. Activity Logs

The Activity Logs table allows any user logged into a Cordova instance to view activity of all other users on the system including activities such as adding a variant, editing a variant or releasing variants to the public site.

The data used in Cordova spans millions of genomic variants, minor allele frequencies (MAFs), and pathogenicity prediction scores. This data is stored in and supported through a MySQL database described below in Figure 22.

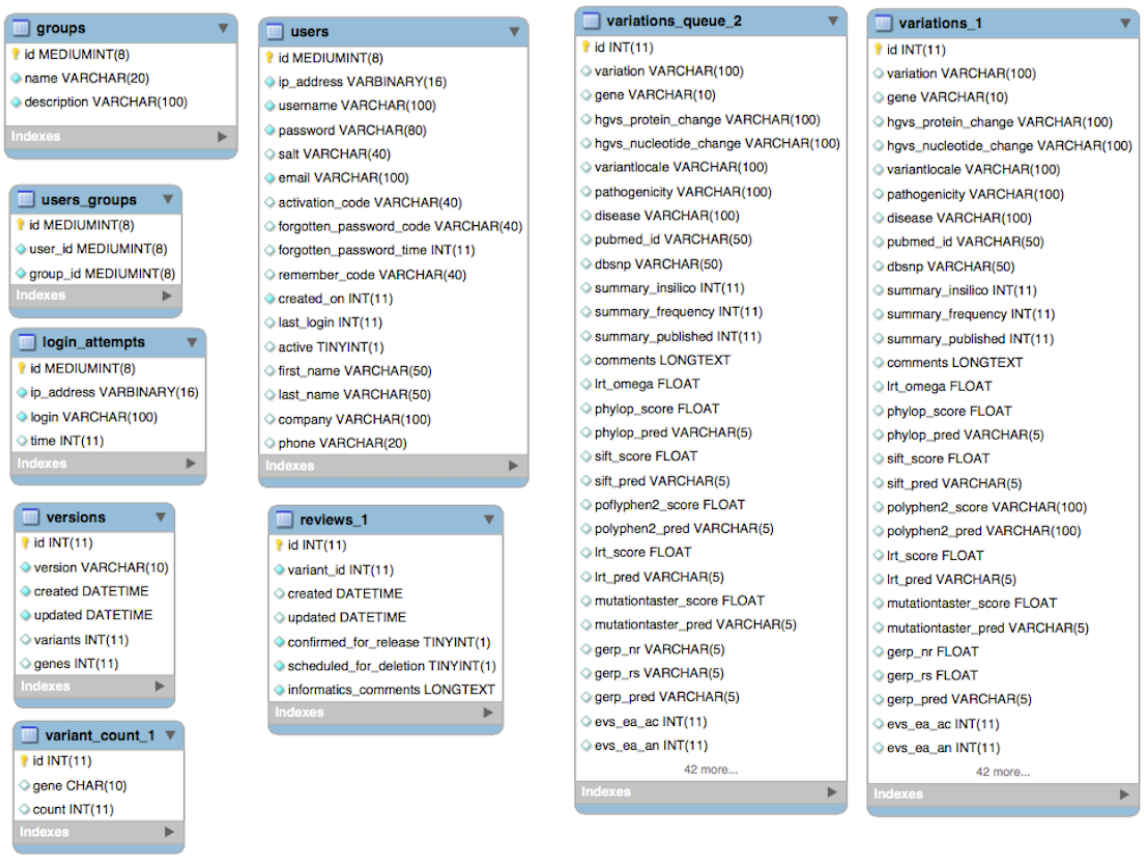


Figure 22. Cordova Database Schema

There are four tables committed to administrative duties. These include: groups, user, user_groups and login_attempts. The variations_queue is used for staging variations to be edited on the secure interfaces. The variations table holds the variations that are shown in the public site. The reviews table is relied upon by the release changes functionality, keeping track of which variations in the queue are scheduled for deletion or confirmed for release. The variant_count table is necessary for variations to display properly in the public site. The versions table is used to keep track of releases of data.

Currently there is a local pipeline involved in collecting formatting and loading this data, and a procedure to periodically update public records used in the pipeline. Each procedure involves several local scripts and manual file management. The collection of applications that acquires variants, MAFs, pathogenicity predictions, and phenotype descriptive terms used in a Cordova instance is called the Kafeen analysis pipeline.

2.2 Kafeen

Kafeen was developed by the authors of Cordova [2][3]. It is a collection of Ruby [30] applications harnessing community tools and public data repositories to collect and format genetic variation data. The Kafeen pipeline is described in Figure 23. Kafeen requires configuration, acquisition and formatting of preparatory data. Data from eight data sources including 1000 Genomes [18], EVS [17], ClinVar [20], ExAC [19], dbNSFP [21], dbSNP [9], HGMD [22] and OtoDB [16] needs to be downloaded and formatted for use with Kafeen and cleaned of duplicates that can arise in and between these data sources.

The input to Kafeen is a list of genes and a configuration file that specifies which data sources to query and what information to pull from each data source. Kafeen first uses bcftools [23] to convert genes to gene regions. These regions are used to query 1000 Genomes [18], EVS [17], ClinVar [20], ExAC [19], dbSNP [9], HGMD [22], and OtoDB [16] for manually annotated variants. It also uses a local Variant Call Format (VCF) file to specify local, manually annotated variants. Next, gene names are added to the variant VCF file. Following that, dbNSFP [21] is queried for 6 pathogenicity predictions and a point-based system is used for determining a final prediction value for variants of unknown significance (VUS) (if >60% of the prediction algorithms predict pathogenic) or likely benign (if >60% of the pathogenicity algorithms predict benign). This score is utilized later in the pipeline. These prediction tools use different nomenclature and conventions to report pathogenicity and therefore undergo normalization. The normalized dbNSFP score is used by Kafeen. However, dbNSFP [21] does not provide an interpretation for GERP++ [15] or PhyloP [14] and must be normalized through

preprocessing in the scripts. In all cases, if multiple predictions are presented for one variant, the most pathogenic is used. Variant Effect Predictor (VEP) [24] is executed to collect coding regions and, when available, protein HGVS nomenclature, in addition to the variant and mutation functional effect. All collected data up to this point is used to make a final pathogenicity prediction [3]. This final prediction takes into account ClinVar [20] pathogenicity annotation, HGMD [22] pathogenicity annotation, MAF, expert curated pathogenicity, and the collected pathogenicity predictions from dbNSFP [21]. Kafeen generates a VCF file summarizing the collected and computed data. This VCF file can undergo preprocessing and formatting for a manual MySQL [6] "file load" into a Cordova instance.

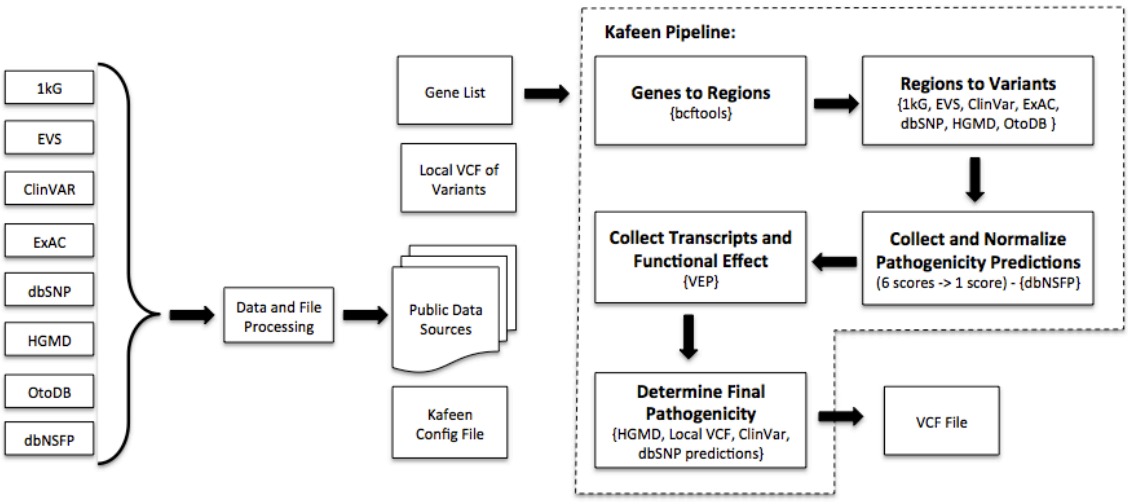


Figure 23. Kafeen Pipeline

This figure describes the data that is required to execute Kafeen and breaks out the Kafeen pipeline in to fives steps leading up to the generation of a VCF file consisting of data collected from the public data sources and computed final pathogenicity scores.

CHAPTER 3 – METHODS

To achieve a design solution for collection, management and automation with respect to the distribution of genetic variation data, missing features for the Cordova system are identified. These features are largely related to automation and data management.

The new system features are implemented with the PHP CodeIgniter [7] framework that employs the use of a Model View Controller software architecture. The use of PHP CodeIgniter [7] allows for incorporation with the existing Cordova application. Bootstrap 2.0 [25] was used as a CSS styling framework and MySQL [6] database is used as the database management system. New views were implemented to allow for users to interact with the data without intimate knowledge of the underlying system, pipeline, or data flow. Each of these views has a corresponding controller. Each controller relies upon several new models developed to act as an interface between the application and the database system. The dataflow and data management was updated in Cordova to allow for persistence of data and the database schema was modified to allow for persistent and accessible versioning of data.

The description of these new systems components are detailed in Chapter 4.

CHAPTER 4 - VARIANT-CADI

Variant-CADI consists of integrated interfaces which tie closely with existing user interfaces in Cordova. Variant-CADI interfaces include gene upload and variant collection, nomenclature normalization, and expert curation. These interfaces are integrated with existing gene editing interfaces in Cordova, and are tied closely with the existing review changes interface in Cordova. The System Requirements described in Table 1 were not changed from the existing system requirements previously required of Cordova, which will translate well for users already using a Cordova instance. Figure 24 describes a high-level view of how Variant-CADI encapsulates much of the manual data management and flat file data storage into one system that is incorporated into Cordova. The flat file management is incorporated through integrated database storage and Kafeen is integrated though a user interface in Cordova. For more in depth details for this encapsulation refer to Figure 1 and Figure 2 in Chapter 1. Figure 25 highlights where changes in the Cordova system were implemented and which features underwent updates to support the new functionality and the increase in data seen through the release changes interface.

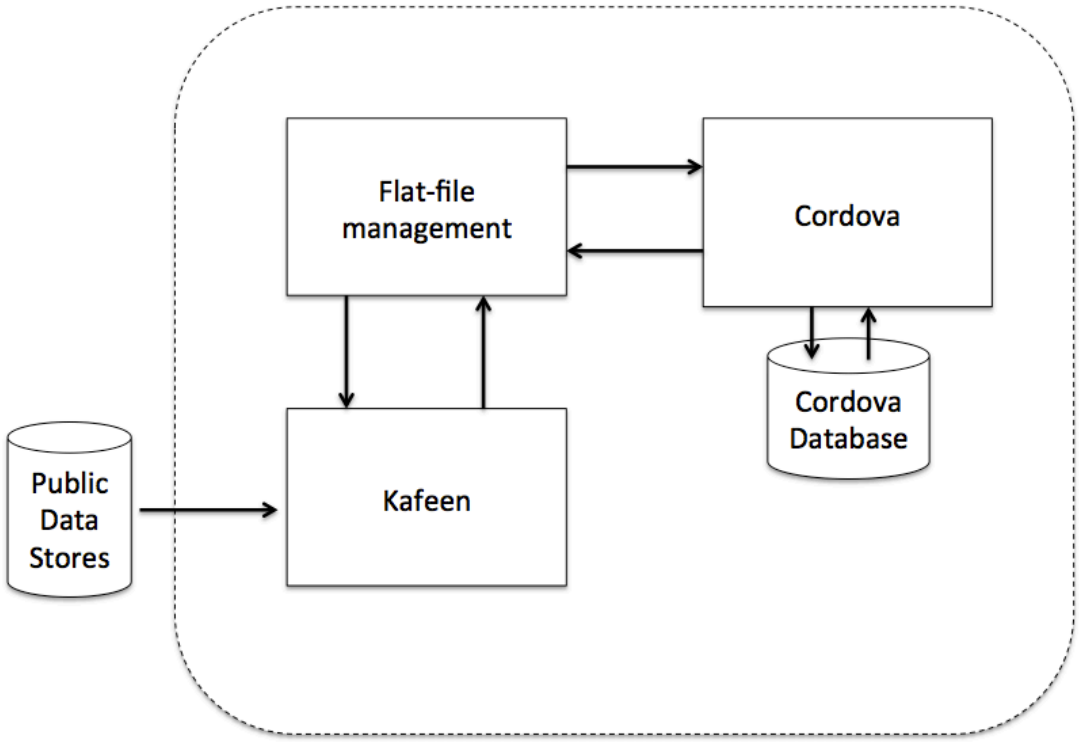


Figure 24. Variant-CADI Overview

Variant-CADI implements functionality to tie together data collection (Kafeen), data management (flat-file), and data distribution on the web (Cordova) into one coherent software application.

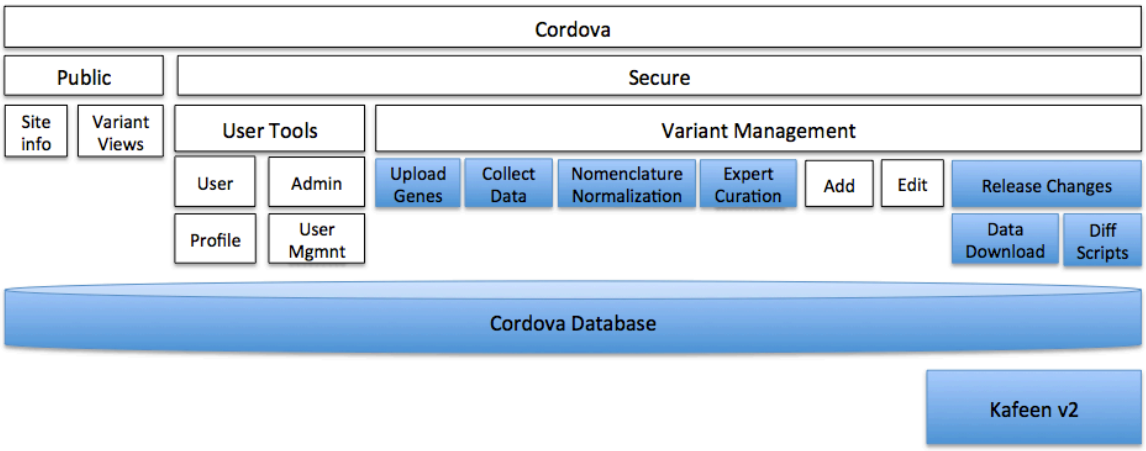


Figure 25. Cordova System with Variant-CADI Functionality

The Cordova system with implemented Variant-CADI functionality significantly expanded functionality under variant management. These additional functionalities have been highlighted in blue. They include the ability to upload a gene list for variant collection and annotation, normalize the nomenclature of variants queued for release (also called "staged" variants), add, manage, and apply expert curations to queued variants, downloads of the staged data and perform and provide reports on how staged variants differ from the public variants in the data base. This allows users to precisely specify what is going to be released to the public-facing interfaces. Updates to how data is released and stored resulted in updates to the Cordova database and release changes. The Kafeen version compatibility was updated to the most recent version.

Table 1. System Requirements

Suite	Version
PHP	5.5.0
CodeIgniter	3.1.0
Bootstrap	2.0.0
Apache	2.2.0
MySQL	5.0.95
Ruby	2.1.5
Kafeen	2.0.0

4.1 Gene Upload

The Gene Upload interface is meant to collect a list of genes of interest from the user. Two input formats may be used. File upload of line separated gene symbols is the first option. This option is available for users who already have their list of genes in an excel file or have a large list of genes. There is no inherent limit to the text box, but copying and pasting more than a few hundred genes can be tedious and error prone. The second option is a text box input. The text box takes in a list of line separated gene symbols. This option is ideal for users who will only insert a few genes. Upon submission to either upload option the user is redirected to the variant collection interface. The dataflow is described in Figure 26 and the interface is described in Figure 27.

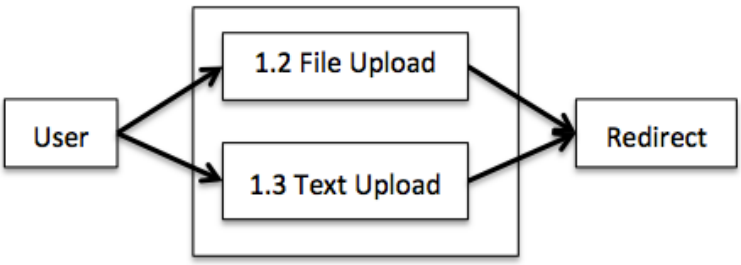


Figure 26. Upload Genes Data Flow

To upload a list of genes the Upload Genes interface was implemented. This figure represents the data flow for the Upload Genes interface. The Upload Genes interface allows users to upload a list of genes as a file or through a text box interface. Upon submission of either form, the user is redirected to Variant Collection and Annotation.

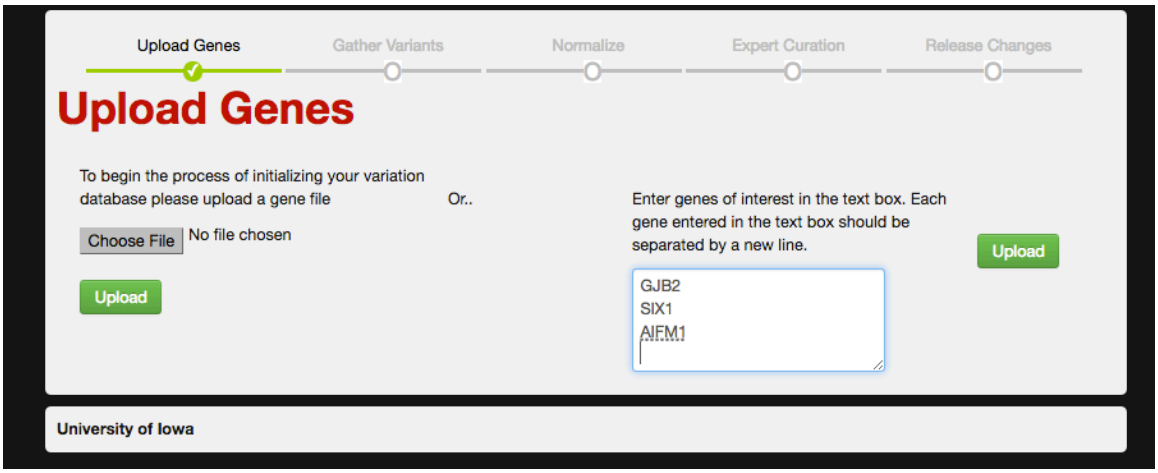


Figure 27. Upload Genes Interface

Upload Genes allows users to insert a list of genes for collection through the Kafeen pipeline. This list of genes can be uploaded through a text box or a file.

4.2 Variant Collection

The Variant Collection interface provides the user with an opportunity to review and verify the list of genes they entered before submitting it for variant collection. It gives the user further instruction as to what to expect in variant collection and notifies the user that the system administrator will receive an email when the collection is complete. Variant collection can take hours, so this process is executed in the background on the server to avoid the complication of loss of internet connection between the browser (client) and the server. At the end of collection a notice is sent to the user via email with a link to continue with the next steps in data curation. The dataflow is described in Figures 28 and 29 and the interface is described in Figure 30.

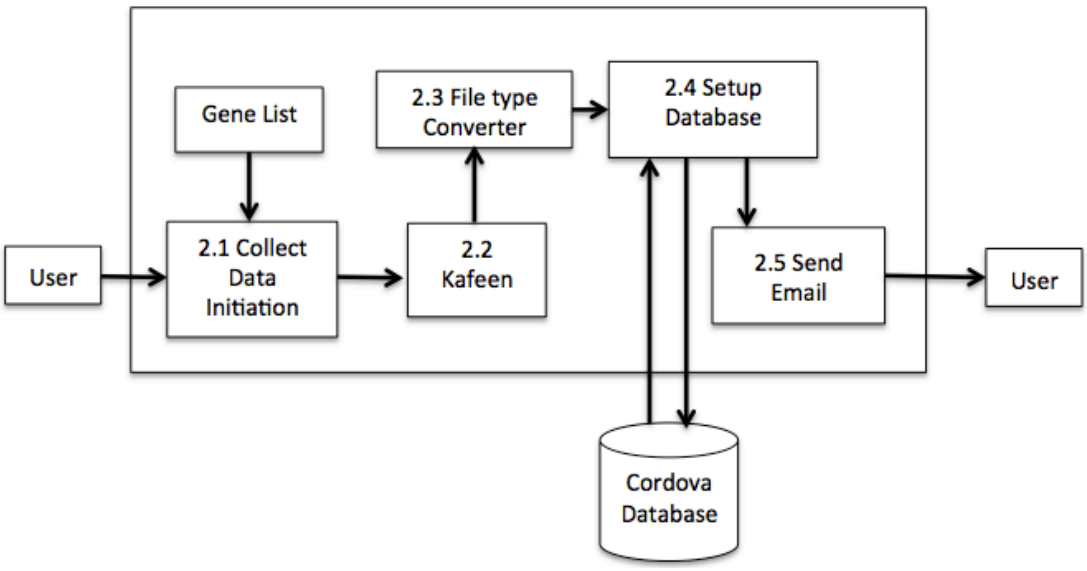


Figure 28. Variant Collection Data Flow

The user interface simply displays the file or list of genes the user uploaded in Upload Genes and a button to proceed. When the user presses the confirmation button a message is displayed to the user notifying them that the variation collection and annotation process has begun and that the Cordova administrator will be notified when collection is complete. On the backend, Kafeen is executed with the list of genes provided and the output is converted to a tab file. This tab file is used to setup the Cordova database. Upon completion an email is sent to the system administrator with information on the Kafeen execution and instructions to further manage the recently uploaded data.

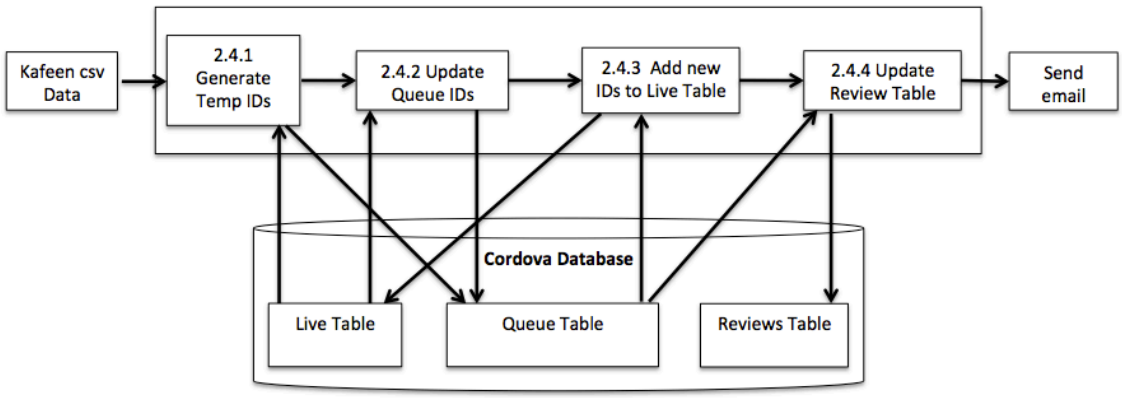


Figure 29. Setup Database Data Flow

Setup Database describes at a high level what needs to be performed to instantiate a Cordova database from a tab file. Temporary ID's must be generated for the variants that will be entered into variations_queue table and the ID's will begin at the maximum ID in the variations table, plus one. These are then inserted into the variations_queue table in an update or add fashion. Next, any variations that exist in the variations and also variations_queue table must have matching ID's. Any variations in variations_queue and not in variations table will have their ID added to variations. Next, any new additions to variations_queue will be reflected in the reviews table.

Figure 30. Variant Collection

The Variant Collection Interface allows users to review their input before initiating variant collection on the system.

4.3 Normalize Nomenclature

The Normalize Nomenclature interface allows users to specify a desired nomenclature for each unique phenotype present in the queue table. The user can interact with this data in one of two ways including file or text input. For text input, each unique phenotype is displayed next to a text box where the user can enter their preferred phenotype descriptive term. Once complete, the user will submit this form and the user's input will be used to update the variant phenotype in the queue table. As a second option, the user can download a file with the disease phenotypes and upload the file, edited to reflect their desired nomenclature. As a result each phenotype is updated in the queue table. To verify these changes have been made the user can download the queue table data from the release changes interface. The dataflow is described in Figure 31 and the interface is described in Figure 32.

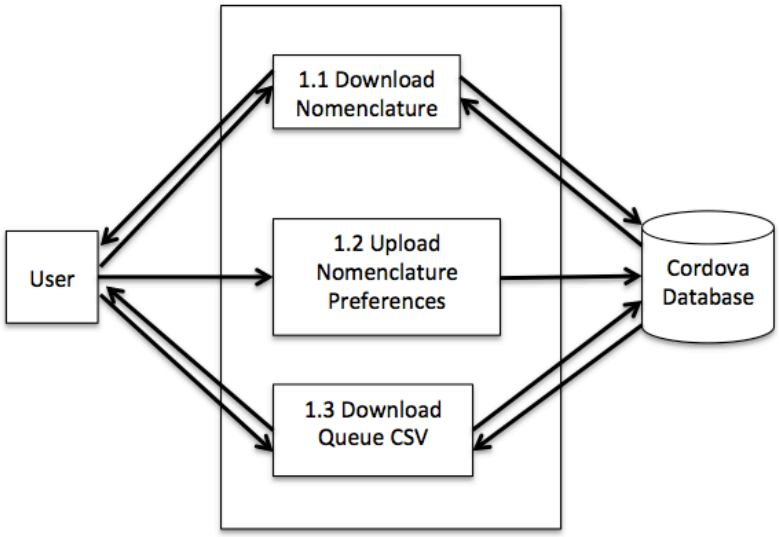


Figure 31. Normalize Nomenclature Data Flow

The Normalize Nomenclature interface displays an upload form, a file download link, and a list of unique phenotypes found in variations_queue table. Each of these has a text box for preferred nomenclature. The normalize nomenclature interface provides the user three features. One is to upload nomenclature preferences -- this can be done through the text boxes on the page or through a file upload. Two, a download nomenclature link will provide the user with a file of the unique phenotypes per gene in a csv file formatted for upload once edited with preferred nomenclature. Three is the ability to download a csv of the entire variation_queue table for inspection before or after nomenclature normalization.

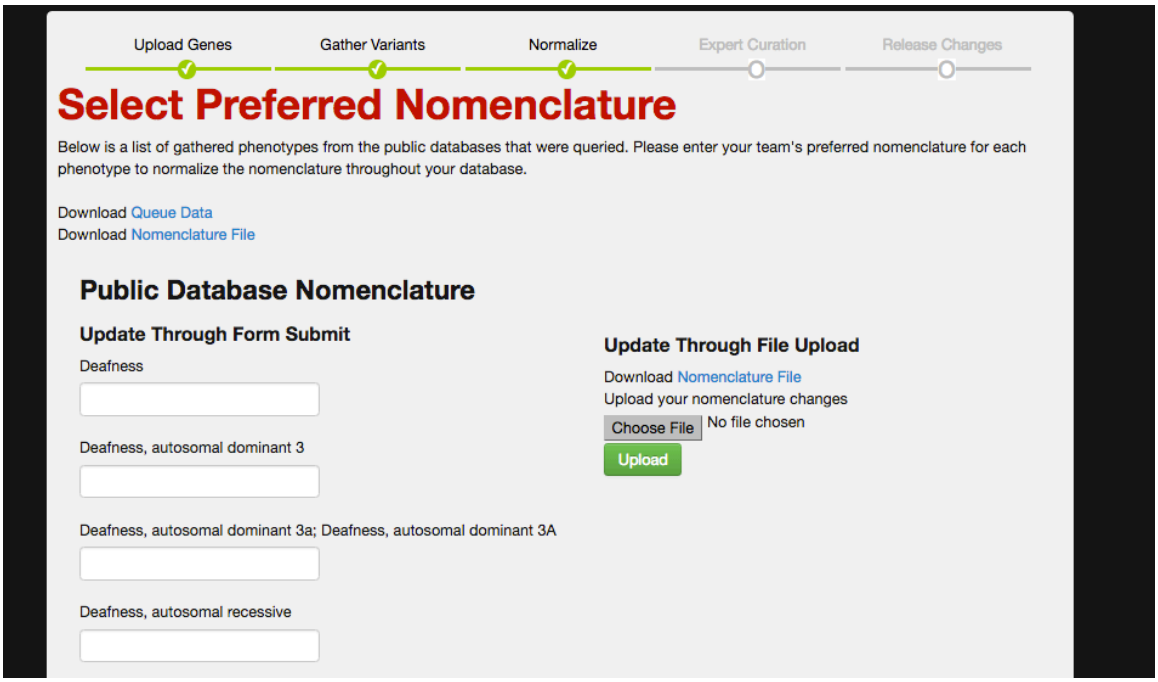


Figure 32. Normalize Nomenclature Interface

The Normalize Nomenclature Interface allows users to edit collected disease nomenclature to standardize the terms used throughout.

4.4 Expert Curation

Expert Curation allows the user to update data in the queue table to reflect their scientific observations. The attributes available for update include pathogenicity, phenotype and PubMed ID. All of these attributes were selected based on existing workflows from genetic testing laboratories and the need of each attribute to be updated from the default collected sources. This also allows for linking the users' own results (or papers) to a variant. The user can upload their expert curated data as a file. This data is saved in the expert data table. When changes are made to this data by uploading variants, matching variants are updated in the expert table and previous variant information is saved in the expert log table. The data in the expert table and the expert log table is available for download and review. The user can apply these expert curations to the collected variation data in the queue table by pressing a button. To verify that these changes have been made, the user can download the queue table data. The dataflow is described in Figure 33 and the interface is described in Figure 34.

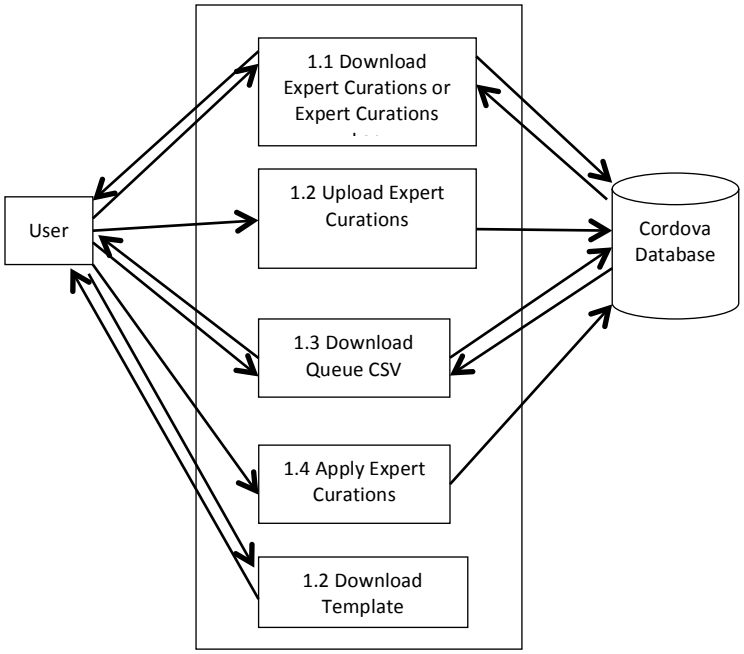


Figure 33. Expert Curation Data Flow

The Expert Curations interface provides functionality to add any number of expert curations to a Cordova database table, expert curations. The user can provide and edit these expert curated variations through a file upload. The user can download the existing variations in csv format from the expert_curation table for inspection. The user can download a file template to format data uploads. The user can also download a csv of the queue table. The user can apply expert curation to variations_queue table. Whenever a variant is changed in the expert curations table, it is added to the expert curations log table to allow for tracking of changes. A download of the log is also available for inspection.

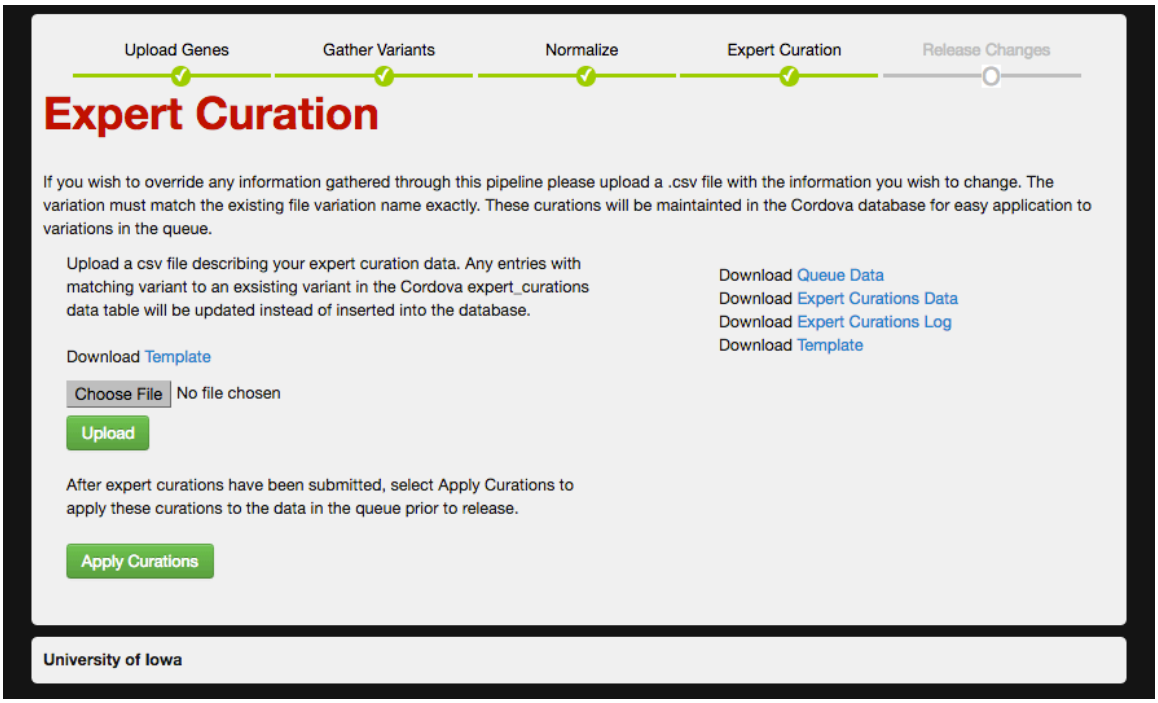


Figure 34. Expert Curation Interface

The expert curations interface allows users to manage and apply expert curations to variations in the queue.

4.5 Review and Release Changes

Release Changes is an updated interface of Cordova to view data that is about to be released to the public site and allows users to make any last edits to the staged data. It allows users the ability to see side-by-side current public data and data staged for release, make any changes to the staged data and select any subset of staged data the user wants to release to the public site. With a press of a button, the data is sent to update the public facing pages and the site is either initialized or updated. Added functionalities here include data version tracking. Additional database tables have been put in place to act as a log of revisions of data. Upon release of the data, updated, added or deleted data is added to the log table. Users can download data in the log table to review changes in data over time. A link to download summary statistics of what will be updated upon release of this data was implemented. This provides the user with statistics including the number of variants that will be updated, added or released, the number of variants in each pathogenicity category, each of these numbers broken down by gene, and which variations are in each of these categories. The dataflow is described in Figure 35 and the interface is described in Figure 36.

Previously, Cordova only supported a single variation addition and much fewer variations were being reviewed at one time. The functionality and data flow of the release changes interface was updated to process multiple variations. The versioning system was given an overhaul, there is no longer a need to re-create and re-name tables to maintain versioning. The algorithm for release of data was rewritten to handle the much larger datasets now seen through this interface. The release changes dataflow is described in Figure 37.

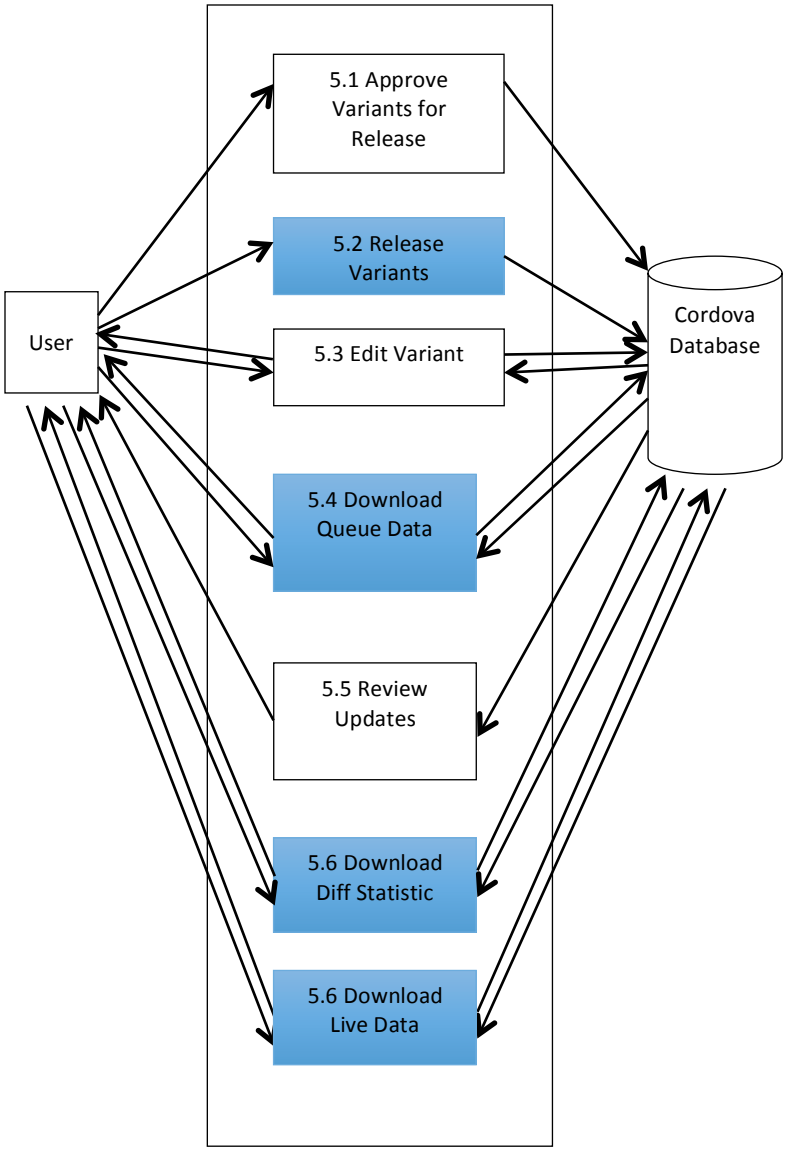


Figure 35. Release Changes Data Flow

Four additional features (depicted in blue) were added to the Release Changes interface. These include the ability for the user to download a csv of variations_queue and variations table for inspection or to perform a difference analysis between variants in variations and variations_queue table. The user is provided with a file of summary statistics of these differences. In addition the release changes functionality was changed to an update procedure from a destroy and recreate procedure. This was necessary because the previous system was not designed to handle the volume of data that is encountered with batch loading (100's of genes, millions of variants).

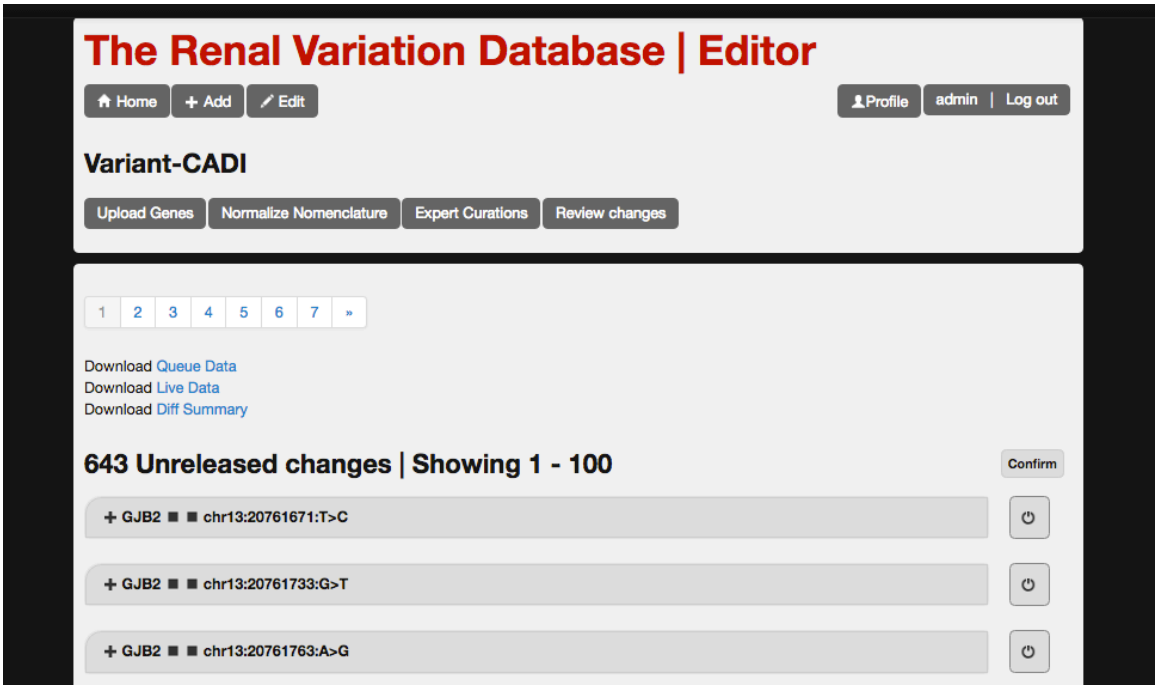


Figure 36. Release Changes Interface After Variant-CADI

The Release Changes interface shows the information available for download by the user. Links for downloading summary "difference" statistics and the variations table and queue table each as a csv are now available.

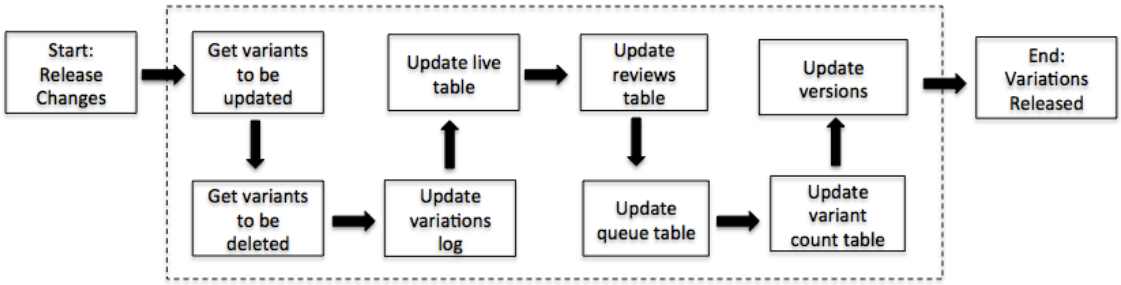


Figure 37. Release Changes Dataflow After Variant-CADI.

The steps necessary to release the data are still required but the major difference is that each step is represented as one or two SQL queries allowing for the MySQL database optimizer to optimize the queries. The previous version performed update operations for each variant one by one. Updating the procedure resulted in substantial speed up of the release of data as the datasets expanded.

4.6 Database Updates

Two additional tables were added to support version tracking for both the public variants and the expert curated variants. In addition one table was added to maintain expert curations through the Cordova instance. Five attributes were added to the three variations tables to further support search functionality and logging. These changes are presented in Figure 38.

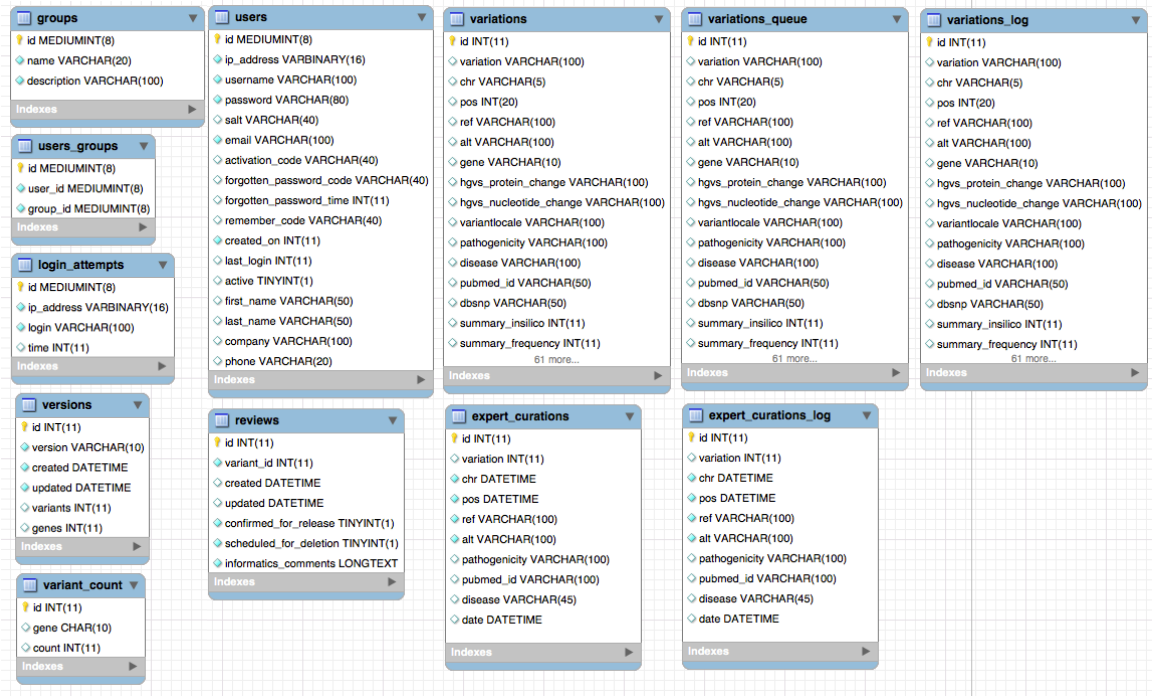


Figure 38. Cordova Database Schema After Variant-CADI

Three additional data tables have been added to the Cordova database schema. These include variations_log, expert_curations and expert_curations_log. The log tables are filled when either the variations table or expert_curations table experiences a change. When a variant is added, deleted or updated, the old record being replaced is added to the corresponding log table. There was also an addition of five attributes to variations, variations_queue and variations_log. These include chr, pos, ref, alt and release_date. The first four provide an additional mechanism for searching for variants in the database. The release_date attribute is used to track changes over time in the variations_log.

CHAPTER 5 - OUTCOMES

Variant-CADI has been integrated into the existing Cordova system and is available through GitHub at https://github.com/clcg/cordova. As a result of this work, Variant-CADI, several new variation databases have been created to demonstrate the utility of this software system. These include the Renal Variation Database [26] (http://cordova-dev.eng.uiowa.edu/cordova_sites_ah/rdvd), the Head and Neck Cancer Variation Database [27] (http://cordova-dev.eng.uiowa.edu/cordova_sites_ah/cvd), the Vision Variation Database [28] (http://cordova-dev.eng.uiowa.edu/cordova_sites_ah/vvd) and the Cleft Variation Database [29] (http://cordova-dev.eng.uiowa.edu/cordova_sites_ah/cleft_vd). The process to create a new site is described as follows.

Beginning on a server with PHP [5], MySQL [6], Ruby [30], Bootstrap [24], Apache [31] and Kafeen [2] installed, the Cordova software is cloned from the GitHub repository (https://github.com/clcg/cordova). Following the Cordova installation document on GitHub, the site is instantiated and configured according to documentation. This includes creating and editing three configuration files on the server. The site title is edited, the site's URL is specified, a MySQL user and password is specified, an administrator email is entered and an empty database is created and the name is specified. Now, the Cordova instance is available through a web browser. Through the release changes interface in Cordova, migrations are used to build the database tables and create an initial release. From here, genetic data can be entered. Using the upload genes interface, a list of genes is entered. Upon completion, the system sends the administrator an email notification. During the collection process, expert curations can be uploaded through the expert

curations interface from a csv file. When variant collection is complete and the variants are staged in the queue table, expert curations can be applied with the click of a button through the expert curations interface. The disease nomenclature can be updated using the normalize nomenclature interface by downloading the current nomenclature file, editing and uploading the preferred nomenclature or editing the nomenclature through the form on the interface. In the review changes interface, the user can download a "difference" summary prior to release of the data. This file summarizes what variant pathogenicity updates will occur and what variants of pathogenic or likely pathogenic status will be dropped or added from the variations table. All of the variants are confirmed for release and released to the variations table, any variations that are updated are added to the variations log.

This process is repeated to instantiate the other variation databases. Figure 39 describes an actual timeline for setting up the Head and Neck Cancer Variation Database with 19 genes resulting in 75,319 collected and annotated variants. Cordova Tables 2-5 describes the genes representing the disease-specific instances that were instantiated with the Variant-CADI software.

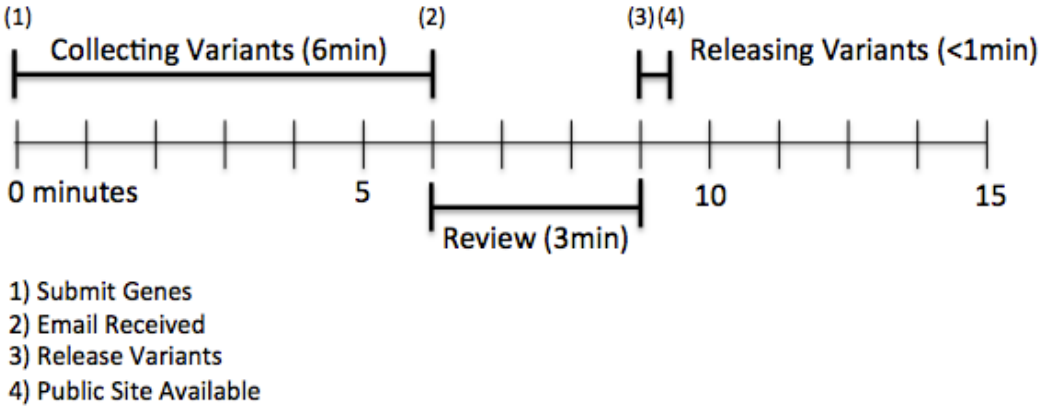


Figure 39. Cancer Variation Database Initiation Timeline

With the Cordova instance set up on the server, it took less than ten minutes (wall clock time) to collect and release over 75,000 variations. The time spent collecting and annotating the variations took no more than six minutes. The email feedback showed that there were no errors. Visiting the release changes interfaces and releasing all changes populated the live site in less than one minute.

Table 2. Head and Neck Cancer Genes
(Genes: 19 - Variants: 75,319 - Time: 12 minutes)

Gene	Number of Variants	Gene	Number of Variants	Gene	Number of Variants
AJUBA	639	HRAS	442	PTEN	4563
CASP8	2286	KMT2D	4645	RB1	7456
CDKN2A	1580	NFE2L2	1503	TGFBR2	3868
CUL3	4827	NOTCH1	4645	TP53	1528
FAT1	8376	NSD1	7616	TRAF3	5625
FBXW7	7279	PIK3CA	3454		
HLA-A	1361	PIK3R1	3626		

Table 3. Renal Genes
(Genes: 8 - Variants: 18,710 - Time: 10 minutes)

Gene	Number of Variants	Gene	Number of Variants
CFH	4948	THBD	429
CFI	3108	ADAMTS13	3105
CFB	782	PLG	2864
CFHR5	1829	DGKE	1645

Table 4. Cleft Genes
(Genes: 1 – Variants: 3,110 – Time: 5 minutes)

Gene	Number of Variants
FGFR1	3110

Table 5. Vision Genes
(Genes: 280 - Variants: 1,799,784 - Time: 2.5 hours)

Gene	Number of Variants	Gene	Number of Variants	Gene	Number of Variants	Gene	Number of Variants
A2M	2704	BBS7	1971	CFHR5	1829	F12	845
A2M-AS1	143	BBS9	18114	CFI	3108	F2	1296
ACE	2363	BMP4	454	CFP	297	F3	660
ACTN4	4135	BSND	661	CHD1L	2924	F5	4213
ADAMTS13	3105	C12orf29	660	CLCN5	2690	F7	1374
ADCK4	1444	C1QA	241	CLCNKA	1406	F8	3041
ADM	252	C1QB	448	CLDN16	1281	KLHL3	4128
ADM2	409	C1QC	359	CLDN19	470	KLHL41	822
AGT	946	C1R	642	CLU	1150	KLKB1	1791
AGTR1	1784	C1S	875	CNNM2	6028	LAMB2	1735
AGTR2	191	C2	754061	COA3	137	LMNB2	2068
AGXT	1091	C3AR1	511	COL4A1	8568	LMX1B	3783
AHI1	8623	C4BPA	1733	COL4A3	7494	LOC654841	435
ALG1	1518	C4BPB	598	COL4A4	7818	LYZ	343
ALMS1	11105	C4B_2	1627	COL4A5	5060	MAP3K5	9131
ANLN	3048	C5AR1	775	COLEC11	2932	MASP1	3920
APCS	215	C5AR2	621	COQ2	997	MASP2	1653
APOA1	1101	C6	5624	CPN1	1646	MBL2	455
APOL1	1140	C7	3787	CR1	5230	MBTPS1	4288
APRT	425	C8A	3037	CR2	1946	MKKS	1470
AQP2	627	C8B	1971	CRB2	1975	MKS1	1034
ARHGAP24	19900	C8G	433	CRCP	1828	MUC1	571
ARHGDIA	452	C9	3818	CREBBP	8734	MYH9	6297
ARL13B	3258	CACNA2D1	19562	CRP	270	MYO1E	12100
ARL4D	157	CASR	4639	CSPP1	5186	NAT8	269
ARL6	1346	CC2D2A	5340	CTNS	1829	NEK1	8661
ATP6V0A4	4812	CCDC146	7091	CTXN2	330	NEK8	1027
ATP6V1B1	1663	CD2AP	5920	CUL3	4827	NEK9	2125
ATP7B	4717	CD46	1856	DGKE	1645	NLRP3	2296
ATXN10	7189	CD55	1669	DHCR7	1165	NOTCH2	6494
AVPR2	582	CD59	1322	DMP1	779	NPHP1	3241
AXDND1	8846	CEP290	4440	DSTYK	2987	NPHP3	13456
B2M	296	CEP41	2066	DTX2	1874	NPHP4	7207
B9D1	1636	CFB	782	DYNC2H1	16489	NPHS1	2349
B9D2	539	CFD	431	EGF	4500	NPHS2	982
BBS1	1457	CFH	4948	EHHADH	3732	NR3C2	14330
BBS10	556	CFHR1	768	ENPP1	4111	OCRL	1348
BBS12	733	CFHR2	929	EYA1	6653	OFD1	1256
BBS2	1779	CFHR3	1017	F10	1605	PAX2	3399
BBS4	2477	CFHR4	1797	F11	11271	PDSS2	10919

Table 5. Continued

PHB	562	ROBO2	25252	SLC26A4-AS	298	TRPM6	7486
PHEX	4462	RPGRIP1L	4520	SLC34A1	1202	TSC2	5681
PI4KA	8341	RPL36A-	871	SLC34A3	1020	TTC21B	4082
PKD1	7859	SALL1	1250	SLC3A1	2972	TTC8	2161
PKD2	3468	SALL4	1290	SLC4A1	1543	UMOD	1407
PKHD1	20512	SCARB2	2389	SLC4A4	14439	UPK3A	798
PLAT	1904	SCNN1A	1941	SLC5A2	1033	VEPH1	9121
PLAU	614	SCNN1B	3906	SLC7A9	2233	VKORC1L1	3291
PLCE1	12697	SCNN1G	1748	SMARCAL1	3259	VSIG4	665
PLCG2	12901	SERPINA1	1091	SOX17	326	VTN	504
PLG	2864	SERPINA5	879	STAG3L3	116	VWF	9984
PLGLB2	20532	SERPINC1	1007	TCTN1	1664	WDR19	4296
PMM2	3191	SERPINE1	929	TCTN2	2046	WDR35	3612
PMS2P5	73509	SERPINF2	974	THBD	429	WDR73	781
PREPL	2632	SERPING1	1329	TMEM138	411	WNK1	7277
PRKD2	2235	SIX1	339	TMEM216	306	WNK4	1523
PROC	1061	SIX2	357	TMEM237	1219	WNT4	1195
PROS1	4452	SIX5	613	TMEM67	3187	WT1	3538
PTPRO	11330	SLC12A1	4320	TRAP1	4289	XPNPEP3	3144
REN	851	SLC12A3	3613	TRIM32	741	ZMPSTE24	1619
RET	3199	SLC26A4	3101	TRPC6	6136		

CHAPTER 6 – FUTURE ENHANCEMENTS

Consideration of the software implemented and described in this thesis identifies potential future enhancements. Including these features would further improve the usability of Cordova and Variant-CADI. These features include the following. Added search functionality on each of the interfaces would reduce time spent searching for variants in the site or relying on web-browser search functionality. To further enhance this feature, search and filter functionality could be implemented. This would potentially allow users to filter variants by genomic regions, pathogenicity predictions, coding region, and other genomic features. A second feature would be to shift from a simple user input in normalize nomenclature to a model more closely representing expert curations. This would allow users to maintain their list of disease phenotype preferences by gene in the Cordova database and apply the preferences as desired. To enhance expert curations, interfaces to view and edit expert curations should be implemented to avoid the need for downloading/uploading data to edit curations. When modification of expert curations is required, this would keep all data within Cordova interfaces and databases. These interfaces may resemble many of the other interfaces already in place to edit variations but would need specific adjustments to fit the differing attributes of the data. Currently, summary "differential" statistics are provided to the user in a flat file. An additional interface to display this data would be desirable. Kafeen is currently configured for use with a defined set of data sources. Creating a Kafeen tool to specify the data sources configurations would allow for integration into Cordova and allow users to choose data sources and what data to select from each data source. This would expand use cases of

these tools. In addition, incorporating the download and preprocessing of Kafeen input could be integrated into the Cordova system through an interface.

REFERENCES

1. Ephraim S. S. Coordinated Laboratory for Computation Genomics. Cordova. Available at https://github.com/clcg/cordova

2. Ephraim S. S. Coordinated Laboratory for Computation Genomics. Kafeen version 2.0. Available at https://github.com/clcg/kafeen

3. Ephraim, S. S. (2014). Design and application of methods for curating genetic variation databases.

4. Ephraim, S. S., Anand, N., DeLuca, A. P., Taylor, K. R., Kolbe, D. L., Simpson, A. C., ... & Casavant, T. L. (2014). Cordova: Web-based management of genetic variation data. *Bioinformatics*, *30*(23), 3438-3439.

5. PHP Development Team. PHP Language Manual, version 5.5.0. Available at http://php.net/manual/en/langref.php

6. Sun Microsystems. MySQL Documentation, version 5.0.95. Available at http://dev.mysql.com/doc/

7. British Columbia Institute of Technology. CodeIgniter User Guide, version 3.1.2. Available at http://www.codeigniter.com/user_guide/

8. Cotton, R. G. H., & Horaitis, O. Human Genome Variation Society. *eLS*.

9. Sherry, S. T., Ward, M. H., Kholodov, M., Baker, J., Phan, L., Smigielski, E. M., & Sirotkin, K. (2001). dbSNP: the NCBI database of genetic variation. *Nucleic acids research*, *29*(1), 308-311.

10. Ng, P. C., & Henikoff, S. (2003). SIFT: Predicting amino acid changes that affect protein function. *Nucleic acids research*, *31*(13), 3812-3814.

11. Harvard Division of Genetics. PolyPhen version 2. Available at http://genetics.bwh.harvard.edu/pph2/

12. Chun,S. and Fay,J.C. (2009) Identification of deleterious mutations within three human genomes. Genome Res., 19, 1553–1561.

13. Schwarz, J. M., Rödelsperger, C., Schuelke, M., & Seelow, D. (2010). MutationTaster evaluates disease-causing potential of sequence alterations. *Nature methods*, *7*(8), 575-576.

14. University if California Santa Cruz. PhyloP. Available at http://hgdownload.cse.ucsc.edu/goldenPath/mm10/phyloP60way/

15. Sidow Labs. GERP++. Available at http://mendel.stanford.edu/SidowLab/downloads/gerp/

16. Shearer, A. E., Eppsteiner, R. W., Booth, K. T., Ephraim, S. S., Gurrola, J., Simpson, A., ... & Happe, S. (2014). Utilizing ethnic-specific differences in minor allele frequency to recategorize reported pathogenic deafness variants. *The American Journal of Human Genetics*, *95*(4), 445-453.

17. NHLBI GO Exome Sequencing Project. (2013). Exome variant server. *Seattle, WA: http://evs. gs. washington. edu/EVS*.

18. Siva, N. (2008). 1000 Genomes project. *Nature biotechnology*, *26*(3), 256-256.

19. Exome Aggregation Consortium. (2015). ExAC Browser (Beta): Exome Aggregation Consortium. *Cambridge, MA: Exome Aggregation Consortium (ExAC). Available at: http://exac. broadinstitute. org [Nov, 2014]*.

20. Landrum, M. J., Lee, J. M., Riley, G. R., Jang, W., Rubinstein, W. S., Church, D. M., & Maglott, D. R. (2014). ClinVar: public archive of relationships among sequence variation and human phenotype. *Nucleic acids research*, *42*(D1), D980-D985.

21. Liu, X., Jian, X., & Boerwinkle, E. (2013). dbNSFP v2. 0: a database of human non-synonymous SNVs and their functional predictions and annotations. *Human mutation*, *34*(9), E2393-E2402.

22. Cooper, D. N., Ball, E. V., Stenson, P. D., Phillips, A. D., Howells, K., Mort, M. E., & Thomas, N. S. T. (2013). The Human Gene Mutation Database (HGMD®).

23. Samtools. bcftools. Available at https://samtools.github.io/bcftools/bcftools.html

24. McLaren, W., Pritchard, B., Rios, D., Chen, Y., Flicek, P., & Cunningham, F. (2010). Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor. *Bioinformatics*, *26*(16), 2069-2070.

25. Twitter Bootstrap Team. Bootstrap version 2.0.Available at http://getbootstrap.com/getting-started/

26. Smith, Richard J. Renal Genetic Variations [Interview]. (2015, September 23).

27. Cancer Genome Atlas Network. (2015). Comprehensive genomic characterization of head and neck squamous cell carcinomas. *Nature*, *517*(7536), 576-582.

28. Stone, Edwin M. Vision Genetic Variations [Interview]. (2015, September 23).

29. Schnieders, Michael. Cleft Genetic Variations [Interview]. (2016, September 15).

30. Matsumoto, Yukihiro. Ruby Documentation version 2.1.5. Available at https://www.ruby-lang.org/en/documentation/

31. The Apache Software Foundation. Apache User's Guide, version 2.2.0. Available at http://directory.apache.org/studio/users-guide.html