**2024 Capstone Project: Goals, Ingredients, and Outline**

**Goal:** The goal of this project is to allow researchers or clinicians to understand the importance of rare variants discovered in patients affected by a particular disease. More specifically, it is to provide a more granular classification to variants of uncertain significance identified in their patients affected by a single disease or one that falls in a class of related diseases. In our current paper we used the groundwork of Pejever et al. to do this for Congenital central hypoventilation syndrome (CCHS) for which PHOX2B is the only associated gene. The goal of this project is to create an R package or RShiny module that can be used by any researcher who has a set of genes for which they would like to understand what level of evidence from in silico annotation would be necessary to classify a VUS into different pathogenicity/benignity classes.

**Ingredients**

1. Ability to define exonic positions for a gene (mostly done)
2. Tool to get variants in these regions from:
   1. GnomAD
   2. REVEL
   3. CADD
   4. BayesDel
   5. AlphaMissense
   6. ClinVar (other? HGMD?)
3. A method and code to classify variants as benign / vus / pathogenic
4. Code to derive the annotation values corresponding to the pathogenicity/benignity support levels. (Mostly done)
5. Code that shows the distribution of annotation scores across the gene(s) (Mostly done)
6. Code to estimate the AUC and other relevant metrics (Mostly done)
7. Code that would allow a user to enter a list of variants and would return the confidence level for each variant being benign/pathogenic (and maybe put it into the context of the training set).

**Outline**

**Task 1:** Spitball to think about what the final product will look like.

**Task 2:** Create function and object that will determine and contain the exonic positions of interest with necessary labels (genes, exons, others?)

**Task 3:** Create functions / objects to collect annotation information for the list of information in ingredient 2. Region information will be provided as input.

***Subtask*:** Some of these resources are dynamic (change over time). Think about plan to update, preferably in an automated way.

**Task 4:** Create code and object to use ClinVar and GnomAD information to define a variant as benign, uncertain (vus), or pathogenic.

***Subtask***: It is likely that if insufficient information is available (e.g. not enough benign and pathogenic variants) then the prediction model built will not be powerful enough to set interesting levels of support. How should this decision be made?

**Task 5:** Write code to define thresholds for support and to communicate information about the model, including:

1. Summarize the data used as input:
   1. Total number of variants
   2. Distribution of variants across genes.
   3. Distribution of B/U/P across genes
   4. Distribution of scores across genes/exons
2. Predict the level of support for all B/U/P variants and:
   1. Summarize concordance of B/P across genes
   2. Distribution of classifications across genes and exons w/i genes
   3. Predict hotspots (how?)
   4. Calculate AUC across all and within genes.
3. Return individual level output for each variant including:
   1. GnomadAF, annotation scores, predicted pathogenicity, clinvar info, P/U/B classification, prediction classification.
4. Other ideas?

**Task 6:** Create tools to allow user to predict the classification of new VUSs based on the results of our analysis package. How to do this?

1. Automate the creation of an Excel table
2. Create an RShiny dashboard that would allow them to enter the list of genes.
   1. Model parameters to determine the level of evidence for P/B could be entered by the user or we could keep a database of parameters on the app from previous analyses.
3. If analyses are performed on the app, then we could keep the gene list / disease fix and then examine clinvar for new info and perform the analysis again. Thus should think about version control and easy automation for updates.

**07/15/2024**

**Accessing *in silico* and variant population and clinical information**

**In silico annotation**

* 1. **REVEL**: <https://sites.google.com/site/revelgenomics/downloads> (<https://www.google.com/url?q=https%3A%2F%2Frothsj06.dmz.hpc.mssm.edu%2Frevel-v1.3_all_chromosomes.zip&sa=D&sntz=1&usg=AOvVaw2DS2TWUYl__0vqijzzxp5M>)

There are also files with 500k variants per file. That might be preferrable if they are organized by position.

* 1. **CADD**: <https://cadd.gs.washington.edu/download> (SNVs only <https://krishna.gs.washington.edu/download/CADD/v1.7/GRCh38/whole_genome_SNVs.tsv.gz>, <https://krishna.gs.washington.edu/download/CADD/v1.7/GRCh38/whole_genome_SNVs.tsv.gz.tbi>)
  2. **BayesDel**: <https://fenglab.chpc.utah.edu/BayesDel.html> (<https://drive.google.com/drive/folders/1K4LI6ZSsUGBhHoChUtegC8bgCt7hbQlA?usp=sharing>)

Note these variants’ coordinates are hg19. They will need to be converted to hg38. This is done via a program called LiftOver (there is likely an R/Bioconductor package that can do it)

* 1. **AlphaMissense**: <https://github.com/google-deepmind/alphamissense> (<https://console.cloud.google.com/storage/browser/dm_alphamissense;tab=objects?prefix=&forceOnObjectsSortingFiltering=false> (https://console.cloud.google.com/storage/browser/\_details/dm\_alphamissense/AlphaMissense\_hg38.tsv.gz)

Population frequency information

**GnomAD**: <https://gnomad.broadinstitute.org/downloads#v4> (download Joint Frequencies)

Clinical information

**Clinvar:** <https://ftp.ncbi.nlm.nih.gov/pub/clinvar/vcf_GRCh38/> (<https://ftp.ncbi.nlm.nih.gov/pub/clinvar/vcf_GRCh38/clinvar.vcf.gz> (and the tbi file)

**Thoughts**: Check out VEP and ensemblVEP R/bioconductor package. See how much of what is above we can get from this.

Try for this week:

1. Explore ClinVar. Look at what information is collected for variants. Think about what would be useful for making a definitive decision for calling it pathogenic/likely path/vus/likely benign/benign.
2. Check out VEP. See if we can use VEP for some or all of these annotation. Note that the value that we want for GnomAD is faf95max (read about what that is). I’m almost positive that we can’t get that from VEP, but maybe I’m wrong.
3. Choose an annotation, download the data (or a chromosomes worth if applicable). Write a function that will extract out all variants for a user-specified region (e.g. chr1:5000000-6000000).