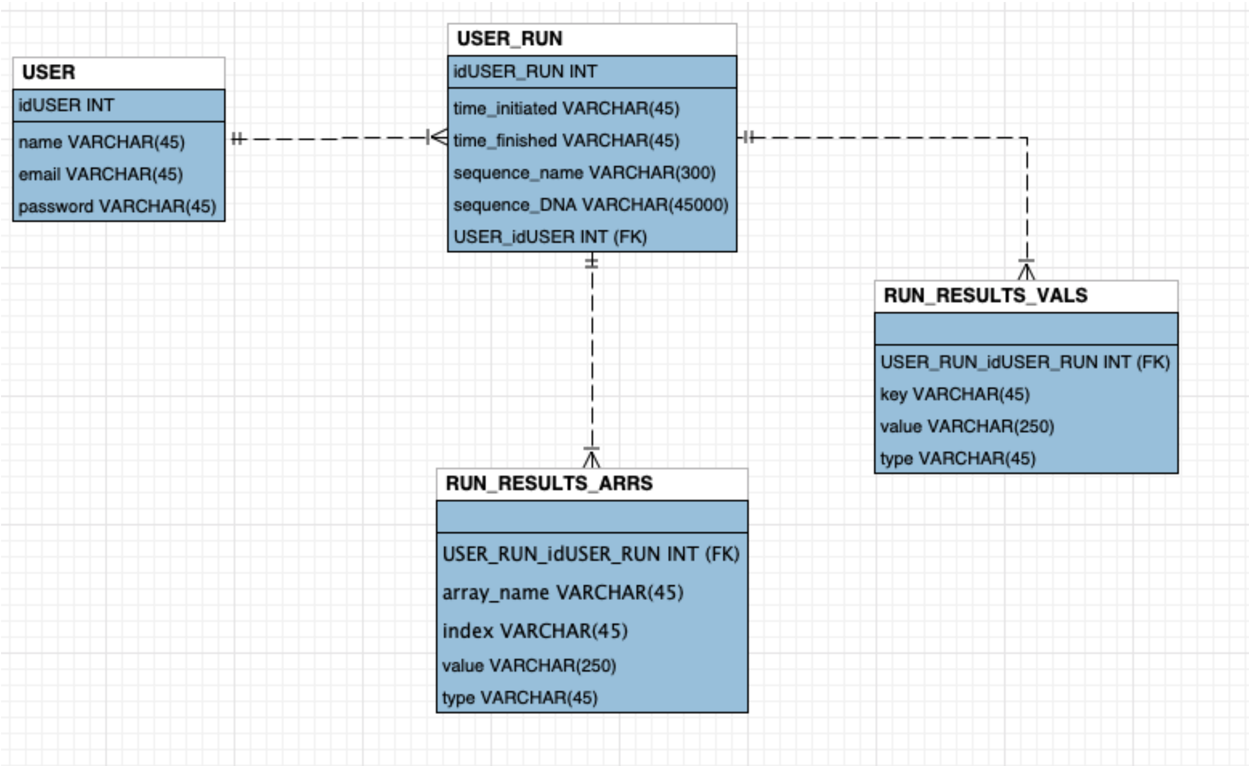
410.712.81
Final Project Proposal

Mauro Chavez
mchavez8@jh.edu

Implementing a front end for VADR

Classifying viral sequences has never been a more critical task, but luckily the process is made simpler by some fantastic web apps such as Nextclade (https://clades.nextstrain.org/) and Pangolin (https://pangolin.cog-uk.io/). Both of these exhibit a clean and intuitive user interface for the submission and analysis of viral sequences. The rapid analysis and annotation of SARS-CoV-2 genomes to classify emerging variants is extremely important to surveillance of the virus itself. That being said, one important tool is missing from the webapp space. VADR (Viral Annotation DefineR) is a software package for classifying and analyzing viral sequences against sets of reference genomes [Schäffer 2019]. This tool is important because it serves as an institutional check that determines whether or not a submitted viral sequence is accepted into the GenBank database. For my final project, I propose developing a web app wrapper around VADR such that researchers have a quick place to go to see if their SARS-CoV-2 sequences pass this check in the GenBank submission process. As sequences are submitted to this database, the results of analysis would be tracked in a relational schema for the collection of metadata. The schema would support tracking input sequences, analysis runs, and results of analysis. This would enable queries to understand patterns in the data. For example, what are the most common annotations of the sequences being submitted, or what are the most common errors that keep sequences from passing VADR validation? I propose the simple schema below that is flexible to different types of results pulled from the various VADR output files….



I am not yet extremely familiar with the output that VADR performs but this structure is very flexible and enables a pretty generic interface for data presentation in the front end. One part of the view will display results in the form of (Key, Value) pairs from records in the

RUN_RESULTS_VALS table and another will display lists of data from the RUN_RESULTS_ARRS where values are grouped under lists of array_name.

To keep things simple, users will navigate to a web page where they will be able to create an account. This will be useful for keeping track of who owns each analysis job. In the case that the analysis itself takes a while to complete, having a user's email on file can be helpful in notifying them once the job is finished. There will be a job submission page where they can paste in a viral sequence and give it a name. From there, they can click submit to start a VADR run. I would like to containerize VADR within docker so that I can make sure to have a stable version that I know runs**. Is it possible to get docker privileges on our server and am I allowed to mount the filesystem to my docker containers?** If not, I will need VADR to be installed for me and I'll need the appropriate permissions for running the package (https://github.com/ncbi/vadr). Once the analysis of the single sequence has finished, a script will parse out the results and save them into the relational schema described above. Users will have a view of all their submitted sequences where ones that have completed analysis will be linked to pages that display the VADR results. I plan to write the entire backend for this application in Python. NCBI hosts a database of SARS-CoV2 sequences so I'll have lots of test data to validate the code.

The first phase in this project would be to get VADR running on the server and make sure that the analysis can be completed in a reasonable amount of time. From there, I'd develop the system for account registration and job tracking. Finally, I'd implement the front-end triggering jobs on the server and write the code for populating the database along with presenting that data in the UI.

If all this goes smoothly, I'd implement a summary page that displays metadata about the analysis jobs people are submitting so that users can get a sense of what types of issues researchers are running into when attempting to get their sequences validated.

As a backup project proposal, it might be cool to implement a service that takes a VCF resulting from variant calling of a COVID sample and split it into two FASTA files. Research has shown that co-infection by two different SARS-CoV-2 strains at once is possible [Da Silva 2021]. This service would group variants by their allele frequency in a VCF file and return two sequences, one containing variants with higher allele frequency and one containing variants with lower allele frequency. This way someone could take the returned fasta file and feed it into programs like Nextclade or Pangolin to see if either sequence is a legitimate COVID strain. While I know where to find test data for my first idea involving VADR, I am not sure where I'd find the VCF files needed for testing this second idea.

Sources:

Da Silva Francisco, Ronaldo, et al. "Pervasive Transmission of E484K and Emergence of VUI-NP13L with Evidence of SARS-CoV-2 Co-Infection Events by Two Different Lineages in Rio Grande Do Sul, Brazil." 2021, doi:10.1101/2021.01.21.21249764.

Schäffer, Alejandro A, et al. "VADR: Validation and Annotation of Virus Sequence Submissions to GenBank." 2019, doi:10.1101/852657.