SARS-CoV-2 Early Detection Pseudocode/Algorithms

The following details the algorithms of the entire pipeline to score and rank variant constellations or single amino acid mutations.

1.) Analysis from FASTA data

Input: Wuhan-Hu-1 reference in FASTA, last four months of SARS-CoV-2 NT sequences in FASTA, an optional protein

Output: List of variant constellations in form “[protein]_H10K,[protein]_D100G, …” scored and ranked by the *Composite Score* or single amino acid mutations ranked by *Mutation Prevalence Score*

Step 1: Quality Control

a. Iterate through all sequences in the FASTA and filter out sequences with high ambiguous nucleotide content (default >0.01), missing or incorrect viral species name, low sequence length coverage (default <29400 nucleotides), missing region, or missing date

Step 2: Pairwise Alignment to Wuhan-Hu-1 Reference Genome

For each SARS-CoV-2 NT sequence …

   a. Pairwise align to Wuhan-Hu-1 reference genome
   b. Parse the alignment cigar string to extract positions of all insertions and deletions in the query sequence and store appropriately
   c. Using the annotated reference sequence gene start and end coordinates (Nextstrain/ViPR) and the positions of insertions and deletions on the query sequence, compute the gene start and end coordinates relative to the query
   d. Translate each gene start and end on the query into the gene polypeptide chain
   e. Abort current sequence and skip to next if a translated query gene is frameshifted (frameshift → ((len(ref polypeptide) – len(query polypeptide)) mod 3) != 0)

   For each query gene polypeptide chain with a mutation …

a. Pairwise align the query gene polypeptide chain to the corresponding reference gene polypeptide chain

b. Parse the alignment cigar string to extract positions of substitutions, insertions, and deletions

c. Covert ORF1a/ORF1ab positions to non-structural protein position

d. Concatenate the protein name, reference amino acid, mutation position, and alternate amino acid as [PROTEIN]_[REF][POS][ALT]

e. Concatenate each mutation into one variant constellation, either for a single protein or the entire proteome

Step 3: Store Variant Counts Across Space and Time

For each computed variant constellation

a. Parse out the collection date and collection region from the original record

b. Store a hash table of variant-to-region-to-date counts

c. Store a hash table of region-to-date sequence isolate counts

d. Parse out each single mutation in the constellation, store in a hash table of mutation-to-region-to-date counts

Step 4: Compute the Variant Dynamics

a. Convert the hash tables into a data frame, with columns for variants, regions, dates, and counts

b. Bin the dates into a period (week/2-week/month) and aggregate the counts per region (or globally if user specified)

c. Use the region-to-date sequence isolates counts to compute a variant prevalence ratio per period per region (region variant count in period p / regional sequence isolates count in period p)

d. Use the region-period prevalence ratios to compute a region-period growth rate per variant (regional prevalence ratio in period p / regional prevalence ratio in period p – 1)

e. OPTIONAL: Use the region-period growth rates to compute a region-period jerk rate per variant (regional growth rate in period p – regional growth rate in period p-1)

f. End up with an analyzable data frame in the following format, stored as a pickle in the data directory of the pipeline

Variant	**Region**	**Date**	**Variant Count**	**Isolates Count**	**Prevalence**	**Growth**	**Jerk**
X	X	YYYY-MM-DD	X	X	X	X	X

Step 5: Score Variants Based on Dynamics and/or Predicted Functional Impact

a. Filter out variants with a variant count less than 10 in the most recent period
b. Exclude all data beyond the most recent three periods (default three months)

For each variant …

a. Count the times the variant has prevalence ratio greater than 0.05 or growth rate greater than 5 in any region-period combination → *Sequence Prevalence Score/Mutation Prevalence Score*

Only for scoring a variant constellation …

b. Count the overlap with *Sequence Features of Concern*; i.e., Spike regions shown to experimentally impact class 1, 2, 3, or 4 mAbs neutralization, convalescent sera/Moderna Abs neutralization, ACE2 binding, overlap with the NTD supersite, overlap with other critical Spike sequence feature regions such as the Furin Cleavage site, or any non-Spike drug resistant sites/active sites/mutatgenesis sites → *Functional Impact Score*
c. Sum the *Sequence Prevalence Score* with the *Functional Impact Score* → *Composite Score*

Step 6: Rank Variants by Composite Score/Mutation Prevalence Score

a. Sort the variants by their *Composite Score/Mutation Prevalence Score*
b. Save the results to a file stored in results directory

2.) Analysis from GISAID Metadata

Input: GISAID metadata file, an optional protein

Output: List of variant constellations in form “[protein]_H10K,[protein]_D100G, …” scored and ranked by the *Composite Score* or single amino acid mutations ranked by *Mutation Prevalence Score* or PANGO Lineages ranked by *Emerging Lineage Score*

Same as Step 3 – Step 6 in the previous example. For PANGO Lineage *Emerging Lineage Score*, parse out PANGO Lineage from GISAID metadata instead of the variant constellation.