Antiviral Strategies Against SARS-CoV-2 - a Systems Biology Approach

Erica T. Prates[1,2+], Michael R. Garvin[1,2+], Piet Jones[3], J. Izaak Miller[1,2], Kyle A. Sullivan[1,2], Ashley Cliff[3], Joao Gabriel Felipe Machado Gazolla[1,2], Manesh Shah[2],[8], Angelica M. Walker[3], Matthew Lane[3], Christopher T. Rentsch[5,6], Amy Justice[6,7], Mirko Pavicic[1,2], Jonathon Romero[3], Daniel Jacobson[1,2,3,4,8*.]

Affiliations:

[1]Oak Ridge National Laboratory, Computational Systems Biology, Oak Ridge, TN; [2]National Virtual Biotechnology Laboratory, US Department of Energy; [3]The Bredesen Center for Interdisciplinary Research and Graduate Education, University of Tennessee Knoxville, Knoxville, TN; [4]University of Tennessee Knoxville, Department of Psychology, NeuroNet Research Center, Knoxville, TN; [5]Faculty of Epidemiology & Population Health, London School of Hygiene and Tropical Medicine, London, UK, WC1E 7HT; [6]VA Connecticut Healthcare/General Internal Medicine, West Haven, CT, 06516; [7]Yale University School of Medicine; [8]Genome Science and Technology, University of Tennessee Knoxville, Knoxville, TN.

***Correspondence: jacobsonda@ornl.gov**

[+]**Contributed equally**

Abstract

The unprecedented scientific achievements in combating the COVID-19 pandemic reflect a global response informed by unprecedented access to data. We now have the ability to rapidly generate a diversity of information on an emerging pathogen and, by using high-performance computing and a systems biology approach, we can mine this wealth of information to understand the complexities of viral pathogenesis and contagion like never before. These efforts will aid in the development of vaccines, antiviral medications, and inform policy makers and clinicians. Here we detail computational protocols developed as SARS-CoV-2 began to spread across the globe. They include pathogen detection, comparative structural proteomics, evolutionary adaptation analysis via network and artificial intelligence methodologies, and multi-omic integration. These protocols constitute a core framework on which to build a systems-level infrastructure that can be quickly brought to bear on future pathogens before they evolve into pandemic proportions.

Key words: antiviral, SARS-CoV-2, COVID-19, Systems Biology, multi-omics, pandemic.

***Running Head**:* Antiviral Strategies from Systems Biology

1. Introduction

COVID-19 has surpassed 93 million cases, leading to the death of nearly 2 million people worldwide. While it took a century to gain a clear mechanistic understanding of the H1N1 virus and the 1918 pandemic, the global scientific community has produced a stunning picture of the SARS-CoV-2 and COVID-19 in just under a year. As a result, vaccines are being deployed worldwide, new therapeutics are being developed, and FDA-approved pharmaceuticals are being repurposed. Arguably the most critical factor in the successes we have achieved to-date has been the open flow of data and scientific insights, which is nearly equal to the scale and distribution of the virus itself. The challenge with this pandemic and future pathogen outbreaks is no longer generating data, but rather extracting and integrating virus-centric, host-centric, and virus-host interaction insights to understand all aspects of viral pathogenesis, i.e., from a systems level. Let us learn from this experience so that we can prevent the next contagious disease from becoming an epidemic, let alonge, of pandemic proportions..

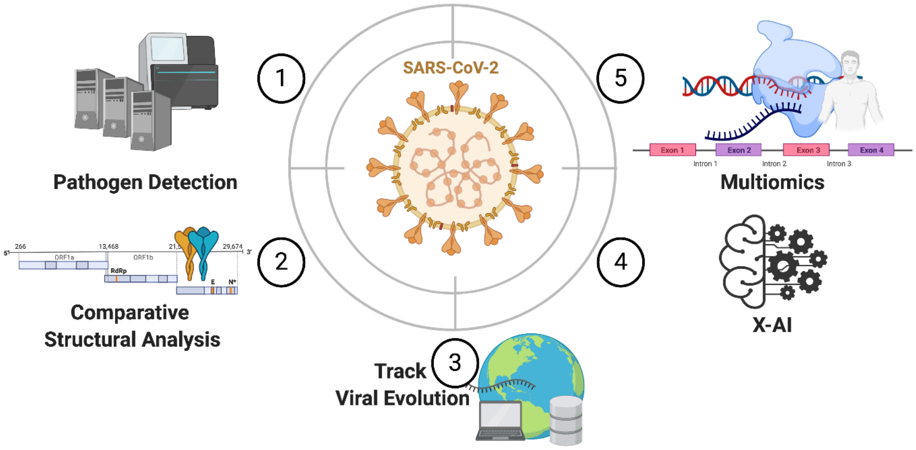


Figure 1. Systems Biology strategies against SARS-CoV-2. (1) The use of HPC for pathogen detection from non-host RNA-Seq and identification of co-occurring microbial communities; **(2)** Proteome-wide structural comparison of SARS-CoV-2 with evolutionary-related species to identify potential molecular features determining pathogenicity and ideal targets for broad-spectrum antivirals; **(3)** Tracking SARS-CoV-2 evolution; **(4)** Explainable artificial intelligence predictive models (X-AI) such as iterative Random Forest Leave One Out Prediction (iRF-LOOP) and Random Intersection Trees (RIT) to determine feature importance and interaction for various applications such as identifying which mutations may be coevolving; **(5)** Understanding COVID-19 pathogenesis using multi-omics integration and using disease and biological function gene ontologies (GOs).

Computational Systems Biology is a rapidly advancing field that leverages high-performance computing (HPC), machine learning (ML), and artificial intelligence (AI) algorithms to mine complex and diverse data sets to extract critical networks of information. A crucial first step in this system-level approach is to assemble a multidisciplinary team and provide an infrastructure and common language for rapid and efficient communication among members. Here we describe a system-level antiviral strategy built by a team with expertise in pathogen detection, molecular evolution, structural biology, AI algorithm development, multi-omics integration, and clinical medicine. Herein we detail five methods (Fig. 1) to be employed **synergistically** to collect, analyze, and report a diversity of information underlying SARS-CoV-2 pathogenesis towards establishing promising strategies to restrain viral spread. Ultimately, we devised this material to be applicable against the next potential outbreak. Future efforts will likely increase the use of AI and ML approaches to provide more accurate predictive models in shorter timescales as pathogens emerge in the human population.

1.1 Pathogen detection

Identification of the pathogen from early clinical samples is crucial for limiting the expansion of the virus. Furthermore, the detection of opportunistic bacterial and fungal pathogens and their impact on the host-microbiome are important factors that can influence treatments and patient outcomes. Here, taxonomic identification was performed on non-host RNASeq using a parallelized version of Kraken2, ParaKraken ***(1)*** using HPC facilities to cover an extensive range of potential taxa and analyze a large set of samples. Identification of a broad range of taxa at large scales may allow for the detection of emergent, potentially novel pathogens and assessment of putative zoonotic events. For individuals with limited access to HPC resources, Kraken2 can be used with a more focused range of taxa on fewer samples. After taxonomic classification is performed, subsequent analyses are needed to mitigate potential false positives, identify potential microbial dysbiosis, and determine the potential enrichment of putative pathogens. Furthermore, RNA transcripts assigned to a particular taxon can then be extracted and assembled for additional analyses if there is sufficient read depth. These assembly-based analyses may be informative for understanding viral evolution (in the case of a viral-identified taxa), or improve classification.In Fig. 2, we give an overview of the process, which is detailed in **Section 3.1**.

A community analysis of the microbial taxa is important to determine the potential for dysbiosis. Pathogenicity of viruses, and viral-associated disease outcomes are strongly influenced by microbial dysbiosis. Therefore understanding the community structure allows for the potential to influence treatment, and improve patient outcomes.

1.2 Proteome-wide structural comparison of SARS-CoV-2 with evolutionary-related species

In the scenario of a next pandemic, it is probable that little information will be available for the emerging pathogen, as was the case with SARS-CoV-2. However, the breadth of available genome-level information for millions of species, including viruses, is expanding rapidly. Comparative proteome and genome analyses can provide rapid insights into the biology of a pathogen as it spreads. For example, similarities with the closely related SARS-CoV coronavirus quickly established the host protein ACE2 as the receptor for the virus and the intricacies of how the spike protein binds to it and allows the virus to enter into the cell ***(2)***. As shown in our published reports, the comparative analyses of proteins from SARS-CoV-2 and evolutionary-related species can be a valuable approach to quickly establish potential therapies and a mechanistic understanding of the effects of the virus on the human immune system.

In Prates et al. ***(3)***, by applying the method described here, we expand the usual focus from the spike glycoprotein, and suggest that molecular differences between SARS-CoV and SARS-CoV-2 proteomes in other regions, such as in the nsp1 and nsp3 proteins, likely have a significant contribution due to their distinct pathogenic profiles. On the other hand, based on the comparison with another coronavirus, the porcine epidemic diarrhea virus (PEDV), we suggest that the highly conserved binding site of the SARS-CoV-2 main protease may be able to bind and cleave the NF-kB essential modulator ***(4)***, possibly resulting in an additional mechanism of circumventing the activation of the host immune response by NF-kB signaling – a hypothesis that has been recently associated with microvascular brain pathology in a preprint manuscript ***(5)***. Moreover, whereas exploring mutations can shed light on the mechanistic causes for pathogenicity, such conserved functional regions may be promising targets for developing broad-spectrum antivirals. Additionally, in Garvin et al. ***(6)***, structural proteomics was applied in synergy with median-joining network (MJN) analysis (**Sections 1.3** and **3.3**) to unravel the likely molecular basis of adaptive mutations, and to identify understudied mutation sites that may have major pathogenic consequences.

The sensitive mutations that we aim to find with such comparative structural analysis are not always clearly detectable. For example, Zhang et al., showed that a single peripheral mutation involving residues of similar properties (Q33E) in human Pin1 caused a significant reduction of protein thermostability ***(7)***. Therefore, we note that although the present method does not lead to conclusive results *per se*, it is a valuable approach to identify likely key mutations for phenotypic variation and, with that, establish priorities for further investigation through more extensive computational and experimental techniques. Additionally, the integration of structural proteomics with other ‘omics layers of information is crucial for enhanced robustness of the proposed hypotheses regarding the functional impact of specific mutations.

1.3 Tracking SARS-CoV-2 evolution

The mutations occurring as the SARS-CoV-2 virus spreads across the globe into millions of human (and in many cases non-human) hosts represent potentially adaptive responses. These changes have implications for drug and vaccine development throughout the pandemic. They can also provide an important means of real time tracking of the spread of strains that cause varying disease severity and may affect the use of antiviral treatments or vaccines. A case in point is a “variant of concern” known as the B.1.1.7 lineage, or listed as VOC-202012/01 by the CDC. It was first detected in the United Kingdom in November 2020 and is suspected to be the strain that is overwhelming medical facilities and increasing mortalities across the globe. Much attention has been focused on a single mutation in the spike protein (N501Y) as the cause of its dominance, but the geospatial distribution and temporal appearance of co-segregating mutations so far have not been considered systematically. The availability of hundreds of thousands of sequences of the virus and corresponding metadata allows one to track its spatial and temporal molecular evolution. Unlike phylogenetic trees, MJN of haploid (typically) non-recombining genomes such as the SARS-CoV-2 virus facilitates the visualization of many valuable layers of information simultaneously, which can provide valuable insights into variants such as the B.1.1.7 lineage as it spreads. We have developed a computational systems biology pipeline to ingest, annotate, curate, interpret and display these diverse data types in the context of viral molecular evolution. Most of the methods we have employed have never before been used on this scale and therefore, we provide detailed notes on how individuals with limited access to HPC systems can execute this pipeline on typical user workstations, desktops or laptops.

1.4 Explainable artificial intelligence models

The main advantage of using Explainable Artificial Intelligence (X-AI) algorithms over traditional linear algorithms or Black Box AI is that X-AI is able to combine the accuracy and efficiency of modeling complex systems (like Black Box AI) while maintaining the ability to produce results that are human-interpretable (like traditional methods). X-AI methods such as, iterative Random Forest (iRF) ***(8)***, iterative Random Forest Leave One Out Prediction (iRF-LOOP) ***(9)***, and Random Intersection Trees (RIT) ***(10)*** are used to determine feature importance and interaction. By applying iRF-LOOP with RIT on the SARS-CoV-2 virus mutations across samples, we gain a better understanding of which mutations are associated and may be coevolving. This allows for the generation of hypotheses, such as on compensatory mutations or if specific mutations are causative for higher mutation rates in other parts of the sequence. Although our application here is to address the molecular evolution of the SARS-CoV-2 virus, these methods can be used on highly diverse data types.

1.5 Understanding COVID-19 pathogenesis through gene ontology and multi-omics network analysis

Understanding the mechanism underlying pathology is critical to developing new treatment strategies against SARS-CoV-2 infection and the resulting COVID-19 disease. Analysis from publicly available transcriptomics datasets from SARS-CoV-2 patients can be integrated with existing databases of human gene expression (e.g., HumanNet ***(11)*** and Genome-Tissue Expression Project, GTEx ***(12)***) to obtain mechanistic insights on pathogenesis. Differentially expressed genes caused by a viral infection can be categorized by using protein function and phenotype ontologies to identify common biological pathways involved in the course of the disease. Incorporating viral-host protein interaction networks into downstream graph traversal analyses can also identify genes of interest that may be differentially expressed in the host due to the direct binding of viral to host proteins. Furthermore, integrating drug-to-target networks (e.g., DrugBank, ChEMBL) with differentially expressed genes involved in pathogenesis can suggest putative treatments based on biologically-informed results.

2. Materials

2.1 Computational resources, softwares, and packages

The SARS-CoV-2 dataset was analyzed using the supercomputing capacity of Summit and Rhea on the Oak Ridge Leadership Computing Facility (OLCF) supercomputer platform at the Oak Ridge National Laboratory's (ORNL). Summit is composed of 4,608 compute nodes, each equipped with 512 GB of DDR4 memory for use by the two 22-core IBM POWER9 processors as well as six NVIDIA Volta V100 graphics processing units (GPUs). Routine calculations were performed on a standard laptop or desktop. Table 1 provides a list of the main publicly available software packages/libraries used in the described methods.

Table 1. Main publicly available software packages/libraries used.

Resource	**Reference / Source**
CLC Genomics (viewer is freeware)	[digitalinsights.qiagen.com/downloads/product-downloads/]
Cytoscape	[cytoscape.org]
data.table	https://cran.r-project.org/web/packages/data.table/index.html
DESeq2	https://bioconductor.org/packages/release/bioc/html/DESeq2.html
EdgeR	https://bioconductor.org/packages/release/bioc/html/edgeR.html
igraph	[https://igraph.org/r/; https://igraph.org/python/]
iterative Random Forest (iRF)	[https://github.com/Jromero1208/RangerBasediRF, https://cran.r-project.org/web/packages/iRF/index.html]
Kraken2	[https://ccb.jhu.edu/software/kraken2/]
Pathview	[https://bioconductor.org/packages/release/bioc/html/pathview.html, https://pathview.uncc.edu/]
PopArt	[popart.otago.ac.nz/index.shtml]
plyr	https://cran.r-project.org/web/packages/plyr/index.html
Random Intersection Trees (RIT)	[https://rdrr.io/cran/FSInteract/man/RIT.html]
Samtools	[http://www.htslib.org/]
Scikit-Bio	[http://scikit-bio.org/docs/0.5.0/index.html]

SRA Tool Kit	https://github.com/ncbi/sra-tools
STAR	https://github.com/alexdobin/STAR
Vegan	[https://cran.r-project.org/web/packages/vegan/index.html]
Visual Molecular Dynamics (VMD)	https://www.ks.uiuc.edu/Research/vmd/

3. Methods

3.1 Pathogen detection and analysis

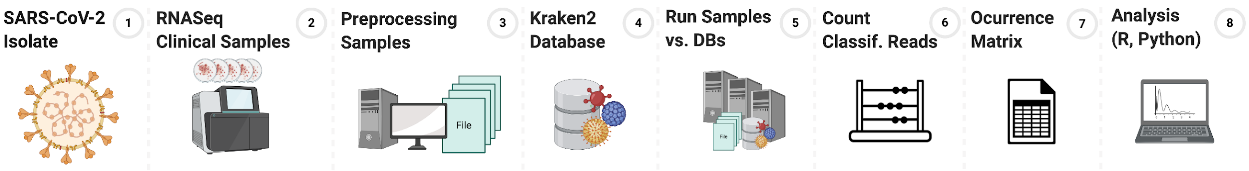


Figure 2. Workflow for Pathogen Detection: Strategy Overview - **(1)** Obtain genome sequence of pathogen isolate; **(2)** Obtain bulk RNASeq clinical samples; **(3)** Preprocessing samples; **(4)** Create or download database with whole-genome sequences of pathogens of interest; **(5)** Compare RNASeq clinical samples against the database; **(6)** Count the classified reads; **(7)** Occurrence matrix; **(8)** Run data analysis.

Pathogen detection

1. Obtain the genomic sequences of viral isolate of interest. The sequence of a closely related virus could also be used.
2. Obtain bulk RNASeq clinical samples from targets of interest with appropriate controls if needed.
3. Download the human genome or transcriptome from the NCBI webportal (assembly database, GRCh38).
4. Preprocessing-samples:
    a) Assess the quality of the RNASeq data using FastQC (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/). Perform adapter and quality score trimming (*see* **Note 1**). Align the RNASeq reads to the host transcriptome using BWA ***(13)***, or against the whole genome using STAR ***(14)*** (*see* **Note 2**). Extract the unmapped reads - these will be sequence samples used in taxonomic identification (*see* **Note 3**). Alternatively CLC Genomics Workbench (v. 20.0.3, Qiagen, Hilden, Germany) can perform trimming using default parameters and alignment.

b) Determine the quality of the alignment (*see* **Note 4**).

c) Splitting Sequence Samples: If the samples are too big, it may be necessary to split the sequence samples into various files to process them in parallel, reducing overall execution time. Be sure to keep their original IDs while renaming the series.

5. Use a pre-built database from Kraken2 (https://ccb.jhu.edu/software/kraken2/) if it contains the viral isolates of interest. Otherwise, build a custom database by utilizing publicly available whole-genome sequences. Remember to add the sequences of the viral isolates in addition to targets/pathogens of interest (*see* **Note 5**).
6. Run all the unmapped sequenced clinical samples against the Kraken2 database (*see* **Note 6**). If the sequence samples were split in the pre-processing phase, it is necessary to merge the resulting files to obtain a valid result with all counts for the taxonomic identification of a given single sample.
7. Count the number of classified reads for a given taxa for each respective sample. Kraken2 uses the letters C/U at the beginning of each line in the output to identify if a read was classified or unclassified (*see* **Note 7**).
8. Generate an occurrence matrix after filtering for fungal, bacteria, and/or viral taxa. Here, in the matrix each row represents a taxa, each column, a sample, and the values indicate the number of reads classified as the given taxa in the particular sample (*see* **Note 8**).

Data analysis

This analysis can be performed using a scripting language of choice, such as R or Python.

9. Identify potential sequence contaminants from the occurrence matrix. These may include for example PhiX174microvirus. These potential contaminants can be discarded.
10. From the occurrence matrix, generate a bar plot of the number of taxa identified per sample. The bar plot should be generated for the lowest level of specificity (*see* **Note 7**). This will indicate samples that may be outliers. A sample can be considered an outlier if it has an abnormal number (either too large or small) of taxa identified for that sample. Discard these outlier samples.
11. Normalize data to account for library size biases, and generate relative abundance values (*see* **Note 9**).
12. Determine if there is sufficient confidence in using the data quantitatively or qualitatively. This can be determined by the original quality assessment of the sequencing run together with the library size count (*see* **Note 10**).
13. Aggregate data to a specific taxonomic level, e.g. phylum. Calculate a sample-based distance matrix, thereby resulting in a sample by the sample matrix. Where each value in the matrix represents the dissimilarity between the respective samples. Perform an

ordination analysis on the samples, such as Principal Coordinate Analysis (PCoA), and plot the result (*see* **Note 11**).

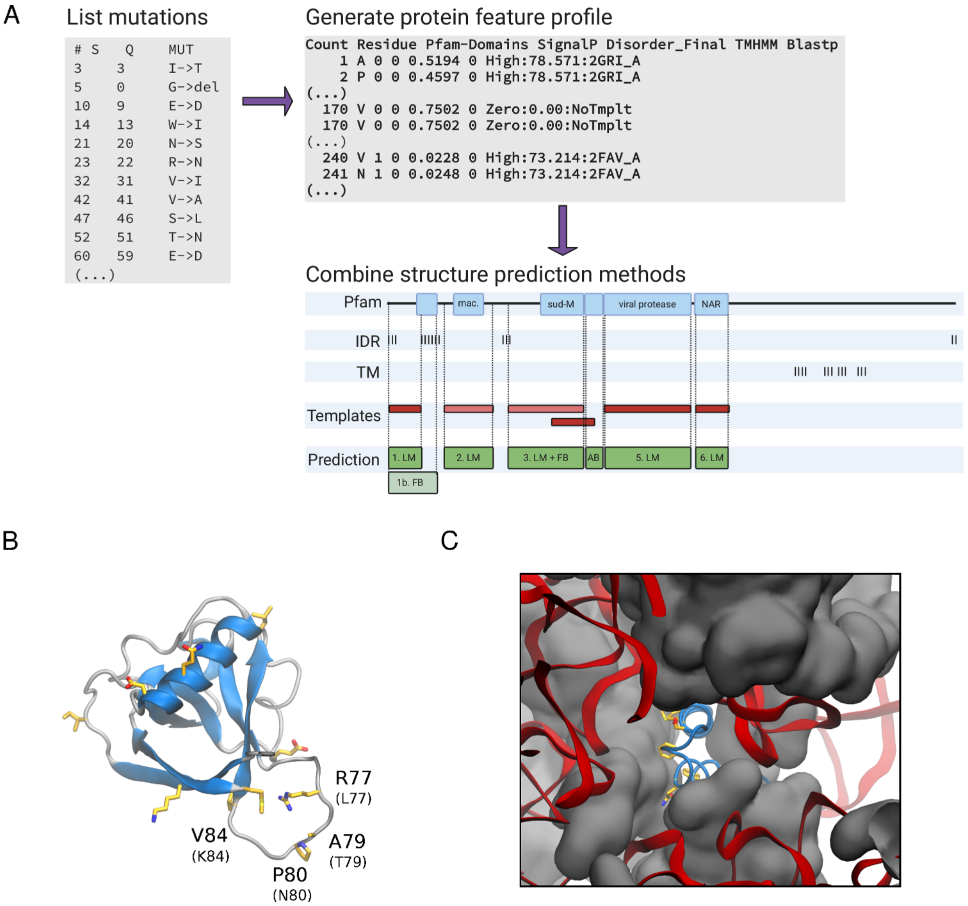
14. Based on the metadata and dispersion in the PCoA plot, factors that may drive the dispersion of the plot can be investigated for statistical significance. This is done by a PERMANOVA analysis (*see* **Note 12**).
15. Perform an alpha-diversity analysis over the samples. Any replicates of samples can be averaged at this step if desired. Standard alpha-diversity indices are Shannon, Simpson, and Chao1 (*see* **Note 13**).
16. Repeat **Steps 13-15** for different taxonomic levels, to better understand the data.
17. Obtain a list of known or potential pathogens from publicly available databases, such as PHI-base (http://www.phi-base.org/) and VEuPathDB (https://veupathdb.org).
18. Determine the influence of potential pathogens on the microbial communities. Perform **Steps 13-14** on a taxa basis (previously the analysis was performed on a sample basis). Classify taxa as either potentially pathogenic or not. Use this in a similar way as the sample metadata was used in **Steps 13-14** (*see* **Note 14**).

Table 2. Classification of potential conservative substitutions applied in our studies. Amino acids in brackets have to be analyzed in the structural context.

Type	**Amino Acid**
Glycine	G
Aromatic	Y, W, [H], [F]
Non-polar	A, I, V, L, M, [F], [P]
Polar, hydrogen bond interacting	S, T, [C]
Amidic, hydrogen bond interacting	N, Q
Acidic	D, E
Basic	R, H, K

3.2 Proteome-wide structural comparison of SARS-CoV-2 with evolutionary-related species

1. Establish the reference genomes and perform proteome-wide primary sequence comparison (*see* **Note 15**): The genome MK062179 can be used for SARS-CoV, for example, and NC_045512 for SARS-CoV-2 (both available on NCBI website) ***(15)***. Download from NCBI the amino acid sequences in FASTA format of all mature SARS-CoV and SARS-CoV-2 proteins. Use Clustal Omega ***(16)*** or ClustalW2 (available for download from http://www.clustal.org/clustal2/) for pairwise sequence alignment. Export in FASTA format. List the position and identity of all mutations (substitutions, deletions, and additions) (Fig. 3A).



```
# S    Q     MUT
3      3     I->T
5      0     G->del
10     9     E->D
14     13    W->I
21     20    N->S
23     22    R->N
32     31    V->I
42     41    V->A
47     46    S->L
52     51    T->N
60     59    E->D
(...)
```

```
Count Residue Pfam-Domains SignalP Disorder_Final TMHMM Blastp
    1 A 0 0 0.5194 0 High:78.571:2GRI_A
    2 P 0 0 0.4597 0 High:78.571:2GRI_A
(...)
  170 V 0 0 0.7502 0 Zero:0.00:NoTmplt
  170 V 0 0 0.7502 0 Zero:0.00:NoTmplt
(...)
  240 V 1 0 0.0228 0 High:73.214:2FAV_A
  241 N 1 0 0.0248 0 High:73.214:2FAV_A
(...)
```

Figure 3. Main steps for comparative protein structure analysis. A) Sequence parsing and structure prediction of SARS-CoV-2 nonstructural protein 3 (nsp3). List site mutations using a text editor of choice (*see* **Step 1**). The "S" column corresponds to the mutation sites in the subject sequence (here, SARS-CoV nsp3) and "Q" column corresponds to the mutation sites in the query sequence (here, SARS-CoV-2 nsp3). Next, collect and assemble a table (the protein feature profile) with functional and structural information for the query sequence from available public repositories (*see* **Step 2**). Then, partition the sequence into logical sub-regions that can be modeled independently (*see* **Step 6**). Choose the appropriate combination of structure prediction methods, as exemplified (LM: local modeling, FB: fragment-based modeling, AB: *ab initio* modeling). Find the detailed decision process for this example case in ***(3)***.

We note that this example represents a moment prior to the release of experimentally solved structures of SARS-CoV-2 nsp3. **B)** SARS-CoV-2 nonstructural protein 1 (nsp1, a.a. 13-127, PDB ID 7K3N) with the β3-4 loop (a.a. 76-81) built *in silico*. Non-conservative substitutions relative to SARS-CoV are depicted in licorice representation (*see* **Step 9**). In Prates et al. ***(3)***, it is suggested that the substitutions in the β3-4 loop, namely, Leu^{77}Arg, Thr^{79}Ala, Asn^{80}Pro, and Lys^{84}Val, may directly impact pathogenicity. **C)** C-terminal fragment of SARS-CoV-2 nsp1 bound to rabbit 40S ribosome complex (protein domains are represented in grey and the rRNA, in red, PDB ID 7JQB) ***(17)***. Non-conservative substitutions are found in the region and may affect the interaction of nsp1 with 40S.

2. Gather information on protein topology and structure-function relationships from the literature related to query proteins and homologs (*see* **Note 16**). UniProt Knowledgebase (UniProtKB) is a valuable resource for locating integrated protein sequence and related functional information ***(18)***. Generate a *feature profile* for each protein by mapping the applicable features enumerated below to the respective residue position(s) in the amino acid sequence. Complement this data curation using recommended prediction methods or servers, listed in parenthesis.

   a) Signal peptide (signalP) ***(19)***.

   b) Intrinsically disordered regions (SPOT-disorder) ***(20, 21)***.

   c) Protein domains (PFAM) ***(22)***

   c) Transmembrane regions (TMHMM) ***(23)***.

   d) Experimentally solved regions, either of the query sequence itself or of homologs: Use the Protein BLAST (blastp) ***(24)*** server to search for the available solved structures. Configure the blastp query to use the Protein Data Bank (PDB) as the search set and use the expected threshold of 0.001. Register the PDB identification code(s) (PDB ID) of the sequence(s) with the highest alignment score and its corresponding identity value relative to the query sequence.

   e) Known posttranslational modification (PTM) sites, such as phosphorylation, ubiquitination, O/N-glycosylation, palmitoylation, disulfide bridges, proteolytic cleavage, and sumoylation ***(25)***.

   f) Catalytic and auxiliary catalytic residues.

   g) Other key functional sites identified with mutagenesis experiments.

3. Query each feature server or use a local software package, as appropriate, with the protein sequence as input and obtain and parse the results into a per residue array of values. Depending on the feature, the entry for each residue could be presence/absence, a category type, or a quantitative measure (Fig. 3A). Create a protein sequence profile file for marking per residue feature information, with a column for each feature. Thus, the first two columns would have the residue

numbers and amino acid codes, respectively, followed by the feature columns. This file can then be used for visual inspection or as an input to a script for the application of a set of rules (like those described in **Step 6**) for partitioning the sequence into logical, independent regions to which the appropriate structure prediction methods can be assigned.

4. To date, most of SARS-CoV and SARS-CoV-2 proteins are at least partially experimentally solved, including some of the protein complexes they form with host proteins. The summary of available structures is frequently updated on SARS-CoV-2 NCBI resources ***(26)***. Download solved structures from the PDB website (https://www.rcsb.org/).

5. Create a code (for example, in python) to easily shift residue sequence numbers (columns 23-26) of PDB files - it will be helpful in different steps of this protocol.

6. If a predicted structured region is not yet experimentally solved, the feature profile generated in **Step 2** for each protein can be used to define the optimum combination of the state-of-the-art methods of protein structure prediction, in a case-by-case manner (Fig. 3A). The following main decision steps are suggested:

   a) Identify the regions that are amenable for modeling along the protein sequence. An overlap with templates and/or predicted domains, in contrast to predicted intrinsically disorder, indicates stable structured regions.

   b) Search for models generated with well-established methods of protein structure prediction. We particularly recommend the models predicted with the AlphaFold2 system (*see* **Note 17**), which was used to solve understudied SARS-CoV-2 proteins, such as, M, nsp2, nsp4, nsp6, and the C-terminal domain of nsp3 ***(27)***.

   c) For regions of high identity (>70%) between target and template: In this case, map the substitutions on the template structure to evaluate their likely structural impact, as described in **Step 9**. If all the substitutions are structurally conservative (*see* **Note 18**), they can be locally modeled (LM) directly on the template structure, as well as any short missing loop (*see* **Note 19**) (Fig. 3B). However, follow the next item (d) for targets involving structurally non-conservative substitutions, including deletions and additions on structured regions or long missing loops (>10 amino acid residues). Check for missing loops (not terminals) on the header of the template PDB. Build the short missing loops using RosettaRemodel ***(28)***, following these main steps:

      i) Install Rosetta as described on available documentation online (https://www.rosettacommons.org/). We have been using Rosetta 3.10.

ii) Generate a blueprint file from the starting PDB:

```
rosetta/tools/remodel/getBlueprintFromCoords.pl -pdbfile  [starting pdb] -chain [chain
id] > [blueprint file]
```

iii) Edit the generated blueprint file to build the loop (see Fig. 2 in Huang et al. ***(28)***) . For example.:

```
496 Y .
497 R .
498 K L PIKAA K
0 X L PIKAA P
0 X L PIKAA N
0 X L PIKAA G
0 X L PIKAA T
0 X L PIKAA N
0 X L PIKAA P
499 G L PIKAA G
500 V .
```

The blueprint above will insert the loop between positions 498-499 and leave the other positions fixed (*see* **Note 20**).

iv) Create the flag file defining the input/output files and the parameters to run RosettaRemodel. Find an example of a flag file under `[rosetta_path]/demos/tutorials/loop_modeling` in the Rosetta folder. In a multi-core machine, run RosettaRemodel with:

```
[rosetta_path]/main/source/bin/remodel.mpi.linuxgccrelease @flag_missing_loops
```

After building the missing loops, renumber residue IDs according to the target protein sequence. Then, model substitutions using Rosetta fixbb application ***(29)***. The first step for that is to change the name of the residues to be mutated on the PDB file (columns 18-20). Then, remove the lines corresponding to its side chain atoms, as shown in the example below:

```
ATOM     25  N   PRO A   4      32.444 -19.963  25.233  1.00 23.61           N
ATOM     26  CA  PRO A   4      31.025 -20.351  25.373  1.00 25.45           C
ATOM     27  C   PRO A   4      30.656 -21.452  24.410  1.00 24.09           C
ATOM     28  O   PRO A   4      31.273 -21.576  23.345  1.00 25.08           O
ATOM     29  CB  PRO A   4      30.228 -19.078  24.989  1.00 24.50           C
ATOM     30  CG  PRO A   4      31.206 -17.996  25.084  1.00 27.60           C
ATOM     31  CD  PRO A   4      32.593 -18.562  24.871  1.00 23.74           C
ATOM     32  N   MET A   5      29.610 -22.194  24.729  1.00 24.91           N
ATOM     33  CA  MET A   5      28.978 -23.082  23.738  1.00 23.63           C
ATOM     34  C   MET A   5      28.512 -22.367  22.492  1.00 24.81           C
ATOM     35  O   MET A   5      27.936 -21.275  22.537  1.00 21.89           O
(...)
```

In this case, the residue name in position 5 was changed to methionine, keeping only the lines corresponding to backbone atoms. This will be the input to run fixbb as described in *demos* of the Rosetta documentation online.

d) Medium identity (30-70%) between target and template, water-soluble region: Use the I-TASSER suite ***(30)*** for fragment-based structure prediction (FB) (*see* **Note 21**). Restrict the length of disordered terminals that do not overlap with the templates to not more than five amino acid residues. By doing so, the I-TASSER quality metrics (c-score) will better reflect the prediction accuracy of conserved structured domains.

e) Low identity (<30% or no template found) between target and template, water-soluble regions: Submit the protein sequence region to deep learning-guided *ab initio* modeling (AB) using the trROSETTA workflow (*see* **Note 22**) ***(31)***. The sequence can be submitted to the trRosetta webserver https://yanglab.nankai.edu.cn/trRosetta/. Alternatively, the software can be downloaded for local use from https://github.com/gjoni/trRosetta.

f) Transmembrane regions of medium to low identity between target and template or template free: Submit the corresponding sequence region to AB modeling using the C-I-TASSER server.

As shown in Fig. 3A, one may combine the approaches above to model regions overlapping templates of different sizes and identities.

7. Collect information about the biological assembly of each protein. This can often be found on the Structure Summary of the corresponding PDB webpage. The oligomeric state of a protein or the complex it forms with other biomolecule(s) may be built by structural alignment with available structures (Fig. 3C).

8. Structural alignment of homologous proteins can be done using the *multiseq* plugin of Visual Molecular Dynamics (VMD) ***(32, 33)***. The alignment of specific regions of the proteins that do not necessarily have the same size can be done using LovoAlign ***(34)***. In this case, the selection of the regions to be aligned can be done by identifying it in the column of occupancy (columns 55-60) of the PDB files, for example.

9. Locate the mutation sites on the solved/predicted protein structures: For each protein, load the structures on the VMD program. Set display mode to *orthographic*, use *NewCartoon* representation for the full protein, and display mutation sites using the *licorice* representation (Fig. 3B, *see* **Note 23**). Visual analysis can help to predict if the mutation will affect the local/global protein structure and dynamics. Consider checking the following aspects for each mutation site:

a) It is a buried or a surface-exposed site. For a more quantitative analysis, count the number of contacts of specific residues using the VMD *Timeline* plugin (*calc. inter-sel. contacts*). Define selected atoms as pairs "`{resid [residue number] and name CB}`" and "`{protein}`". Set a distance cutoff of 4 Å. One can assume that buried residues have a higher number of contacts than exposed residues and define a cutoff between the two types.

b) It likely affects the interaction with other residues, such as hydrogen bonds, salt bridges, π-stacking, and hydrophobic contacts. As a preliminary analysis, one may leave out conservative substitutions (*see* **Note 18**, Table 2). On the VMD Display window, press "1" and click on a residue to see its identity and number. Distances and angles within and between residues can also be measured (activate measurement pressing "2" and "3", respectively) to check the possibility of interaction with neighbouring residues (*see* **Note 24**).

c) It likely forms or subtracts a disulfide bridge (*see* **Note 24**).

d) It adds or subtracts a PTM site.

e) It likely affects the local secondary structure and, thus, protein flexibility. Check residue-specific secondary structure propensity ***(35)***.

f) It is located on the interface with a neighboring protomer in the biological assembly (*see* **Step 7**).

Taken together, the assembled feature profile and this structural analysis can be used to identify likely functionally and/or structurally non-conservative substitutions (*see* **Note 25**).

3.3 Tracking the evolution of SARS-CoV-2

1. Obtain weekly download sequences in FASTA format along with corresponding metadata. Process them using the following steps and append them to previous sequences (*see* **Note 26**).
2. Align full sequences with MAFFT ***(36)*** to an established reference genome (NC_045512 for SARS-CoV-2) (see the manual for MAFFT).
3. Upload aligned sequences into the software in which alignments can be trimmed (*see* **Notes 27-28**).
4. Trim sequences to the start and stop codons (nsp1 start site and ORF10 stop codon).
5. Export aligned sequences and generate a count of mutations per site (*see* **Notes 29-30**).
6. Generate a count of "non-reference" mutations (i.e., the number of individuals that differ from the reference per site). Subtract this number and the number of "N" at each site from the total number of sites. Remove sites with fewer than ten variable (i.e., non-reference) sites (*see* **Note 31**).
7. Remove likely noise. First, divide the "N" counts at each site by the total number of variable sites and remove the sites in which the "N" count is more than 20% of the total variable sites. Next, calculate the total number of non-"N" for the remaining sites and remove those with fewer than ten non-"N" counts. From this final set of "true variable sites", remove all individuals (rows) that have an "N" at

any site. The remaining data set should maximize the number of important variable sites in the context of molecular evolution and maximize the sample size.

8. Create the **Variable Site File.** Format the file so that column 1 is the GISAID accession number, column 2 is a "null" character such as a period (".") and the remaining columns are the variable sites. Have a header for each column. Save the file as a tab-delimited file, open it in BBBedit or any text editor of choice (*see* **Note 29**) and remove tab spaces ("find and replace" "\t" by nothing, ""). Replace the period character in column 2 with a tab space . This produces a 2-column file where the first column is a unique identifier and the second is a concatenated sequence of all variable sites - *the haplotype* - for downstream analyses.
9. Open the file in R using "header=TRUE" and produce an object that consists of the haplotype column. Remove all duplicate sequences using the "unique" function. This is the list of unique haplotypes for the entire population of sequences. This can also be done in Excel^{TM} using the "remove duplicates" tool but with large numbers of variable sites it may overload the system.
10. Create the **Unique Haplotype File.** Label each of the haplotypes (e.g., Hap001, Hap002, etc.). This will produce a 2-column file in which column 1 is the haplotype name and column 2 is the haplotype sequence.
11. Create the **Haplotype Network File.** Using R, open the **Variable Site File** and the **Unique Haplotype File**. Make sure that the columns with haplotype sequences in each of the two files have the same header name. Use the "join_all" function in the package *plyr* to "left join" the **Variable Site File** to the **Unique Haplotype File** using the header of the haplotype sequence column. This will produce a new file in which column 1 is the GISAID accession, column 2 is the null character, column 3 is the haplotype sequence, and column 4 is the haplotype name. If using Excel^{TM}, the VLOOKUP formula may work but with large numbers of variable sites it may take very long time on a standard personal computer.
12. Create the **Median-Joining File.** Produce a new object in R that consists of columns 3 and 4 from the **Haplotype Network File**. Remove duplicate haplotypes using the haplotype name and change to NEXUS format (*see* **Note 32**). One should now have a file of unique haplotype sequences and the n haplotypes labels.
13. **Cytoscape Network File**: Open the **Median-Joining File** in the software PopArt ***(37)*** , produce a median-joining network setting epsilon to 0. Export the network as a table - column 1 is a haplotype, column 2 is an edge, and column 3 is the haplotype that is connected to the one listed in column 1.
14. Create the **MetaTable File** for upload to Cytoscape ***(38).*** Open the metadata file downloaded from GISAID, and the **Variable Site File** in R. Use the *plyr* package

and “left join” the GISAID metadata csv file to the **Variable Site File** using the GISAID accession number.

15. Calculate **Success Metric:** Generate counts of individuals per haplotype, number of geographic regions that a haplotype is found in, and the number of days it has persisted (*see* **Note 33**). Divide the number of individuals by the number of days and then by the number of geographic regions (*see* **Note 34**). This is the absolute success number. Unitize the metric by dividing each of these numbers by the lowest success number (1 will then be the lowest number).
16. **Cytoscape MetaTable File:** Generate a file from the **MetaTable File** and the **Success metric** data with the haplotype name, the number of individuals with that haplotype, and the success metric of each haplotype.
17. **Cytoscape Network:** Load the **Cytoscape Network File** in Cytoscape and define source, edge, and target nodes (columns 1, 2, and 3 respectively). Import the **Cytoscape MetaTable File.** Cytoscape is very flexible and one can incorporate any information of interest, as long as it is imported with the same haplotype naming structure as the **Cytoscape Network File**.
18. Reduce feedback loops due to homoplasy by removing haplotypes with fewer than 10 individuals. See Cytoscape manual for further information on network formatting.

3.4 X-AI driven predictive models

Run iRF-LOOP:

1. Format data into sample-row and feature-column format. Remove sample IDs.
2. For each feature, create and run an iRF (*see* **Notes 35-36**) model using an iRF codebase, either C++ ***(9)*** for big data or R ***(8)*** for small data (*see* **Note 37**).
3. Normalize each feature importance score for each iRF model by dividing each feature importance value by the sum of that model’s feature importance.
4. Add a ‘Dependent Variable’ Column to the normalized feature importance files.
5. Combine all the normalized feature importance files together (*see* **Note 38**)

Run RIT:

6. Run RIT ***(10)*** on the path files from each of the resulting iRF models, producing the sets of interacting features in each model. (*see* **Notes 39-41**)

7. Analyze the sets for anomalies, such as features that appear in many sets or always appear with specific other features.

<u>iRF-LOOP and RIT applied to molecular evolution:</u>

By running iRF-LOOP on the mutations in the sequences of SARS-CoV-2 virus one can obtain information regarding site coevolution.

8. Format the virus sequence data such that each row is a unique sequence and each column is a mutation from a given reference, with the matrix cell values being binary - zero for ‘does not contain this mutation’, and one for ‘does contain this mutation’.
9. Run iRF-LOOP (*see* **Steps 1-5**) on the mutation data file.
10. Create a graphical network of the results. Mutations that act as a destabilizer, triggering several new mutations, will be connected toward them and mutation hotspots will be pointed at by the number of preceding mutations.
11. Run RIT (*see* **Steps 6-7**).

3.5 Understanding effects of SARS-CoV-2 infection on hosts

<u>Transcriptomics from infected patient and control samples</u>

1. Obtain samples and reference genome as described in **Section 3.1, Step 2**. PRJNA605983 includes nine bronchoalveolar lavage fluid samples from patients infected with SARS-CoV-2. NCBI's SRA run selector provides a table of metadata for each sample. This study does not include control samples; therefore, the metadata are used to identify PRJNA434133, which includes forty control samples sequenced using similar methods. Download the FASTQ data files using NCBI's SRA Toolkit software ***(39)***:

    a) Find the study on the SRA Run Selector site (acc=PRJNA605983): https://www.ncbi.nlm.nih.gov/Traces/study/

    b) Select Accession List to download the list of runs as a text file (SRR_Acc_list.txt)

    c) Download the SRA archive files. This command will download an SRA compressed file for each run in the study:

    ```
    prefetch --option-file SRR_Acc_list.txt
    ```

    d) For each run, extract the fastq files from the SRA compressed file with

```
fasterq-dump SRR11092056.sra
```

2. Align the sample reads to the reference. The SARS-CoV-2 genome can be used as a reference (MN908947) in addition to the human reference genome (GRCh38).

3. Once the sample data and reference genomes have been downloaded, align reads to the reference as described in **Section 3.1, Step 4**. Alternatively, create a combined reference with alignment software such as STAR ***(14)*** to simplify assigning reads.

4. After alignment, reads mapped predominantly to repetitive regions can be discarded. This includes, for example, reads that map to Alu elements, pseudogenes, and repetitive sequences should be removed.

5. Following alignment and read counting, counts should be normalized using the Trimmed Mean of M-values (TMM) method for cross-study comparisons (i.e., if infected and healthy samples are from different sources). This method is available in the edgeR package via the `calcNormFactors` function ***(40)***.

6. Ordination plots (e.g., PCA or UMAP) can be used for exploring the initial results; these methods project high-dimensional data (n-dimensions=n-genes) into fewer dimensions (typically 2-3) for visual inspection. Particular attention should be paid to any outlier samples that may need to be removed before differential expression analysis.

7. Differential gene expression (DGE) analysis can be done with edgeR or DESeq2 ***(41)***. Inspect transcript-level expression values for differentially expressed genes (DEGs) of interest. Remove transcripts mapping to artifacts that were not removed automatically, as well as those that encode a truncated form of the protein. Once these are removed, repeat the analysis to verify reproducibility.

8. Example workflow for differential expression analysis with edgeR:

```
dge <- DGEList(count_matrix, group = design_matrix)
dge <- calcNormFactors(dge)
design <- model.matrix(~design_matrix)
dge <- estimateDisp(dge, design = design)
fit <- glmFit(dge, design = design)
lrt <- glmLRT(fit)
topTags(lrt)
```

Multi-omics integration and generating a mechanistic model

9. Pathway analysis. Pathway visualization can be done with Pathview either online (https://pathview.uncc.edu/) or with the R package ***(42, 43)***. This tool maps numerical data (e.g., expression data or fold-change of DEGs) to genes in KEGG pathways as a color gradient. The user provides a table with gene IDs as the first column and expression or fold-change values in rows across one or more additional

columns. This can be used in combination with enrichment analysis to explore DEGs in pathways of interest.

10. Enrichment analysis.

    a) In order to derive biological relevance, lists of differentially expressed genes can be tested for functional enrichment through various ontologies (i.e., biological processes, phenotypes, coexpression, diseases, and transcription factor binding sites) in the ToppGene Suite (toppgene.cchmc.org) ***(44)***, which integrates Gene Ontology with various databases, including PANTHER ***(45)***, KEGG ***(46)***, Reactome ***(47)***, MSigDB ***(48)***, and DrugBank ***(49)*** (*see* **Note 42**).

    b) On the ToppGene web portal, access ToppFun for functional enrichment lists. Copy and paste lists of DEGs by HGNC symbol, Entrez ID, Ensembl ID, or RefSeq ID (*see* **Note 43**).

    c) Under Calculations, select among the several options (e.g., GO, Human Phenotype, Domain Pathway etc.) for ontology features. Ontology enrichment thresholds (*p*-values and *p*-value correction) can also be adjusted as desired. Default values for this section can be used if exploring a gene list for the first time (all features, FDR-corrected *p*-value cutoff < 0.05).

    d) Compare reported clinical symptoms from the disease to be studied to ontologies from ToppFun. In particular, the GO: Biological Process, Human Phenotype, and Disease ontologies can be used to determine which genes contribute to certain pathological processes. Select "Genes from Input" to identify which genes from the input list contributed to the ontology and "Genes in Annotation" to see the complete list of genes in the ontology (*see* **Note 44**).

11. Integrating omics datasets through network analysis.

    a) Another strategy for integrating results from differential gene expression analysis is to use gene lists as starting genes (also known as "seed genes") for graph traversal using graphs composed of multiple gene-to-gene networks. To build a multi-layer network, first download existing published networks (e.g., HumanNet ***(11)*** or STRING ***(50)***) containing gene-to-gene edges from various lines of experimental evidence, including: high-throughput omics datasets, protein-protein interaction data, and genes co-cited in the same publication.

    b) Predictive-expression networks (PEN) can be useful components for integrated analyses and graph traversals.

(i) Genetic data are obtained through the RNASeq V8 data sets hosted by the Genotype-Tissue Expression (GTEx) project ***(12)*** . Extract individual parent tissue (SMTS) and accurate tissue site (SMTSD) RNAseq sample ID data from the V8 Annotations Sample Attributes data file. Genetic data per tissue type can then be matched with corresponding sample IDs in the Gene TPM data set.

(ii) For a tissue, apply iRF-LOOP, following the steps in **Section 3.4** (**Steps 1-5**), with each individual corresponding to a sample-row and the genes to the feature-columns (*see* **Note 45**).

(iii) See **Note 38** for the generation of a graphical network from the total concatenated importance files. The generated network contains directed edges between any two genes X and Y. Edges are weighted such that the greater its weight, the higher the probability of gene X predicting gene Y (*see* **Note 46**).

c) Once all desired network layers are downloaded, read in the files using R with header = FALSE. Be sure that all gene IDs are consistent across all networks (do not mix Ensembl IDs with Entrez IDs, RefSeq IDs, etc.) Create each network layer as an independent graph using the *graph_from_data_frame* function from the igraph package (*see* **Note 47**).

d) Using the *union* function from the igraph package in R to generate a gene-to-gene network composed of multiple network layers.

e) Read in lists of differentially expressed genes identified from the transcriptomic analysis. Ensure that these “seed gene” IDs are in the same format as all network layers. Using the parameters “unreachable = FALSE, order = TRUE, rank = TRUE”, use the *bfs* function from the igraph package to perform a breadth-first search (BFS) of the combined graph layers from each seed gene in the list. See Fig. 4 as an example of how BFS uses seed genes to explore communities with predictive power.

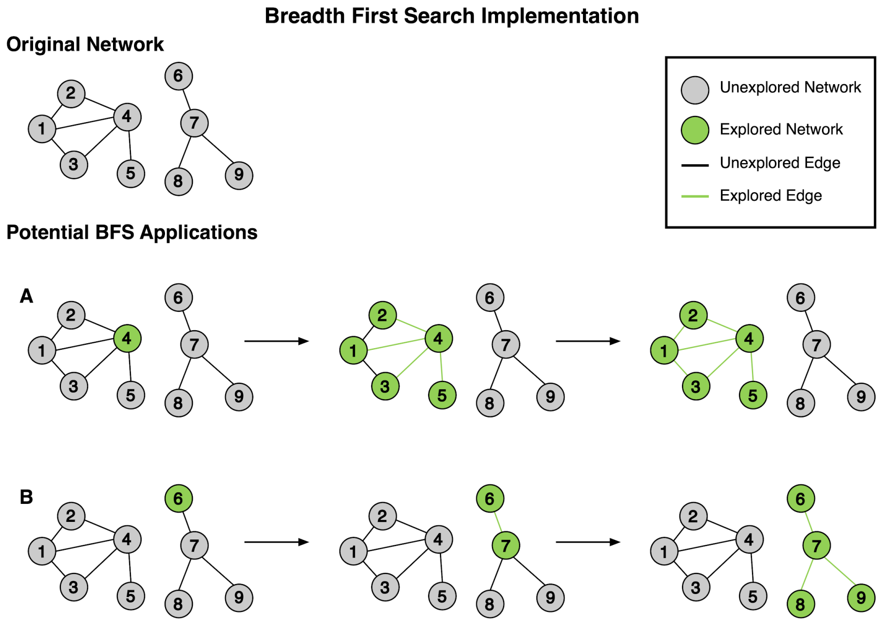


Figure 4. BFS on gene-to-gene network data. Two separate BFS implementations are shown. By implementing BFS from a list of seed genes of interest, a community of genes predictive of those seed genes can be extracted. Assuming nodes 4 and 6 in the above network are the genes of interest, implementing BFS with each as a seed (examples **A** and **B,** respectively) yields two separate communities with predictive power respective to these genes.

f) Use highly-ranked genes from BFS (e.g., top 50-100 ranked genes) and identify if these genes are differentially expressed in COVID-19 samples. Repeat the "Pathway Analysis" and "Enrichment Analysis" sections to glean biological or pathological insights.

12. Integrating host-viral protein-protein interaction networks. Download existing viral-host protein-protein interaction networks. These networks contain edges from high-throughput experiments used to determine which SARS-CoV-2 proteins interact with human proteins, with gene IDs used as edges ***(51)***. Human host genes of interest identified via differential expression or pathway, enrichment, or network analysis can be cross-referenced with these networks to determine if SARS-CoV-2 proteins can directly affect corresponding human proteins.

13. Repurposing approved drugs based on output.

a) Using genes of interest identified from **Steps 1-12**, putative treatments can be proposed using approved drugs that target the proteins encoded by these genes.

b) Download publicly available drug-to-drug target protein networks: examples include DrugBank ***(49)*** and ChEMBL ***(52)***. Import these networks into R, Python, or Excel^{TM}.

c) Search for edges containing genes of interest to identify possible drugs of interest that interfere with viral pathogenesis (*see* **Note 48**).

4. Notes

Notes from Section 3.1

1. Adapter and poly(A) tail sequences need to be removed from the reads. The Phred ***(14, 53)*** quality score of reads varies based on several factors. A suggested minimum threshold to apply is 20. This threshold corresponds to an estimated base call accuracy of 99%.
2. Using a splice-aware aligner (such as STAR) for RNASeq data is preferred. However, alternative strategies can include a pseudo-aligner, such as Kallisto or Salmon, against the transcriptome. If these options are not available, a genome aligner (for example BWA) can be used against the transcriptome.
3. Some aligners allow for the option to output unmapped reads as separate files. Use this option if available. Alternatively, Samtools (http://www.htslib.org/) can extract the unmapped reads using the command: `samtools view -f 4 sample.bam > unmapped.sam`.
4. Numerous potential biases can impact alignment. It is therefore impossible to do an exhaustive quality assessment. However, a few factors can be considered, such as: Are there any dramatic differences in the percentage of mapped reads between samples? How many reads are in the sample (i.e. library size)? Do the reads from a particular sample align preferentially to repetitive or low complexity regions? Is there a high percentage of mismatches among the aligned reads for a sample? Considering these can help determine if the mapping of a sample is reliable.
5. If computing capacity is limited, we suggest that a small set of genomes be used when constructing the custom database. The building of a Kraken2 database can be computationally intensive, and the resultant database may be too large for the subsequent classification. Furthermore, if a new database is being built, the FASTA sequences need to be processed to include the taxaID in the sequence identification line. Please refer to the Kraken2 manual for the required format (https://github.com/DerrickWood/kraken2/wiki/Manual).

6. If the flag `--use-names` is used, the subsequent taxa identification may be easier. Without this flag Kraken2 will use the numeric taxaID, which would then have to be manually parsed.

7. For the particular scientific question, it is important to determine the lowest level of taxonomic identification that is needed. For example, at the strain level, it may be unlikely that the particular strain in the database is also present within the clinical sample. However, taxa of the corresponding genera or species may be present. It is therefore up to the user to determine the level of specificity required for their analysis. We suggest an initial analysis at the species level together with additional analysis at the genera level.

8. We suggest that the taxonomic names include the full classification from kingdom to phylum all the way down to the level of specificity. This makes it easier to aggregate the occurrence matrix to a different taxonomic resolution.

9. There are several approaches to adjust for library size biases. We use a basic per sample normalization approach by dividing each taxa read count by the total number of reads in that sample. The normalized values are then scaled to a reasonable range by multiplying by a constant factor, such as 10e4. The scaling factor is important as some downstream analyses may require integer values. To obtain these values, we take the floor of the respective values, e.g. 42.7 becomes 42.

10. The threshold for choosing between qualitative or quantitative is hypothesis and data dependent. We suggest choosing a minimum read count threshold (or relative abundance threshold) and doing an initial qualitative analysis. This can be followed by a quantitative analysis if needed.

11. There are several software options for this. We suggest using either the Scikit-Bio package in Python, or the vegan package in R. The process is quite straightforward and is generally detailed by the package itself. For the distance matrix either UniFrac distance can be used (if the full phylogeny is available) or Bray-Curtis dissimilarity. Color the respective samples using metadata to investigate the resultant pattern in the plot.

12. The choice of analysis is dependent on the nature of the data and how appropriate the model assumption may be. In general, PERMANOVA is used when an Analysis of Similarity (ANOSIM) is performed to determine the significance of group variables (metadata). There are appropriately named functions (anosim) to do this in both Scikit-Bio and vegan.

13. Both the Scikit-Bio and vegan packages have documentation on how to perform these diversity analyses using the respective package.

14. There are several potential caveats at this step. The most likely scenario is that the number of putative pathogens may be small compared to the non-pathogens, thus leading to a class imbalance. In this case several runs of downsampling the non-pathogens can be used in a bootstrapping approach to investigate statistical significance from the ANOSIM analysis. Alternatively, for matrix values within a reasonable range a Fisher Test can be used. Finally, a descriptive approach may also be appropriate without a statistical analysis. An example of the latter would be clustering the dissimilarity results from a taxa-based Unifrac/Bray-Curtis calculation and then seeing where pathogen/non-pathogens cluster.

Notes from Section 3.2

15. Here, as an example, we consider the comparative structural proteomics of SARS-CoV and SARS-CoV-2, as in Prates et al. ***(3)***. Circulating in humans since, at least, December 2019, researchers have now registered more than 360,000 SARS-CoV-2 variants (GISAID database) ***(54)***. Here, the choice of using as the reference genome the first released sequence of SARS-CoV-2 (NC_045512) is based on the assumption that it has all the molecular features that drive its extremely differing spread rate relative to SARS-CoV ***(55)***.
16. Besides conducting a literature review using online engines for broad literature research, such as Google Scholar, PubMed Central, and Web of Knowledge, we particularly recommend looking for information about protein structure-function relationships at the literature cited on the Protein Data Bank (PDB) webpage of the protein structures related to the study.
17. The structures predicted with AlphaFold2, a newly developed deep learning-based engine, are particularly recommended as it showed a remarkable superior performance relative to other methods in the last Critical Assessment of Structure Prediction competition (CASP14). For instance, the predicted structure of ORF3a was solved by AlphaFold2 despite being a challenging target due to the unavailability of related sequences. In AlphaFold2, a folded protein is treated as a "spatial graph", where residues are the nodes and edges connect the residues in close proximity. It employs an attention-based neural network system and multiple sequence alignment (MSA) with evolutionary-related sequences to interpret and refine the structure of this graph, while reasoning over the implicit graph that it's building. By iterating this process, the system develops strong predictions of the protein's underlying physical structure and can efficiently determine highly-accurate structures. Additionally, AlphaFold2 can predict which parts of each predicted protein structure are reliable using an internal confidence measure.

18. Here, we classify the mutations that meet at least one of the following aspects as structurally non-conservative: it breaks/form disulfide bridges, it involves residues with different physicochemical properties (potential conservative substitutions are shown in Table 2), it likely adds/subtracts key interactions, or it breaks the local secondary structure. On the other hand, surface exposed conservative mutations that change the PTM pattern or the intermolecular interactions in a protein complex are also sites that may worth further investigation regarding their functional effects, even though they do not necessarily affect protein structure.

19. LM can be easily performed using the CHARMM-GUI interface ***(56)***, through the Solution Builder tool, for example. However, here we describe the use of Rosetta applications as they can be run locally and integrate a pipeline for proteome-wide structure comparison.

20. Note that position numbering in the blueprint file starts with 1 and it is not discontinuous on the missing loop regions. Thus, the blueprint numbering will not match the residue number of the PDB file. We suggest creating a copy of the template PDB file with renumbered residue positions starting with 1 to help identify the position in the blueprint file in which the loop will be inserted.

21. The newer C-I-TASSER workflow ***(57)***, which applies deep-learning for contact prediction generates more accurate models than I-TASSER, but a standalone package of C-I-TASSER is not yet available. Here, the choice of using the I-TASSER suite is due to the possibility of performing embarrassingly parallel runs of multiple targets in local machines. I-TASSER was ranked the best function prediction in CASP9 and the workflow has been continuously among the top prediction methods in the subsequent contests. In general, the method provides the correct global topology for the cases described in **Step 6** (**Section 3.2**), which is sufficient for the proposed goals. The estimated quality of the predicted topology is assessed with the TM-score (TM-score>0.5 indicates a correct topology). However, further refinement is highly advised ***(58)*** to use the I-TASSER predicted models on molecular dynamics simulations, as even minor local inaccuracies can be propagated once velocities are assigned to the atoms, leading to major conformational deviations.

22. The trRosetta (transform-restrained Rosetta) workflow generates protein structure models through two main steps: 1) the prediction of inter-residue orientations and distances via a deep residual-convolutional neural network which takes an MSA as input, and 2) energy minimization via a fast Rosetta model building protocol using distance and orientation restraints derived from (1).

23. Visual inspection is a relatively simple and important step in studying protein structure-function relationships. Therefore, getting familiar with the many user interface components and molecular drawing methods of VMD (or another

preferred software) is an important part of the procedure described here. For visual clarity, we recommend using a graphical representation that depicts the secondary structure backbone and the side-chain atoms of the mutation sites. For example, one can set:

- Protein backbone: Drawing method - *NewCartoon*; Coloring Method - *ColorID 13* (mauve);

- Mutation sites: Drawing method - *Licorice*; Coloring Method - *Name* (using orange for C atoms - the color of atom types can be defined in Graphics > Colors); Select only heavy side-chain atoms in Selected Atoms with `resid [mutation site numbers] and not name C O N and noh`.

24. Consider the estimated quality/resolution of the protein structure being used to make any inference about possible interactions between amino acid residues. For example, a typical hydrogen bond may be identified adopting as geometric criteria a cutoff of 3.0 Å for donor-acceptor distance and 20° for acceptor-donor-H angle. Similarly, a S-S distance cutoff of 3.0 Å is applied for disulfide bonds in the PDB database. These conditions should be relaxed depending on the resolution of the model. Evaluate if such criteria would be achieved with side-chain dihedral rotations or translational motions of residues in flexible regions like loops and coils.

25. We emphasize that the present method does not aim to provide, *per se,* conclusive information about the functional effects of site mutations. Ideally, it should instead be applied in conjunction with other omics layers to gather multiple lines of evidence for the hypotheses raised and guide further experimental and *in silico* studies. For the latter, we point out the DynaMut server ***(59)*** for a quantitative estimation of the impact of specific mutations of interest on protein flexibility and stability, for example. Moreover, a valuable part of the current efforts against the SARS-CoV-2 pandemic is the search for potential antiviral compounds targeting viral proteins with virtual screening. Approaches relying on ensemble molecular docking and machine-learning-based scoring functions have been described ***(60)***. To expand the study of proteins of higher interest using such an approach, which involves molecular docking and molecular dynamics simulations, achieving the highest possible accuracy of predicted models cannot be overlooked, despite the longer times it will demand. Additionally, it is possible that AlphaFold2, once publicly available, may reduce the need for sequence partitioning, described in **Step 6** in **Section 3.2**. However, CASP14 overall results indicate that AlphaFold2 did require human input in some edge cases and nearly one-third of its predictions did not achieve comparable quality to experimental structures. Also, other methods did outperform AlphaFold2 in a few cases. Therefore, preprocessing sequences may still be valuable to rapidly identify and dissect edge cases.

Notes from Section 3.3

26. This molecular evolution protocol is based on our published analysis ***(61)*** and the pipeline we provide above includes steps leading to improved signal to noise ratio. In addition, because bioinformatic expertise may differ among readers, we provide steps to perform this in widely used software such as Microsoft Excel^{TM}.

27. If the number of aligned sequences is greater than 3,000, work with smaller batches if using a standard personal computer.

28. Many sequence alignment viewers including CLC Genomics do not export the insertion character, so change "-" to "d" before uploading. Although CLC Genomics is a commercial product, the Viewer is a free resource with many options for manipulating alignments and sequences. This graphical user interface (GUI) presents sequences in a manner that makes multiple sequence alignments easy to edit.

29. Export the file as .csv for Excel or .txt for the R environment (https://cran.r-project.org/). To easily manipulate large files, we recommend the text editors BBBedit, 010Editor, or Textwrangler. Count all IAPUC codes that are not "A", "T", "G", or "C" as "N" unless the "-" character was changed to "d". If using Excel^{TM}, save the .csv file exported from CLC Genomics as a tab-delimited text file. Open it in a text editor and replace each nucleotide label by itself followed by a tab character (e.g. find "A" and replace it with "A\t"). This generates a file from which site counts can be easily calculated (*see* **Note 30**).

30. Generating site counts can be done in R or Excel^{TM}. If using R, large files are easily uploaded with the 'fread" function in the *data.table* package. In Excel^{TM}, there is a limit of 16,000 columns. Therefore, split the alignment in half prior to exporting it from CLC Genomics. Then, perform site counts, reduce to variable sites, and merge the halves once the number of sites has been reduced. A useful package for manipulating files is *plyr* in R using the "join_all" function that allows for defining a common feature to join files. In Excel^{TM}, the VLOOKUP function (see Microsoft manual) can be used to merge files, but with large files this may overload a less powerful desktop or laptop.

31. If using Excel^{TM}, the most straightforward means is to have the NC_045512 reference sequence in the first row and use the COUNTIF function for each column. Then subtract the total number of rows (using COUNTA). This is the total number of non-reference sites that includes "N" sites.

32. Formatting files for different software packages is a tedious task in Data Science. If using CLC Genomics Workbench, import a FASTA alignment of the haplotypes with their corresponding haplotype names and then export it as a NEXUS formatted file that PopArt will accept. Do **not** use the dash ("-") character in the haplotype

name; it generates an unreadable file in PopArt. Check the example files of PopArt for comparison (*see* PopArt manual). The median-joining algorithm can rapidly increase in execution time on a computer system if more than 1,000 haplotypes are used. The processing can be significantly accelerated if rare haplotypes are removed (e.g., less than five occurrences). This also reduces homoplasy loops that can occur from either back mutations to an ancestral state or recombination among strains. There are two other packages for generating median-joining networks. NETWORK is an older, windows-based software that is efficient, but will not consider more than 500 variable sites and its graphical interface is not as versatile as PopArt. The R package PEGAS has a median-joining algorithm implementation but due to the quadratic nature of its complexity time, execution time can grow quickly, not converging if there are any sizable number of sites or sequences. The same occurs with PopArt, but it is currently the best option.

33. In Excel^{TM}, create a file from the MetaTable in which column 1 is the GISAID accession, column 2 is the haplotype name, and column 3 is the sample date of the GISAID accession. Sort by ascending date and then use the tool "remove duplicates", using the haplotype name as the character that identifies duplicates. One date for each haplotype will be left, which will be the earliest sample registered. Now repeat this procedure, but this time sort by *descending* date. This will return the latest date for each haplotype. Use either *plyr* in R or VLOOKUP in Excel^{TM} to create a file with column 1 as the haplotype name, column 2 as the earliest date sampled, and column 3 as the latest. Subtract column 2 from column 1. This is the half-life of the haplotype.

34. Create a file from the MetaTable in which column 1 is the GISAID accession, column 2 is the haplotype name, and column 3 is the geographic region sampled. Remove all duplicates using both haplotype name and geographic region sampled. Use the COUNTIF function in Excel^{TM} to count the numbers of each haplotype remaining. This is the count of countries that the haplotype is found in.

Notes from Section 3.4

35. Iterative Random Forest (iRF) is an X-AI method that takes advantage of the benefits of the classic machine learning method Random Forest and expands upon it with a few extra steps, including Random Intersection Trees (RIT). These steps increase the explainability and add the ability to determine feature interaction.

36. iRF-LOOP uses iRF to build relationships within a feature set by determining the importance of each feature when predicting each other feature, while including the background information of the other features in the dataset.

37. The C++ version is designed for use on HPC systems. This version should be used on a large, distributed system when a desktop or laptop struggles to load or analyze the data set.

38. Optional: Generate a graphical network in a tool such as Cytoscape, treating the 'Feature' column and the 'Dependent Variable' column as node labels and the normalized feature importance values as edge weights.

39. RIT is a method to quickly and efficiently find sets of features that co-occur in a dataset by using beneficial properties innate to decision trees.

40. The FSInteract R package, which contains RIT, is loaded when the iRF R module is loaded.

41. To use the RIT R package with the C++ version of iRF requires the additional step of transforming the iRF-produced pathfile into the appropriate format by keeping only the feature names on each row from the pathfile and removing all other information.

Notes from Section 3.5

42. While this analysis focused on transcriptomic results, the same logic can be applied to results using protein IDs (Uniprot, Entrez, or RefSeq IDs) in order to glean biological understanding from proteomics data.

43. To identify significant ontologies if many genes are differentially expressed, try different combinations of gene lists to identify possibly significant phenotype/biological function/diseases associated with differentially expressed genes. Examples of different thresholds to try: top 50 or 100 upregulated genes (ranked by lowest p-value with log fold change > 0.00), top 50 or 100 downregulated genes (ranked by lowest p-value with log fold change < 0.00), top 50 or 100 differentially expressed genes (combined upregulated and downregulated genes). Thresholds for significant differential expression can also be adjusted ($p < 0.05$, $p < 0.01$, etc.).

44. Integration and interpretation of results are crucial for making conclusions about the molecular basis of pathology. While including more networks and ontologies can help facilitate this process, human input is necessary to establish new connections. In this respect, having multiple individuals participate in the integration process with varied biological backgrounds can result in a more holistic understanding of the viral pathology.

45. Only tissue-specific data sets with greater than 100 individual samples should be used for the application of iRF-LOOP to ensure enough variance exists within the data set.

46. The question is raised as to whether using a PEN as opposed to using a traditional correlation network is necessary. PENs generated from iRF-LOOP have previously illustrated a greater likelihood of finding more biologically relevant Gene Ontology annotation edges than correlation networks ***(9)***.

47. While this protocol uses the R programming environment, Python also implements igraph, which can be used to build multiplex networks.

48. If necessary to convert gene or protein IDs for DrugBank or other networks, using tools such as gProfiler (https://biit.cs.ut.ee/gprofiler/convert) and Biomart (http://www.biomart.org/) can be used.

5. Conclusions and future directions

Here we provide a series of protocols that we recommend be used in an integrative approach to understand the current, expanding SARS-CoV-2 pandemic. This series of methods should provide a framework for rapid deployment when the next pandemic begins to emerge.

We are continually improving this series and recommend several additions. For example, the GISAID database has been a critical and necessary tool to combat the current pandemic. Continued support from the global scientific community can ensure that this becomes a central repository for all virus data that can expand to include sex, age, days since symptom onset, list of specific symptoms, zip code-equivalent, a rank of severity on a scale 1-10, and full nucleotide sequence data, preferably from long-read sequencing technology. We also recommend improved graph traversal strategies for our data integration approaches. While algorithms such as breadth-first search can help identify closely-related genes that contribute to pathology, more advanced techniques may also be useful for understanding the mechanism of COVID-19 disease. One such strategy that may prove useful is the guilt-by-association algorithm Random Walk with Restart (RWR), a graph traversal method compatible with multiplex networks. We have recently developed a new version of this approach to integrate it into this workflow. Prior versions of this technique have been successfully used to discover biologically-relevant genes contributing to human pathologies ***(62, 63)***, and an open-source R package is available ***(62)***. Moreover, in addition to using network-based analyses on homogeneous networks, generating heterogeneous networks (e.g., combining gene-to-gene networks with gene-to-drug target and host_gene-to-viral_gene networks) may facilitate biological understanding by reducing the required human input.

A critical component that we are currently implementing is the extraction of relevant information from longitudinal electronic health record (EHR) data at a national level in the Veterans Affairs Healthcare System This rich data comprises comprehensive and highly diverse data on the patient's health such as drug prescription fills and refills, diagnostic and procedure codes, laboratory test results, patient demographic data, open field progress notes, among others. The depth and breadth of these are well suited for the use of advanced X-AI tools such as iRF and RIT to identify important signals related to COVID-19 prognosis. For example, iRF and RIT can be used to identify FDA-approved drugs for immediate repurposing in phase 2 and 3 clinical trials and those that may worsen disease-related symptoms or increase the probability of mortality. Further, iRF and RIT are a powerful combination of tools for identifying comorbidities, vital signs, and critical laboratory abnormalities that increase the risk of death in COVID-19 positive patients, providing means for clinicians to triage cases. Similarly, these methods will be critical in both characterizing the Long Covid Phenotype in survivors of the disease, also known as Long Hauler Syndrome, and determining who is at greatest risk of long term complications of COVID-19.

Finally, environmental factors also contribute to the spread of SARS-CoV-2 and COVID-19 severity. As with large scale EHR data, global-scale environmental data are available for integration into this workflow. For example, iRF predictive models can be created at county-level granularity using environmental, demographic, and other features to predict disease outcomes. Application of RIT to these results can determine which sets of features interact, even across feature types (i.e., an interaction between an environmental and demographic feature). Together, these described methods allow for a comprehensive systems biology view of an ongoing pandemic and can help guide clinical practices and provide the scientific community with novel hypotheses to be tested, shortening the time required to find solutions.

Acknowledgments

This research funded by the Laboratory Directed Research and Development and Seed funding programs of Oak Ridge National Laboratory (LOIS:10074,10124) and the DOE Office of Science through the National Virtual Biotechnology Laboratory (NVBL), a consortium of DOE national laboratories focused on response to COVID-19, with funding provided by the Coronavirus CARES Act. This research used resources of the Oak Ridge Leadership Computing Facility (OLCF) and the Compute and Data Environment for Science (CADES) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. Figures generated with Biorender and VMD.

Author Contributions:

Erica Prates (Conceptualization, Investigation, Methodology, Visualization, Writing - original draft, review & editing), **Michael Garvin** (Conceptualization, Investigation, Methodology, Visualization, Writing – original draft, review & editing), **Piet Jones** (Investigation, Methodology, Writing – original draft, review & editing, visualization), **J. Izaak Miller** (Writing - original draft, Writing - review & editing), **Kyle A. Sullivan** (Investigation, Writing - original draft, Writing -

review & editing), **Ashley Cliff** (Investigation, Methodology, Writing - original draft, Writing - review & editing), **Joao Gabriel Felipe Machado Gazolla** (Investigation, Methodology, Visualization, Writing - original draft, Writing - review & editing), **Manesh Shah** (Investigation, Methodology, Writing - original draft, review & editing), **Angelica M. Walker** (Investigation, Methodology, Writing – original draft, review & editing), **Matthew Lane** (Investigation, Methodology, Writing – original draft, review & editing), **Christopher Rentsch** (Investigation, Methodology, Writing - original draft, review & editing), **Amy Justice** (Investigation, Methodology, Writing - original draft, review & editing), **Mirko Pavicic** (Writing - original draft, review & editing), **Jonathon Romero** (Investigation, Methodology, Writing – original draft, review & editing), **Daniel Jacobson** (Conceptualization, Formal Analysis, Funding acquisition, Methodology, Supervision, Writing - original draft, Writing - review & editing)

6. References

1. Garcia BJ, Labbé JL, Jones P, et al (2018) Phytobiome and Transcriptional Adaptation of Populus deltoides to Acute Progressive Drought and Cyclic Drought. Phytobiomes Journal 2:249–260
2. Wang Q, Zhang Y, Wu L, et al (2020) Structural and Functional Basis of SARS-CoV-2 Entry by Using Human ACE2. Cell 181:894–904.e9
3. Prates ET, Garvin MR, Pavicic M, et al (2020) Potential pathogenicity determinants identified from structural proteomics of SARS-CoV and SARS-CoV-2. Mol Biol Evol
4. Wang D, Fang L, Shi Y, et al (2016) Porcine Epidemic Diarrhea Virus 3C-Like Protease Regulates Its Interferon Antagonism by Cleaving NEMO. J Virol 90:2090–2101
5. Lampe J, Wenzel J, Müller-Fielitz H, et al The SARS-CoV-2 main protease Mpro causes microvascular brain pathology by cleaving NEMO in brain endothelial cells.
6. Garvin MR, T Prates E, Pavicic M, et al (2020) Potentially adaptive SARS-CoV-2 mutations discovered with novel spatiotemporal and explainable AI models. Genome Biol 21:304
7. Zhang M, Case DA, and Peng JW (2018) Propagated Perturbations from a Peripheral Mutation Show Interactions Supporting WW Domain Thermostability. Structure 26:1474–1485.e5
8. Basu S, Kumbier K, Brown JB, et al (2018) Iterative random forests to discover predictive and stable high-order interactions. Proc Natl Acad Sci U S A 115:1943–1948
9. Cliff A, Romero J, Kainer D, et al (2019) A High-Performance Computing Implementation of Iterative Random Forest for the Creation of Predictive Expression Networks. Genes 10
10. Shah R and Meinhausen N (2014) Random intersection trees. J Mach Learn Res 15
11. Hwang S, Kim CY, Yang S, et al (2019) HumanNet v2: Human gene networks for disease research. Nucleic Acids Res 47:D573–D580
12. The GTEx Consortium (2015) The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans. Science 348:648–660
13. Li H and Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25:1754–1760
14. Dobin A, Davis CA, Schlesinger F, et al (2013) STAR: ultrafast universal RNA-seq aligner. Bioinformatics 29:15–21
15. NCBI SARS-CoV-2 Resources, https://www.ncbi.nlm.nih.gov/sars-cov-2/
16. Madeira F, Park YM, Lee J, et al (2019) The EMBL-EBI search and sequence analysis tools APIs in 2019. Nucleic Acids Res 47:W636–W641
17. Yuan S, Peng L, Park JJ, et al (2020) Nonstructural Protein 1 of SARS-CoV-2 Is a Potent Pathogenicity Factor Redirecting Host Protein Synthesis Machinery toward Viral RNA. Mol Cell 80:1055–1066.e6
18. UniProt Consortium (2019) UniProt: a worldwide hub of protein knowledge. Nucleic Acids Res 47:D506–D515
19. Almagro Armenteros JJ, Tsirigos KD, Sønderby CK, et al (2019) SignalP 5.0 improves signal peptide predictions using deep neural networks. Nat Biotechnol 37:420–423
20. Hanson J, Paliwal KK, Litfin T, et al (2019) SPOT-Disorder2: Improved Protein Intrinsic

Disorder Prediction by Ensembled Deep Learning. Genomics Proteomics Bioinformatics 17:645–656

21. Nielsen JT and Mulder FAA (2019) Quality and bias of protein disorder predictors. Sci Rep 9:5137
22. El-Gebali S, Mistry J, Bateman A, et al (2019) The Pfam protein families database in 2019. Nucleic Acids Res 47:D427–D432
23. Krogh A, Larsson B, Heijne G von, et al (2001) Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. J Mol Biol 305:567–580
24. Altschul SF, Gish W, Miller W, et al (1990) Basic local alignment search tool. J Mol Biol 215:403–410
25. Fung TS and Liu DX (2018) Post-translational modifications of coronavirus proteins: roles and function. Future Virol 13:405–430
26. SARS-CoV-2 conserved domains and 3D structures, https://www.ncbi.nlm.nih.gov/Structure/SARS-CoV-2.html
27. Computational predictions of protein structures associated with COVID-19, https://deepmind.com/research/open-source/computational-predictions-of-protein-structures-associated-with-COVID-19
28. Huang P-S, Ban Y-EA, Richter F, et al (2011) RosettaRemodel: a generalized framework for flexible backbone protein design. PLoS One 6:e24109
29. Kuhlman B and Baker D (2000) Native protein sequences are close to optimal for their structures. Proc Natl Acad Sci U S A 97:10383–10388
30. Yang J, Yan R, Roy A, et al (2015) The I-TASSER Suite: protein structure and function prediction. Nat Methods 12:7–8
31. Yang J, Anishchenko I, Park H, et al (2020) Improved protein structure prediction using predicted interresidue orientations. Proc Natl Acad Sci U S A 117:1496–1503
32. Roberts E, Eargle J, Wright D, et al (2006) MultiSeq: unifying sequence and structure data for evolutionary analysis. BMC Bioinformatics 7:382
33. Humphrey W, Dalke A, and Schulten K (1996) VMD: visual molecular dynamics. J Mol Graph 14:33–8, 27–8
34. Martínez L, Andreani R, and Martínez JM (2007) Convergent algorithms for protein structural alignment. BMC Bioinformatics 8:306
35. Koehl P and Levitt M (1999) Structure-based conformational preferences of amino acids. Proc Natl Acad Sci U S A 96:12524–12529
36. Katoh K, Misawa K, Kuma K-I, et al (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic Acids Res 30:3059–3066
37. Leigh JW and Bryant D (2015) popart: full-feature software for haplotype network construction. Methods Ecol Evol 6:1110–1116
38. Shannon P, Markiel A, Ozier O, et al (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13:2498–2504
39. Leinonen R, Sugawara H, Shumway M, et al (2011) The sequence read archive. Nucleic Acids Res 39:D19–21
40. Robinson MD, McCarthy DJ, and Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26:139–140
41. Love MI, Huber W, and Anders S (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol 15:550
42. Luo W and Brouwer C (2013) Pathview: an R/Bioconductor package for pathway-based

data integration and visualization. Bioinformatics 29:1830–1831

43. Luo W, Pant G, Bhavnasi YK, et al (2017) Pathview Web: user friendly pathway visualization and data integration. Nucleic Acids Res 45:W501–W508
44. Chen J, Bardes EE, Aronow BJ, et al (2009) ToppGene Suite for gene list enrichment analysis and candidate gene prioritization. Nucleic Acids Res 37:W305–11
45. Mi H, Ebert D, Muruganujan A, et al (2021) PANTHER version 16: a revised family classification, tree-based classification tool, enhancer regions and extensive API. Nucleic Acids Res 49:D394–D403
46. Kanehisa M, Furumichi M, Tanabe M, et al (2017) KEGG: new perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Res 45:D353–D361
47. Fabregat A, Korninger F, Viteri G, et al (2018) Reactome graph database: Efficient access to complex pathway data. PLoS Comput Biol 14:1–13
48. Liberzon A, Birger C, Thorvaldsdóttir H, et al (2015) The Molecular Signatures Database (MSigDB) hallmark gene set collection. Cell Syst 1:417–425
49. Wishart DS, Feunang YD, Guo AC, et al (2018) DrugBank 5.0: a major update to the DrugBank database for 2018. Nucleic Acids Res 46:D1074–D1082
50. Szklarczyk D, Gable AL, Lyon D, et al (2018) STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic Acids Res 47:D607–D613
51. Gordon DE, Jang GM, Bouhaddou M, et al (2020) A SARS-CoV-2 protein interaction map reveals targets for drug repurposing. Nature 583:459–468
52. Nowotka MM, Gaulton A, Mendez D, et al (2017) Using ChEMBL web services for building applications and data processing workflows relevant to drug discovery. Expert Opin Drug Discov 12:757–767
53. Ewing B and Green P (1998) Base-calling of automated sequencer traces using phred. II. Error probabilities. Genome Res 8:186–194
54. Elbe S and Buckland-Merrett G (2017) Data, disease and diplomacy: GISAID's innovative contribution to global health. Glob Chall 1:33–46
55. Wu F, Zhao S, Yu B, et al (2020) A new coronavirus associated with human respiratory disease in China. Nature 579:265–269
56. Qi Y, Cheng X, Han W, et al (2014) CHARMM-GUI PACE CG Builder for Solution, Micelle, and Bilayer Coarse-Grained Simulations. J Chem Inf Model 54:1003–1009
57. Zheng W, Zhang C, Li Y, et al Folding non-homology proteins by coupling deep-learning contact maps with I-TASSER assembly simulations.
58. Heo L, Arbour CF, and Feig M (2019) Driven to near-experimental accuracy by refinement via molecular dynamics simulations. Proteins 87:1263–1275
59. Rodrigues CHM, Pires DEV, and Ascher DB (2018) DynaMut: predicting the impact of mutations on protein conformation, flexibility and stability. Nucleic Acids Res 46:W350–W355
60. Batra R, Chan H, Kamath G, et al (2020) Screening of Therapeutic Agents for COVID-19 Using Machine Learning and Ensemble Docking Studies. J Phys Chem Lett 11:7058–7065
61. Garvin MR, Prates ET, Pavicic M, et al (2020) Potentially adaptive SARS-CoV-2 mutations discovered with novel spatiotemporal and explainable-AI models. Genome Biol in press
62. Valdeolivas A, Tichit L, Navarro C, et al (2019) Random walk with restart on multiplex and heterogeneous biological networks. Bioinformatics 35:497–505
63. Li L, Wang YS, An L, et al (2017) A network-based method using a random walk with

restart algorithm and screening tests to identify novel genes associated with Menière’s disease. PLoS One 12:1–19