Novel Coronaviruses Discovery in Bat

with an Innovative Bioinformatics Workflow

by

Tianchen Mu

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved October 2023 by the
Graduate Supervisory Committee:

Efrem Lim, Co-Chair
Kookjin Lee, Co-Chair
Yunro Chung

ARIZONA STATE UNIVERSITY

December 2023

ABSTRACT

This project analyzed the sequencing results of 230 bat samples to investigate novel Coronaviruses (CoVs) appearance. A bioinformatics workflow solution was developed to process the Next-Generation Sequencing (NGS) data to identify novel CoV genomes. A parallel computing scheme was implemented to enhance performance. Among the 230 bat samples, 14 samples previously tested positive for CoV appearance by a pan-CoV quantitative polymerase chain reaction (qPCR). The Illumina NGS techniques are used to generate the shotgun readings. With the newly developed bioinformatics pipeline, the sequencing reads from each bat sample, and a positive control sample were quality controlled and assembled to generate longer viral contigs. They then went through a Basic Local Alignment Search Tool X (BLASTx) query against a customized CoV database from the National Center for Biotechnology Information (NCBI) databases. After further filtering with BLASTx and megaBLAST against the NCBI nucleotide collection (nr/nt) database, the confirmed CoV contigs were used to build bootstrapped phylogenetic trees with several representative Alpha, Beta, and Gamma-CoV genomes. Two bat samples contained potentially novel CoV fragments corresponding to the Open Reading Frame 1ab (ORF1ab), ORF7, and Nucleocapsid (N) gene regions. The phylogenetic trees showed that the fragments are Alpha-CoVs, which are closely related to Eptesicus Bat Coronavirus, Pipistrellus Bat Coronavirus, and Tadarida Brasiliensis Bat Alphacoronavirus 1.

DEDICATION

Thank you to my family for always being by my side.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Dr. Lim for the opportunity to work on this research project and for his excellent guidance and support throughout my studies. I also sincerely thank Dr. Lee and Dr. Chung for their valuable guidance, assistance, and encouragement and for taking the time to serve on my committee. I would also like to appreciate the work shared by Dr. Daniel Becker's lab at the Department of Biology, University of Oklahoma.

I would like to extend my gratitude to my co-workers at Lim Lab for their kind help and patience throughout my time in the lab, especially to the lab members who processed the bat samples, Emily and Rabia, for helping me with the workflow scripts.

I would like to thank my academic advising team, the School of Computing and Augmented Intelligence, Ira. A. Fulton School of Engineering, the Biomedical Informatics Department at the College of Health Solutions, Arizona State University (ASU) Research Computing, and the Biodesign Institute for the opportunity to work on my degrees and their support throughout my studies.

As a student of ASU, I acknowledge that the Tempe campus sits on the ancestral homelands of those American Indian tribes that have inhabited this place for centuries, including the Akimel O'odham (Pima) and Pee Posh (Maricopa) peoples.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Overview

This project conducted an extensive analysis of the sequencing results derived from 230 bat samples (Becker et al., 2022), aiming to explore the emergence of novel Coronaviruses (CoVs). To facilitate this investigation, a bioinformatics workflow solution was meticulously developed. This solution streamlined the processing of Next-Generation Sequencing (NGS) (Behjati & Tarpey, 2013) data, enabling the identification of previously unknown CoV genomes. A parallel computing scheme for the proposed workflow was developed to boost computational efficiency.

Within the collection of 230 bat samples, a subset of 14 had already tested positive for the presence of CoVs using a pan-CoV qPCR (Becker et al., 2022). Leveraging Illumina's cutting-edge NGS techniques (Illumina, 2023), the project harnessed shotgun readings to generate comprehensive datasets for analysis. The core of the analysis rested on the innovative bioinformatics workflow that was conceived. The workflow is adapted from VirusSeeker (Zhao et al., 2017). Each bat sample's sequencing reads, alongside the reads from a positive control sample, underwent an in-house quality control procedure. Subsequently, these reads were assembled to construct elongated viral contigs with metaSPAdes (Nurk et al., 2017) and coronaSPAdes (Meleshko et al., 2021). These contigs were then subjected to a BLASTx (Camacho et al., 2009) search against a custom-built CoV protein database derived from NCBI Protein (National Library of Medicine (US) [NLM], 1988) and Identical Protein Groups (IPG) databases (NLM, n.d.-b). Further refinement ensued with additional filtering rounds using BLASTx and

megaBLAST (Morgulis et al., 2008) against the extensive NCBI nucleotide collection (nr/nt) database (1988). This intricate screening process ultimately led to the identification of confirmed coronavirus contigs.

These fragments exhibited notable matches within the ORF1ab, ORF7, and N gene regions. The bootstrapped phylogenetic trees were constructed with the confirmed CoV contigs and the representative genomes from the Alpha, Beta, and Gamma-CoV groups, enabled the visualization of evolutionary relationships. The resulting phylogenetic trees showcased that these virus fragments belong to the Alpha-CoV group, demonstrating close affinities with known viruses such as *Eptesicus Bat Coronavirus*, *Pipistrellus Bat Coronavirus*, and *Tadarida Brasiliensis Bat Alphacoronavirus 1*.

In summary, this project comprehensively analyzed sequencing data derived from bat samples to uncover novel CoVs. The novel bioinformatics pipeline, parallel computing, and sophisticated analytical techniques collectively played a pivotal role in unraveling the genetic makeup of these viruses and understanding their evolutionary context within the broader CoV family.

Statement of the Problem

Despite existing knowledge about CoVs, the emergence of novel strains in bat populations remains a concern. The need to effectively identify and characterize these new viruses, understand their potential for cross-species transmission and assess their evolutionary connections within the Coronavirus family necessitates using advanced sequencing techniques and the development of a robust bioinformatics pipeline.

This project employed advanced sequencing techniques and a robust bioinformatics workflow to analyze 230 bat samples, resulting in the identification of

novel Alpha-CoVs closely related to known bat coronavirus strains, shedding light on their genetic makeup and evolutionary relationships within the broader Coronavirus family.

CHAPTER 2

BACKGROUND LITERATURE

Genomic Sequencing to Characterize Viruses

Next-Generation Sequencing (NGS) (Behjati & Tarpey, 2013) technologies have accelerated the pace of viral genomics research, enabling rapid, high-throughput analyses of viral genomes. These advances have unlocked new opportunities to investigate various viruses, from well-known pathogens to newly discovered or emerging viral species. Genomic sequencing has become indispensable for identifying novel viruses, tracking viral outbreaks, and studying viral evolution at unprecedented resolution (Kulski, 2016).

NGS allows the reconstruction of viral evolutionary histories and the construction of phylogenetic trees. Researchers can track viruses' origin, spread, and diversification over time by comparing the genetic sequences of different viral isolates (Kreuze et al., 2009). This information is crucial to understand how viruses mutate to changing environments and hosts. Mutations can impact viral traits, including virulence, transmissibility, and drug resistance. Mutation tracking is particularly relevant for rapidly evolving viruses like influenza and HIV, aiding in predicting viral behavior and developing targeted interventions. Besides, looking into host factors influencing viral replication, immune evasion, and pathogenicity contributes to our understanding of disease mechanisms (Kulski, 2016).

Thus NGS plays a critical role in developing diagnostic tests and surveillance strategies. It enables the rapid identification and characterization of viral pathogens, allowing for more accurate diagnosis and timely responses to outbreaks. NGS also aids in designing effective vaccines and antiviral therapies. By targeting conserved regions of the

viral genome, researchers can develop interventions that are less likely to be affected by genetic variability (Kulski, 2016).

As genomic sequencing technologies evolve and become more accessible, their applications in virology are expanding rapidly. From uncovering the origins of zoonotic infections to understanding the genetic basis of viral tropism, these techniques offer a comprehensive toolkit for exploring the intricate world of viruses. By combining genomic sequencing with bioinformatics, epidemiology, and other disciplines, researchers can unravel the complex interactions between viruses and their hosts, ultimately contributing to better preparedness and control of viral diseases (Kulski, 2016).

Coronaviruses and Their Roles in Diseases

Coronaviruses (CoVs) represent a diverse family of viruses that have garnered significant attention due to their potential to cause severe diseases in humans and animals (Fehr & Perlman, 2015). These viruses are enveloped, single-stranded RNA viruses with a unique appearance under electron microscopy, characterized by a crown-like halo of spike proteins on their surface. While several CoVs cause mild respiratory illnesses, such as the common cold, others have demonstrated the capacity to trigger more severe and even fatal diseases, as witnessed in outbreaks like severe acute respiratory syndrome (SARS), Middle East respiratory syndrome (MERS), and the ongoing COVID-19 pandemic caused by the novel coronavirus SARS-CoV-2.

These outbreaks highlighted the critical need for a comprehensive understanding of CoVs, their biology, transmission dynamics, and the host immune response. Furthermore, the recent emergence of SARS-CoV-2 and its rapid global spread has

demonstrated the urgent need for research to elucidate the mechanisms underlying CoV-induced diseases.

CoVs primarily infect the respiratory tract but can also affect other organ systems (Wang et al., 2020). Their pathogenicity arises from a complex interplay between viral factors and host responses. The spike (S) protein of CoVs plays a crucial role in host cell entry, and its interactions with host receptors are central to determining tissue tropism and transmission efficiency. Upon infection, CoVs may trigger a range of immune responses, varying from mild inflammation to exaggerated cytokine production and immune dysregulation, contributing to disease severity.

Understanding the factors contributing to CoV emergence, transmission, and pathogenesis is vital for effective outbreak control, vaccine development, and therapeutics. Genomic sequencing has played a pivotal role in these efforts, enabling rapid identification and characterization of novel CoVs and providing insights into their evolutionary origins. This knowledge has paved the way for the development of diagnostic assays, antiviral drugs, and vaccine candidates, as seen in the case of the remarkable speed with which COVID-19 vaccines were developed (Wang et al., 2020).

As the understanding of CoVs continues to evolve, research focuses on deciphering the intricate molecular mechanisms underlying CoV infections, characterizing the host immune responses, and exploring potential therapeutic interventions. By unraveling the complexities of CoV biology and disease, researchers are better equipped to predict, prevent, and mitigate the impact of future CoV-related outbreaks (Wang et al., 2020).

Viruses in Bats

Bats display unique immune systems and behaviors contributing to their ability to host and transmit viruses (Frutos et al., 2021). While most bat-borne viruses do not directly affect humans, certain viruses have demonstrated the capacity to cross species barriers, prompting a scientific investigation into their ecology, evolution, and potential implications for public health.

The study of viruses in bats encompasses a range of research, including identifying novel viruses, characterizing their genetic makeup, and understanding how these viruses interact with their bat hosts. Some of the most well-known zoonotic viruses, such as coronaviruses like SARS and SARS-CoV-2 (the virus responsible for COVID-19), have been linked to bats. Investigating the relationship between bats and viruses provides insights into how viruses can adapt to different hosts and how spillover events occur (Frutos et al., 2021).

Research on viruses in bats contributes to our understanding of viral evolution, transmission, and the factors that drive spillover events. By identifying potential reservoirs of zoonotic viruses and studying their biology, scientists can develop strategies for early detection, surveillance, and prevention of potential disease outbreaks. As our knowledge of viruses in bats continues to grow, it enhances our ability to respond effectively to emerging infectious diseases. It underscores the importance of balancing conservation efforts with public health concerns (Frutos et al., 2021).

Viruses Discovery Workflows

In the field of virology, the discovery of new viruses has been significantly enhanced by the application of bioinformatics workflows. These computational

approaches are crucial in identifying, characterizing, and classifying novel viruses by analyzing large-scale sequencing data from diverse environmental and host samples. By harnessing the power of advanced algorithms, data integration, and comparative genomics, bioinformatics workflows enable researchers to uncover hidden viral diversity, track viral evolution, and gain insights into the potential impacts of newly identified viruses on human and animal health (Sridhar et al., 2015).

Bioinformatics workflows for virus discovery typically involve several key steps:

1. Data Collection and Preprocessing: Raw sequencing data from various sources, such as metagenomic or transcriptomic studies, are collected and preprocessed to remove noise, filter out host sequences, and prepare the data for analysis.
2. Sequence Assembly and Identification: Short sequencing reads are assembled into longer contiguous sequences (contigs) representing viral genomes. Novel viruses are often identified through sequence homology searches against existing viral databases or by identifying unique genomic features.
3. Annotation and Functional Analysis: Predicted open reading frames (ORFs) in viral genomes are annotated to infer potential functions of encoded proteins. Functional analysis provides insights into the roles of viral genes and their potential interactions with host organisms.
4. Phylogenetic Analysis: Comparative genomics and phylogenetic analysis help classify newly discovered viruses and determine their evolutionary relationships to known viral species. This step aids in understanding the origin and evolutionary history of the newly identified viruses.

The proposed workflow is adapted from VirusSeeker (Zhao et al., 2017), one of the most popular viral discovery pipelines. VirSorter (Roux et al., 2015) and VirSorter2 (Guo et al., 2021), VirFinder (Ren et al., 2017), Lazypipe (Plyusnin et al., 2020), and VIROME (Wommack et al., 2012) are other widely used pipelines. The procedures of these workflows are all similar, as summarized above. Significant differences happen in the QC steps, the choices of contigs building tools, the choice of filtering tools and processes, and the annotation step. They use different software to accomplish these steps with their rationality and test their pipeline with representative datasets.

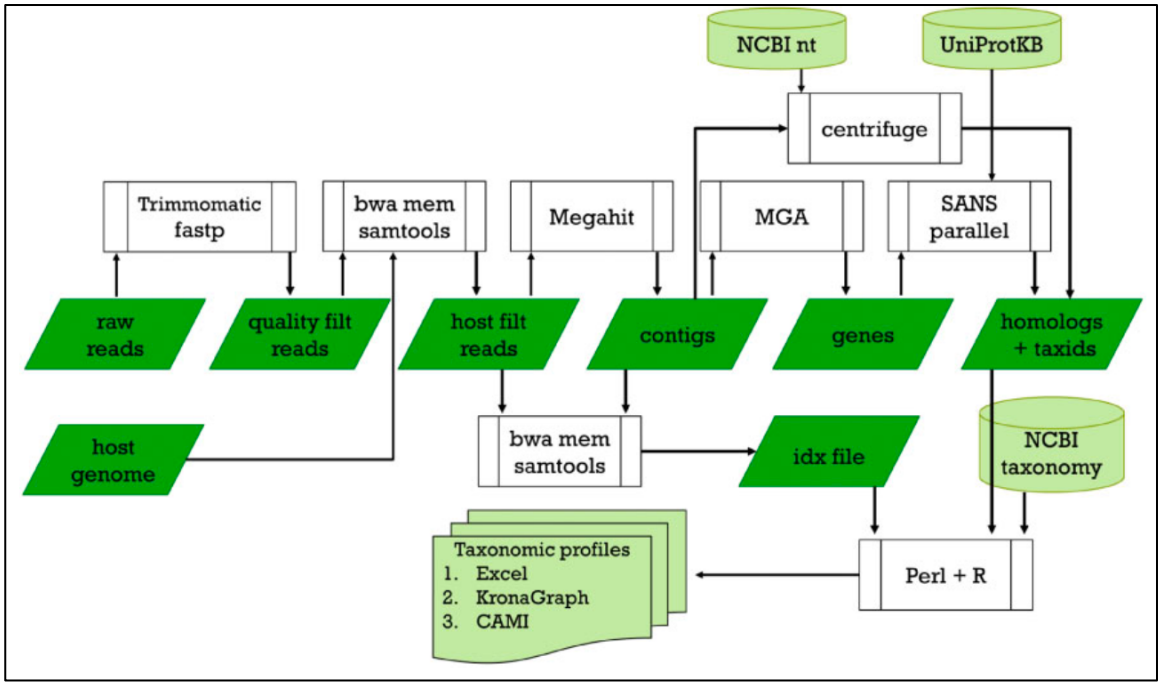

Figure 1. Lazypipe Workflow (Plyusnin et al., 2020).

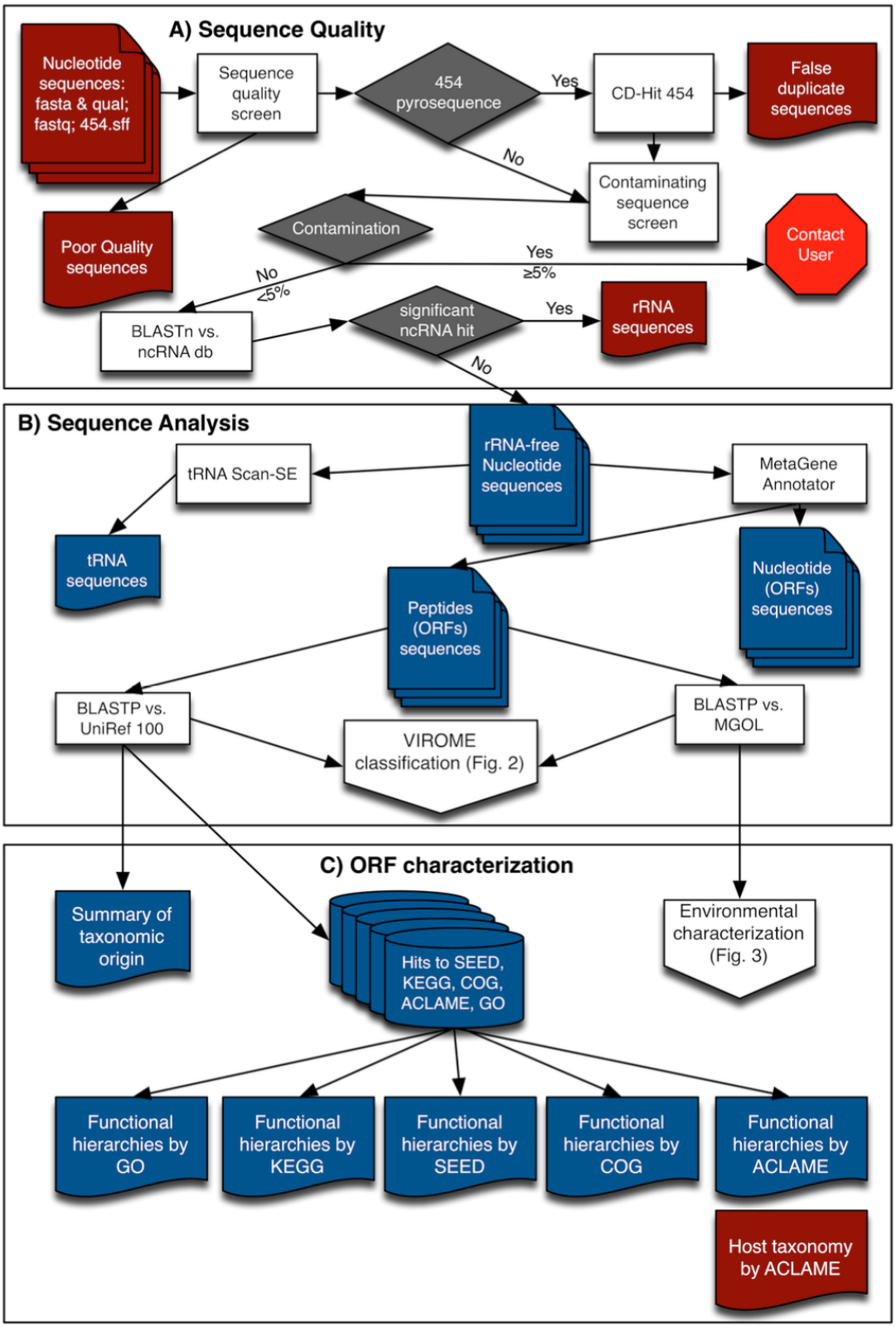

Figure 2. VIROME Workflow (Wommack et al., 2012).

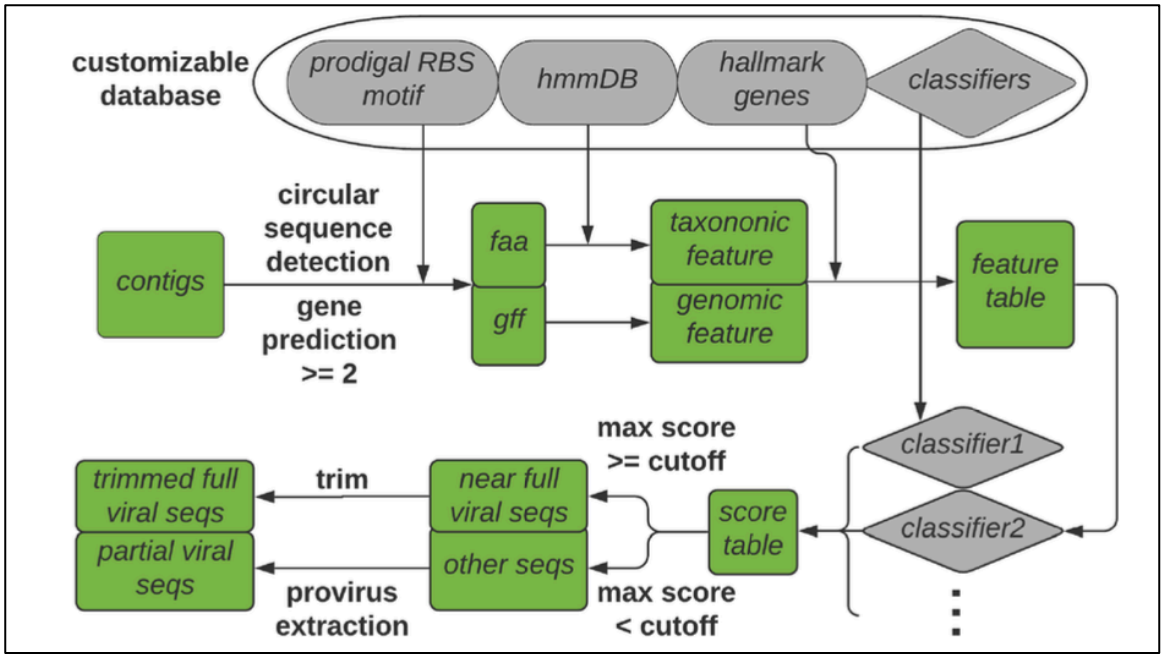

Figure 3. VirSorter2 Workflow (Guo et al., 2021).

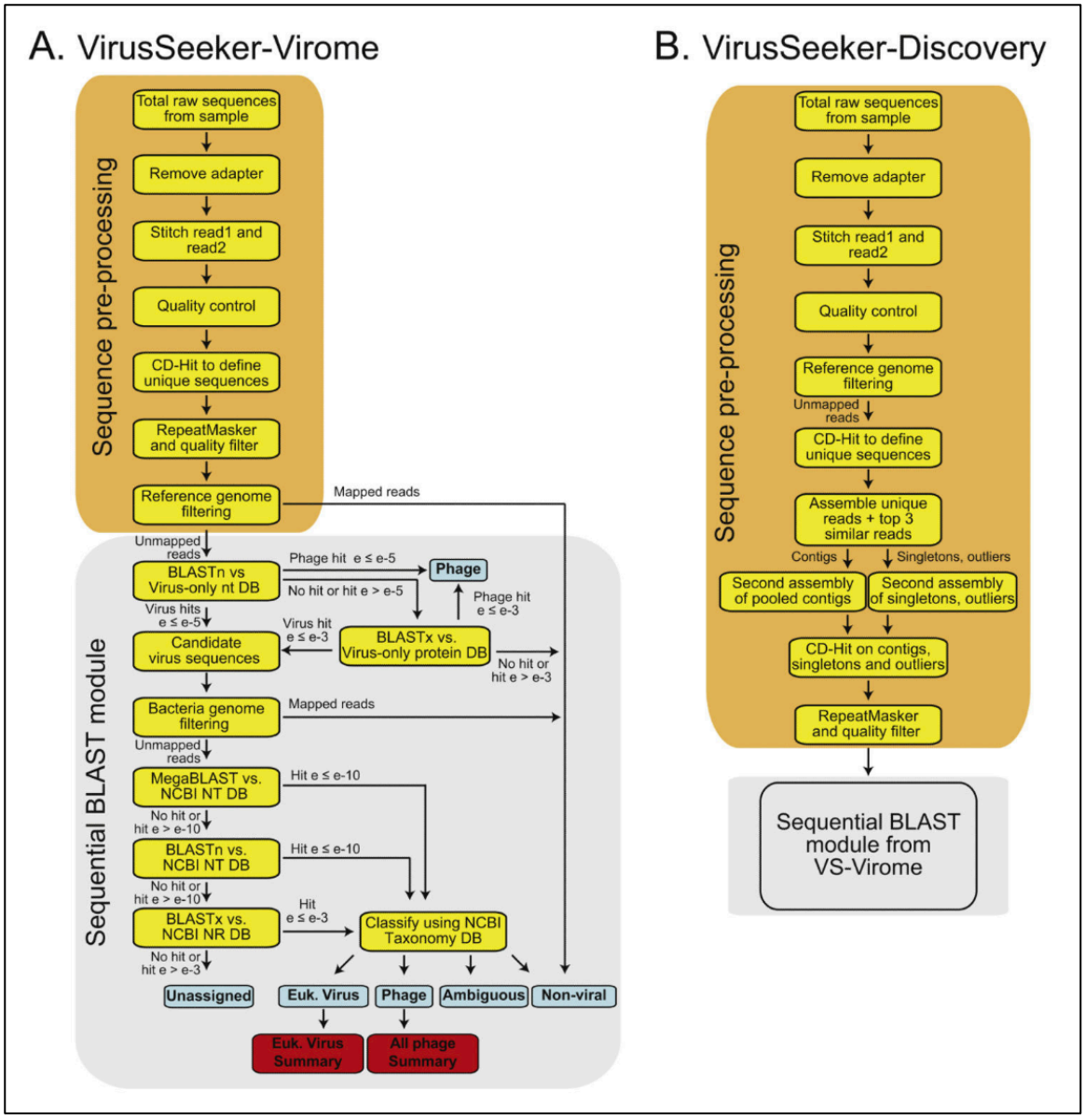

Figure 4. VirusSeeker Workflow (Zhao et al., 2017).

Integrating bioinformatics workflows with cutting-edge sequencing technologies has led to the discovery of many novel viruses across various ecosystems, including the human microbiome, wildlife populations, and environmental samples. These workflows have played a pivotal role in identifying emerging pathogens, tracking viral evolution, and providing insights into viruses' ecological and evolutionary contexts.

Potential Contributions

As mentioned above, there are many established viral discovery pipelines. There are many reasons for developing a new workflow instead of using the existing ones in this project. The proposed new workflow in the project is specially designed for the in-house sequencing workflow that is relatively new, so the QC steps are constructed with previous knowledge from the wet lab, and the software choices are proven effective in many previous projects in the lab.

The second reason is that the raw bat RNA samples are not in their best condition. The samples are transported from overseas, so the temperature, time, and other factors have affected the RNA quality. To identify the viral contigs in this kind of sample with an unknown degree of degradation, the proposed workflow implemented a sequential BLAST (Camacho et al., 2009) search module similar to VirusSeeker (Zhao et al., 2017). Inside the sequential BLAST search module, different kinds of BLAST tools are lined up, so the RNA sequences and their corresponding protein and DNA sequences are all searched, covering all the possibilities. The BLAST tools are the gold standard of genomic sequence query, while the purposed sequential model use various BLAST tools can catch all the possible targets with high accuracy. This way, as much as possible, the proposed workflow can counter the effect bought by the poor sample conditions.

The third difference of the proposed workflow is the customized database it uses. The NCBI databases (NLM, 1988) have been flooded with COVID-related sequences, so including only one SARS-CoV-2 in the customized database can dramatically shrink the search space, thus improving search accuracy and speed. Searching with a general NCBI database instead of the customized database is also tested in this project, and the result

showed less accuracy than using the customized database, which means fewer CoV fragments are identified. The proposed workflow uses less variety of software and new or newer versions of software, such as metaSPAdes (Nurk et al., 2017) and coronaSPAdes (Meleshko et al., 2021), that are easier to implement and debug.

These significant innovations of the proposed workflow led to higher efficiency and processing speed, more confidence in the results, easier usage, and better performance for the unique samples.

CHAPTER 3

METHODOLOGY

Proposed Workflow

The flowchart below shows the proposed workflow. The green fields represent the input or output data for each procedure, the deep blue fields represent the computational procedures, and the light blue fields represent the manual procedures. The workflow can be divided into three major parts: contig processing, BLAST searching (Camacho et al., 2009), and tree analysis. The contig processing part contains the quality control workflow and the contig building step. The BLAST searching part contains the customized BLAST database building and three-level filtering step with various BLAST tools. The final tree analysis contains multiple sequence alignment (MSA) (Edgar & Batzoglou, 2006) and phylogenetic tree-building steps. Each step mentioned here will be explained in detail in the following sections. This workflow covers the entire analysis process, turning the input raw sequencing results into potential novel bat CoVs contigs and their corresponding phylogenetic trees. The workflow can be performed in a traditional sequential style but can also utilize the power of parallel computing on supercomputer clusters. Since the workflow development was agile (Amoros, 2023), multiple iterations and adjustments happened before reaching the final version of the workflow. The figure and the steps described below are the final version.

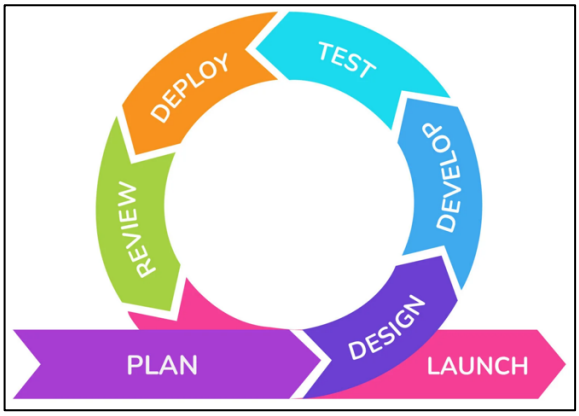

Figure 5. Agile Development.

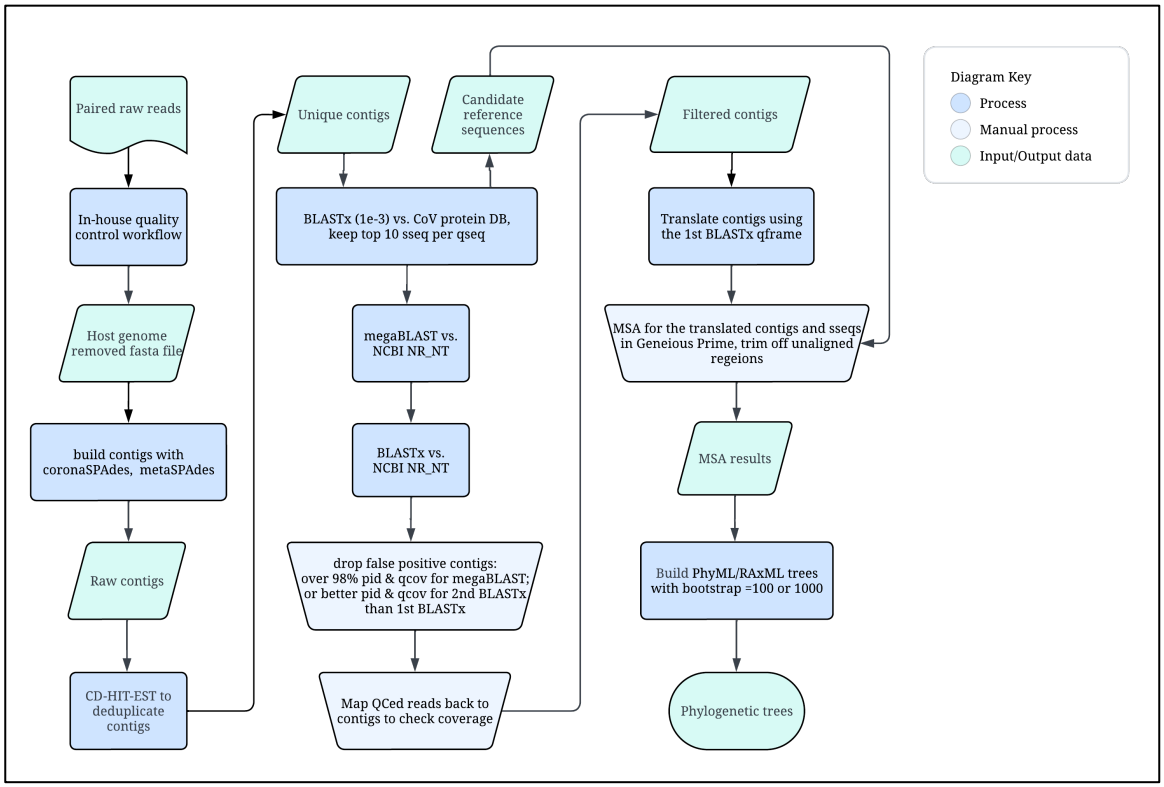

Figure 6. Purposed Bat CoVs Discovery Workflow. The computational process using automated scripts or software are in blue fields. The manual process using Geneious Prime (Geneious Prime 2023.0.4, n.d.) or other software are in light blue fields. The input and output data are in gree fields.

Quality Control

The quality control workflow (QC) was adapted from an in-house workflow in Lim Lab. This QC script is designed specifically for the sequencing process in Lim Lab, proved effective in previous studies, and should be applicable for other labs if using the standard Illumina protocols (Illumina DNA Prep Reference Guide, n.d.) and sequence-independent-amplification (SIA) sequencing technologies protocols published by Liang et al. (2020). QC aims to trim off the unwanted fragments in the raw reads.

Before going into the QC step, the quality of the raw reads files was checked with the FastQC (Babraham Bioinformatics, n.d.) tool to see if the sequencer finished the work normally. All samples passed the check with relatively high scores.

Python 3.8 and a package manager such as Conda or Mamba are needed to set up the environment. The reason for using Python 3.8 instead of the newest version is that some command line tools here do not support newer Python. The bbduk.sh script has to be downloaded from BBtools official website (Bushnell, 2022) and saved to the working directory first but not installed through the package manager. If using conda, cutadapt (version 4.1) (Martin, 2011) and BBmap (version 38.96) should be installed through the bioconda channel. The QC script can run on a local computer and process each sample's paired raw reads file sequentially. It can also run parallelly on a supercomputer cluster by modifying the original Python script into a Slurm Job Array (Slurm Workload Manager, n.d.) script with a manifest file indicating the raw read file directories.

Here is a breakdown of each step in the QC.

In the QC script, the Illumina adapters (Illumina DNA Prep Reference Guide, n.d.) and the SIA_SIB25 primers (Liang et al., 2020) of both R1 and R2, in both

directions, are trimmed off with cutadapt. This step uses the general adapter sequence files that can be downloaded from the Illumina website.

Then the second step is to trim off the low-quality fragments for both R1 and R2 in both directions with bbduck.sh. Among the parameters, trimq=20 means quality-trim to Q20 using the Phred algorithm. According to algorithm calculations, trimming to Q20 is to discard bases with a greater than 1% chance of being wrong. Next, minlength=75 is to discard reads shorter than 75bp, minavgquality=20 is to discard reads with average quality lower than Q20, and removeifeitherbad=f is to discard reads that are shorter than the minimum length 75bp but not automatically discard their corresponding paired reads. Then tpe=t means discarding the entire pair if the length of both reads in a read pair is not the same. Set overwrite=t to grant permission to overwrite files, but it will not affect the original input raw read files.

Next, the phiX sequences, Illumina's standard quality control sequences (Illumina DNA Prep Reference Guide, n.d.), are trimmed off with bbduk.sh. Set k=31 to choose the default kmer length, the window size used for searching and matching the phiX sequences. A longer kmer is stricter, and the default value is acceptable for regular jobs. Set hdist=1to choose the Maximum Hamming distance for ref kmers (subs only). The memory used in this step will be proportional to (3*k)^hdist. The Linux server in Lim Lab and the “Agave” supercomputer cluster managed by ASU Research Computing provided sufficient memory for the jobs in this project. Memory error will cause the job to end early.

Last, the host genome is trimmed off from R1 and R2 with bbmap.sh (Bushnell, 2022). Before running this step, the host genome has to be downloaded from the NCBI

Genome database (NLM, n.d.-a). According to Becker Lab (Becker, 2022), the host of the bat samples used in this project is identified as Common Vampire Bat (*Desmodus rotundus*), and the accession number of the genome on NCBI is GCF_002940915.1 (NLM, n.d.-c). Then to build the index of this reference genome, "bbmap.sh in=reads.fq ref=D.fa path=/file/location/" will write an index for D.fa into /file/location/ref/. This path was saved for the following trimming step as the path for reference genome index since building the index only once can save time and avoid the index being corrupted. The bbmap.sh script can also be downloaded from the BBtools official website. The output files from the "outu" stream are unmapped reads, which are the reads without host genome reads. The "outm" stream files are mapped reads and discarded since they match the host genome. Pairs are always kept together; if one read is mapped and the other is unmapped, both will go out. Other flags are defaulted: "minid=.95 maxindel=3 bwr=0.16 bw=12 quickmatch fast minhits=2 -Xmx64g".

Here is a summary of the reads that were trimmed off in the QC:

1. Illumina adapters, cutadapt
2. SIA_SIB25 primers, cutadapt
3. Low-quality parts, bbduk
4. Q20 (discard the reads with greater than 1% chance of being wrong)
5. Minimum length 75bp
6. phiX sequences, bbduk
7. Host bat genome, bbmap:
8. common vampire bat (*Desmodus rotundus*), GCF_002940915.1

No further trimming or mapping was performed after step 5. The reason for not doing more quality trimming or concatenating R1 and R2 together is to preserve the information as much as possible. The downstream BLAST query filtering steps can filter out false positive sequences. In the project, different ending points of the QC workflow were tested, and the five steps described above are the minimal steps, yielding more usable sequences for downstream analysis. The flag values used in this QC workflow should be adjusted for different samples and NGS workflow.

For the 14 pre-screened bat samples, all five steps were finished. For the remaining 216 samples, only step number one to four were performed in order to preserve more information and consider the possibilities of different hosts.

Contigs Building

Two tools were used for the QC-ed reads: metaSPAdes (Nurk et al., 2017) and coronaSPAdes (Meleshko et al., 2021) from the SPAdes. CoronaSPAdes are a new tool developed specifically for CoVs, adapted from the previous biosynthesisSPAdes. To run these two tools, download the corresponding Python scripts from their GitHub repository and save the files to the working directory. No flag needs to be specified for the tools other than giving the input R1, and R2 reads files, the output filename, and the number of threads used for the job. Previously members of Lim Lab found that SPAdes tools could not run on Agave due to some settings of that specific supercomputer cluster, so this step was run on the Linux servers of Lim Lab. However, the scripts for this step are also available for both local and parallel computing machines. Other contig building tools were tried but failed to generate useable results, while the SPAdes tools proven to be reliable in Lim Lab.

SPAdes uses two techniques for building contigs. The first relies on read pairs and tries to estimate the size of the gap separating contigs. The second one relies on the assembly graph: e.g. if two contigs are separated by a complex tandem repeat that cannot be resolved exactly, contigs are joined into a scaffold with a fixed gap size of 100 bp. Contigs produced by SPAdes do not contain N symbols (SPAdes 3.15.4 Manual, n.d.).

The output file of this step for each sample was a single fasta file in the output folder, called "scaffolds.fasta", containing all the contigs built from the reads, from longest to shortest. Each contig took one row in the file and was assigned a name automatically by SPAdes. For the contig name "NODE_a_length_b_cov_c," the number "a" after "NODE" is the index of the contig, number "b" is the length of this contig, and number "c" is the kmer coverage for the last (largest) k value used. Note that the kmer coverage is always lower than the read (per-base) coverage. Please refer to the SPAdes manual for details (SPAdes 3.15.4 Manual, n.d.).

For the 14 pre-screened bat samples, both coronaSPAdes (Meleshko et al., 2021) and metaSPAdes (Nurk et al., 2017) were used, and the two contigs file for one sample were merged. Because there was not much difference found between metaSPAdes and coronaSPAdes results of the 14 samples, both tools produce similar even same contigs. So only metaSPAdes were used for the remaining 216 samples to save time. The positive control sample used in this project was not a CoV, so it was processed with metaSPAdes.

And then, the contig files are deduplicated with the CD_HIT_EST (Li & Godzik, 2006) tool. It can be installed via conda in the bioconda channel. The flag values in this step deduce the contigs at 95% identity across 95% of the contig length.

Building contigs before or after filtering was tested in this project when constructing the workflow; the later BLAST results showed that building the contigs after filtering would lose more information than building the contigs beforehand. The "viral dark matter" would be lost in the filtering process if directly filtering the reads without building the contigs (Krishnamurthy & Wang, 2017).

Customized BLAST Database Building

The database produced in this step is crucial for the entire workflow. The aim is to find all the potential targets in the NCBI databases (NLM, 1988) and build a customized database from the search results. Since a protein can have multiple codons, mutations may not change the protein sequence, while similarity in protein sequences indicates an evolutionary relationship. So instead of searching directly on RNA or DNA sequences, BLASTx and protein databases can preserve more information and be more sensitive. Thus, in this project, the customized database was built with protein sequences from the NCBI Protein and IPG databases, and the following first filtering was performed with BLASTx on the RNA reads directly.

There are two strategies to query the NCBI database: scientific species names or taxa IDs. The query sentence used for the name query was:

"(Coronaviridae[ORGN] OR (Coronaviridae[ORGN] OR coronaviridae[ALL])) NOT (Severe acute respiratory syndrome-related coronavirus[ORGN] OR Severe acute respiratory syndrome-related coronavirus[ALL])".

The query sentence used for the taxa ID query was:

"(((txid11118[Organism:exp]) NOT txid2697049[Organism:exp]))".

The query sentences mean to query for all the Coronaviridae records but do not include SARS-COVID-2 records. That way, the massive amount of COVID submissions would not affect the query results, and later one single COVID record would be added to the results to build the customized database.

Using these two query styles on the two NCBI databases (NLM, 1988), four sets of query results were generated and contained different numbers of records, as shown in the figure below. The taxa ID query yielded slightly more records than the name query. Since the IPG database is the deduplicated protein database, the IPG database yielded much fewer records than the Protein database. Interestingly, every set of results has unique records, and none entirely overlapped with others.

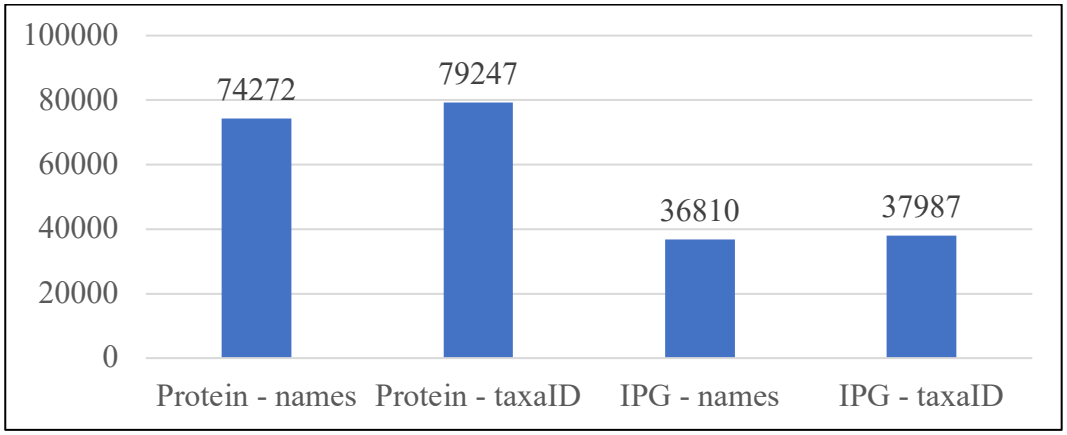

Figure 7. Downloaded CoVs Records Numbers of the Four Query Methods.

After downloading the query result in fasta format, all four files were concatenated into one single fasta file. Then this file was deduplicated with sequin, which can be installed through conda. The flag used in this step for seqkit was “rmdup -n,” aiming to find the records with the same names (ignored the case) in the file and delete the second one in the pair. Then the records with the same sequence and ID but different scientific names were also deduplicated with sequin, using flags “rmdup -s -j -P” only to

keep the positive strands. And then, a list of all the names of the records was generated for further trimming. All the records with “PDB”, “PIR”, and “PRF” in the prefix and “unnamed protein product” in the names were deleted since these sequences are often human antibodies or synthesized proteins, which had caused false positive findings in the downstream analysis. After trimming, 40725 records were maintained from the 228316 records, about 18% of the original file.

After adding one COVID record, “Wuhan-Hu-1” (NLM, n.d.-c), to the trimmed file, a single BLAST database was built with this file using makeblastdb, which can be installed through conda. Since it is a protein database, the flags used for this step were “-subtype prot -parse_seqids.” The database can be built at any location, and to use the database, move the entire folder to the working directory.

Three-Level Filtering – the 1st BLASTx

The filtering steps aim to find the potential novel CoV fragments. The first filtering uses BLASTx against the customized CoVs protein database, and this step is to find all the potential CoVs from the input file and keep as much information as possible. The other two filtering levels are for trimming off the false positive results from the first BLASTx query.

Since adding threads to a single BLAST job will not increase speed according to a benchmarking analysis (Pascal, 2014), there are multiple options to take to increase speed. First, the input query files, the contig file of each sample in this project, can be broken up into smaller files for smaller query jobs. The Biopython SeqIO module offers a script (Biopython, n.d.) to split a giant fasta file into smaller ones by iterating through all the sequence IDs in a file and send fix number of records to output files one by one. And

then, these smaller files can run BLAST jobs parallelly either by using a Python script with a subprocess or a Slurm job array (Slurm Workload Manager, n.d.) script with the corresponding manifest file. Running BLAST jobs sequentially on a local computer will be very time-consuming since the jobs must be run multiple times to find the best query parameters.

In this project, the first BLASTx query used these parameters to form a lenient query to preserve as many as possible records from the input file: "-value 1e-3 -max_hsps 1". According to the BLAST manual by NCBI (NCBI, n.d.), the e-value, Expect value (E), describes the number of hits one can expect to see by chance when searching a database of a particular size. It decreases exponentially as the Score (S) of the match increases. Essentially, the E value describes the random background noise. The higher the E-value, or the further it is to zero, the less "significant" the match is. So setting the e-value to 1e-3 would generate more query results.

Moreover, the target sequences or "sseq" were set to 10, meaning the top 10 target sequences or "hits" from the customized database were kept for each input query record. This set would later be used for phylogenetic tree-building steps as the candidate reference genome for the novel CoVs fragments.

The output format was "fmt11," the archive style, and then "fmt6," the tabular style files were drawn from the archive files. The archive files can be stored in case other information is needed later. For the "fmt6" files, these columns were chosen for later use: "qseqid sseqid evalue bitscore score pident nident length mismatch qframe qstart qend qlen qcovs sframe sstart send slen" and their meanings can be found in the BLAST manual appendices (NCBI, n.d.-b). With "qseqid", which is the sequence ID of the query

records that have hits in the customized database, the potential CoVs records can be retrieved from the input query file. The records that did not have hits in this step were discarded since they were considered not CoVs.

Three-Level Filtering – megaBLAST and the 2nd BLASTx

The query results from the previous step, the "use" of each sample, were fed into Geneious Prime (Geneious Prime 2023.0.4, n.d.). Two levels of filtering were performed with Geneious Prime built-in BLAST tools using default parameters. MegaBLAST (Morgulis et al., 2008) was performed against the NCBI nr/nt database (NLM, n.d.) to query the input records in RNA encodings and their corresponding DNA sequences against all NCBI nucleotide records. Meanwhile, the second BLASTx was performed against the same nr/nt database for the same file. For both queries, the top 100 targets or hits are kept. The criteria for dropping the false positive contigs are

(1) having none-CoV targets with over 98% pid and over 98% qcov in megaBLAST results, or

(2) having none-CoV targets with better pid and better qcov in the 2nd BLASTx results than the 1st BLASTx.

The criteria mean some none CoV targets were matched or better matched (compared to CoVs) to the query contigs, which were supposed to be CoVs. So these contigs were considered false positive results and were discarded.

Three-Level Filtering – Contigs Coverage

The final step of the filtering is to map the previously QC-ed reads back to the filtered contigs for each sample to check the overall coverage visually in Geneious Prime. Some contigs were only built from one or two reads, which may be artifacts of the

algorithm instead of actual CoV fragments. These contigs can be kept for further investigation or dropped at this step. In this project, these kinds of contigs were kept for later since the yield after filtering steps was minimal.

After checking the coverage, the contigs of each sample were concatenated together to form a single long contig without adding space or N bases in the middle, because the MSA algorithms can automatically insert spaces into the contigs during the alignment process. To concatenate the contigs of a single sample together, their positions were determined by their potential CoV gene region, “start” and “send” from the 1st BLASTx, since those two numbers indicate the contigs' positions on the candidate reference sequence. So all the contigs in a sample that belong to the same gene, such as ORF1ab or S-gene, were concatenated together to form longer contigs. Further concatenating was tried by connecting these concatenated contigs by order of a general CoV, “ORF-S-E-M-N.” Below is a figure showing the genome of SARS-CoV-2, indicating the order of the genes (Ellis et al., 2021).

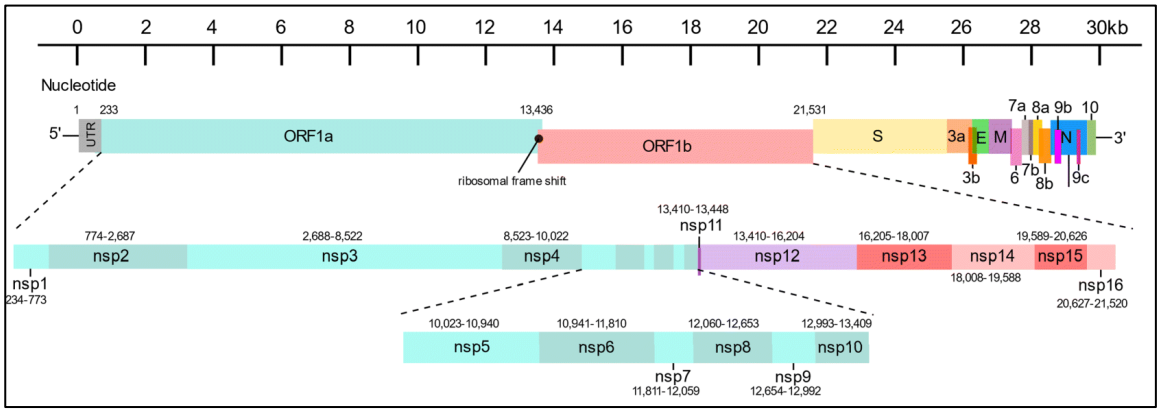

Figure 8. SARS-CoV-2 Genome Structure (Ellis et al., 2021).

Multiple Sequence Alignment

In this step, the filtered and concatenated contigs were first translated to protein sequences, then aligned to the previously found candidate reference protein sequences or the sseq with the multiple sequence alignment tools in Geneious Prime. The translation process was performed with Geneious Prime built-in tool, using the "frame" from the 1st BLASTx results as the translation frame. "N bases do not need to be added to the contig beginnings. More reference sequences were added, such as some representative bat CoVs, some infamous CoVs like Middle East Respiratory Syndrome-related CoV, and some CoVs that are very close to the candidate set.

And then Clustal Omega version 1.2.3 (Sievers et al., 2011) was used to construct the MSA since other MSA tools yielded worse alignment results when tested. The figure below shows the GUI for setting up the Clustal Omega jobs in Geneious Prime.

After the MSA, the regions on the contigs that were not aligned were trimmed off since they did not provide helpful information in the following phylogenetic tree-building steps.

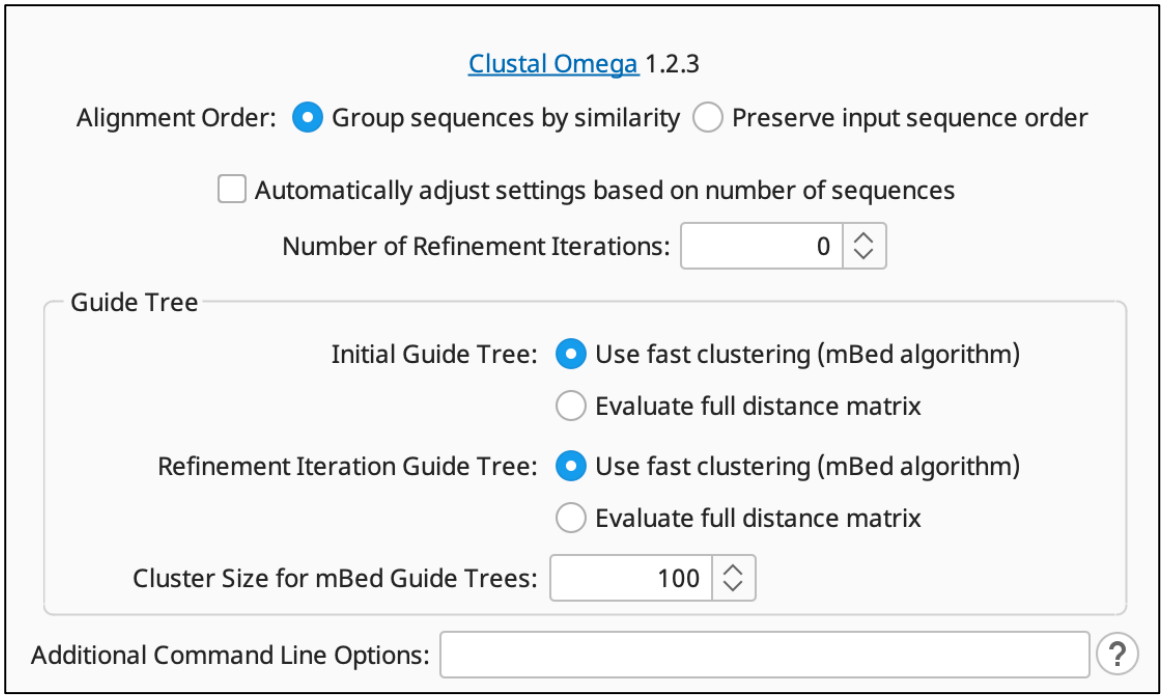

Figure 9. Clustal Omega Job Setup Window in Geneious Prime.

Phylogenetic Tree Building

In Geneious Prime (Geneious Prime 2023.0.4, n.d.), the phylogenetic trees can be built directly from the above-trimmed MSA results. Two maximum likelihood tree algorithms are used in this project. PhyML (Guindon et al., 2010) was used for contigs in the ORF1ab region, and since it cannot work with sequences having stop codons, RAxML (Stamatakis, 2006) was used for the rest of the contigs that happen to have stop codons. Bootstrap is enabled for both tree tools, and the number of bootstraps was tried at 100 and 1000. Although setting bootstrap at 100 or 1000 returned similar results for this project, the workflow recommended running bootstrap at 100 first, then after tunning or adjustment, running bootstrap at 1000 as the final run since it is very time-consuming. The following figures show the setup windows of the tree tools inside Geneious Prime.

The reason for choosing these two tree algorithms is that they are the most popular maximum likelihood methods currently (Geneious Prime, 2020). Maximum likelihood methods are very robust and thorough and have lower variance than other methods, so they are suitable for virus discovery tasks because of the vast number of unknown factors and often poor input data (Galtier & Gouy, 1998). This kind of method generates all the possible trees in the bootstrapping process, then combines all the trees to form the final output, having bootstrap values on the branches of the final tree as confidence level. Although they are CPU intensive, because of the filtering and concatenating step, there were few contigs left for tree building in this project. On the other hand, the bootstrapped tree-building process is easy to run parallelly.

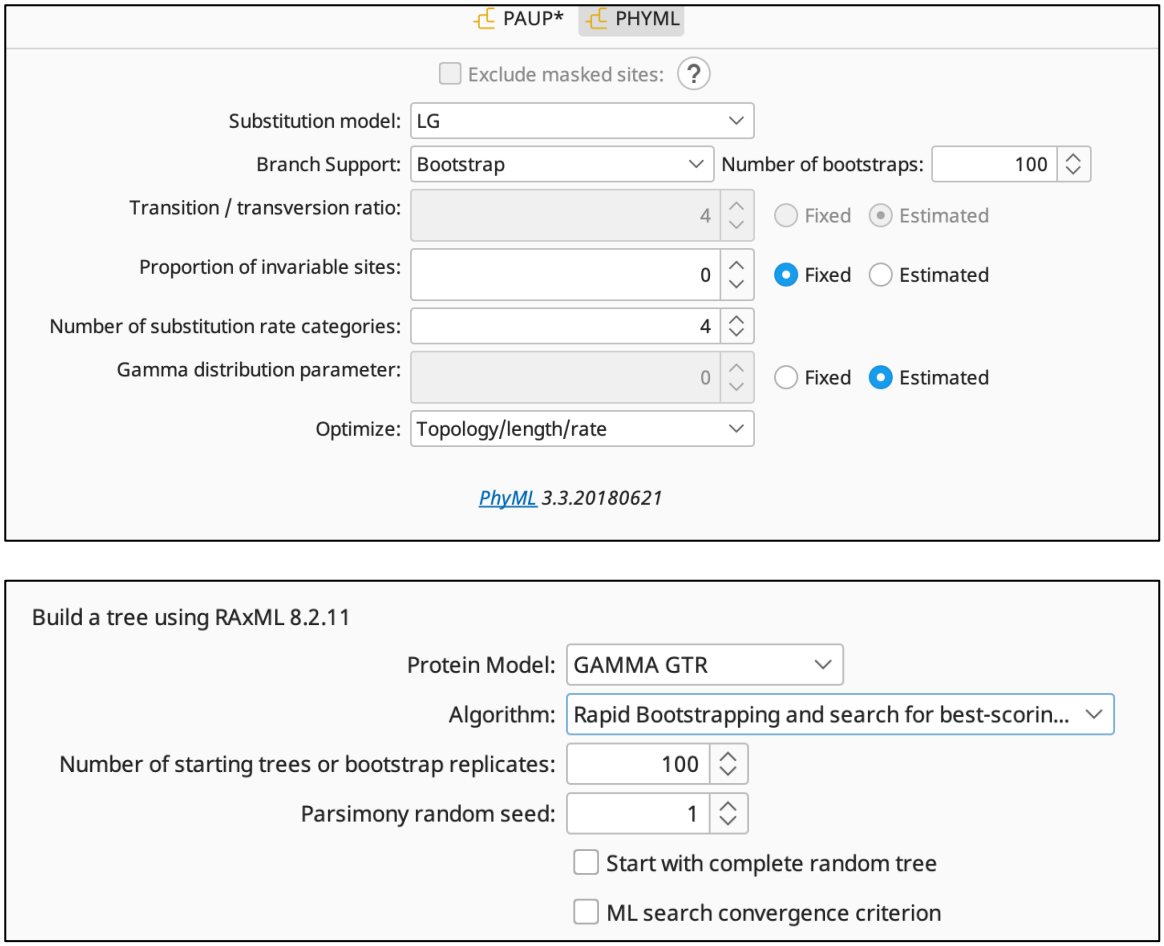

Figure 10. PhyML and RAxML Tree Jobs Setup Window in Geneious Prime. For PhyML, the substitution model chosen was LG, and the version was 3.3. For RAxML the protein model was GAMMA GTR, and the algorithm chosen was "rapid bootstrapping and search for best scoring ML tree". The "start with complete random tree" and "ML search convergence criterion" options did not affect the final tree results generated for the data used in this project, but they should be tested when using for different data.

Parallel Computing

The parallel computing techniques used in this project are two kinds, which are the built-in multithreading functionalities in the various command line tools used, and

also the Slurm Job Array (Slurm Workload Manager, n.d.) mechanism for the supercomputer cluster “Agave” managed by ASU Research Computing.

Slurm is a popular workload manager often used for supercomputer clusters. In Agave, the nodes are similar to a single local computer, and the cores in a node are like the CPUs in a local computer. A repetitive task can be broken into smaller sub-jobs and run on different nodes simultaneously by setting up a slurm job array, which utilizes a batch bash script and a manifest file. The sbatch script reads the input files' location from the manifest file and executes the command line tools for a fixed number of files on the manifest file. Each line of the manifest file will be the input file of a sub-job in the job array. The manifest file can also contain needed parameters or other information. The manifest file was made with the GNU Parallel tool (Tange, 2023) in this project, which takes input strings to generate desired amount of combination results of the strings, minimizing manual work. Once a sub-job is sent to a node and starts running, with the built-in multithreading flags of the command line tools, all the requested number of cores in this node will be allocated and used for this sub-job simultaneously.

The major time-consuming steps of this workflow are QC, BLAST queries, and the tree-building step. The jobs in these steps are highly repetitive, so it is strongly recommended to use a Slurm Job Array with supercomputer clusters to run these steps, if available. Otherwise, setting up a Python subprocess script to run jobs parallelly in some local multi-CPU servers would also be a good choice.

The codes and scripts developed for this workflow is available at Lim Lab public GitHub repository: https://github.com/ASU-Lim-Lab/BatCoVDiscovery.

CHAPTER 4

DATA ANALYSES AND RESULTS

Among the 14 pre-screened samples, one contig is potentially a novel CoV. Moreover, two of the 216 samples contain ten contigs that are potentially novel CoVs. A positive control sample also went through the entire workflow, producing expected results and proving the accountability of the workflow. A similar workflow that does not include the customized protein database querying step was performed on the 216 samples and only yielded two contigs that are potentially CoVs. Interestingly, these two contigs are included in the found ten contigs. This comparison shows the high sensitivity of the proposed workflow.

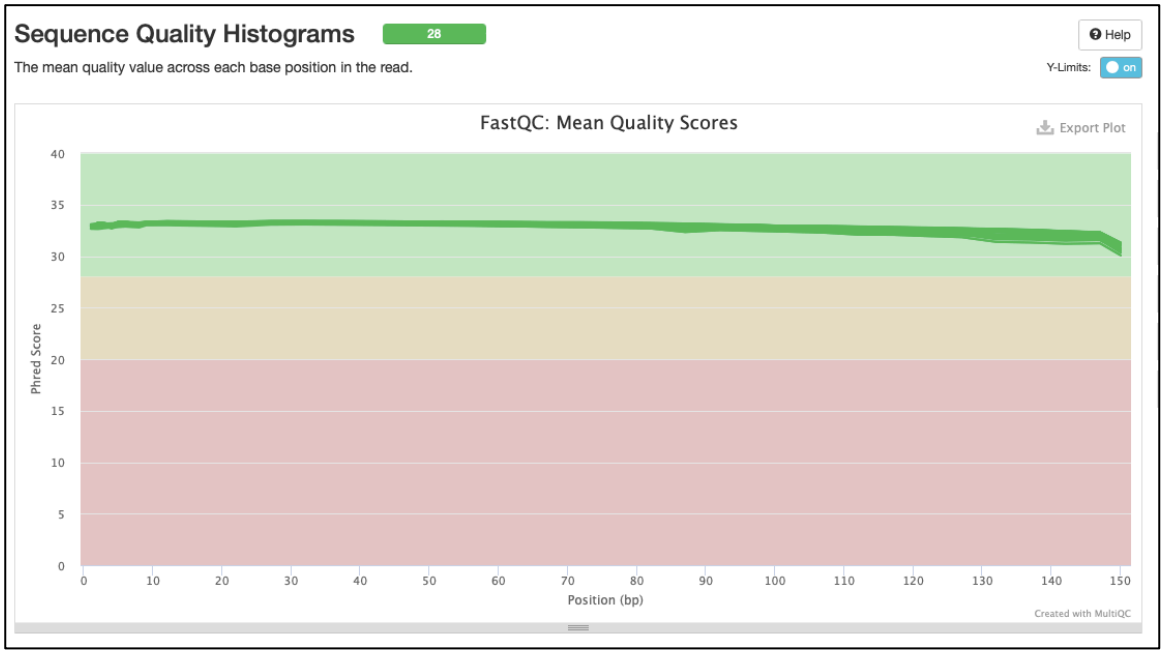

Figure 11. FastQC Result of A Random Sample. The average quality of all samples are like this, which means the raw reads files all have relatively high quality data.

Table 1

Overall Results After the First BLASTx query. All samples have passed the QC step.

Steps	Counts
Before QC	Average 35m read pairs
After QC	Average 35m read pairs
Total contigs built	Around 20k per sample
Contigs remain after the 1st BLASTx	0 to 5 for each sample

BLAST Results

For the 14 pre-screened samples, 5 samples have contigs after the first BLASTx query, but only one contig remained after the entire three-level filtering since most of them are false positive results. The only contig found in them was considered poor quality compared to other positive results. Because first, the alignment result of the contig and its candidate reference sequence is poor, and the pair-wise identity score (pident) from the first BLASTx is low. Besides, only a few candidate reference sequences (sseq) were found for this contig, which is relatively short. The cover value of this contig is less than one (usually a much larger number), and less than five raw reads are mapped to it. Besides, there are no more contigs from the sample for further concatenating or concatenation steps, so no tree was built for this contig.

sample ID	conitg_qseq_id	screen: megablast_nt/nr top 100	screen: blastx_nr top 100	sseq_id	sseq_name	evalue	pident	qcovs
72	NODE_3505_length_252_cov_67.709497	no important	60% qcov 77% pid with a bat seq	gb|AJD09601.1|		1.26E-04	31.579	68
				ref|YP_008439202.1|		2.60E-04	32.727	65
				gb|ASL24654.1|		3.77E-04	32.727	65
	NODE_1651_length_366_cov_5.157556	no important	41% qcov 70% pid with a bat seq	gb|ANZ78845.1|		8.29E-04	29.167	59
				gb|ANZ78844.1|		8.29E-04	29.167	59
73	NODE_4737_length_274_cov_0.751244	no result	no result	gb|AVM80519.1|	spike glycoprotein subunit 1, partial [Swine acute diarrhea syndrome related coronavirus]	8.12E-04	28.049	89
I91185	I91185_Contig_32	no match	matched a lot CoV	gb|UZK98260.1|		3.03E-120	80.66	52
				gb|UED13287.1|		7.67E-120	80.66	52
				gb|URD31237.1|		8.16E-120	80.66	52
				gb|URD31244.1|		8.24E-120	80.66	52

Figure 12. Contigs Examples. From top to bottom there are an example of false positive contigs, the one poor quality contig and an example of the true positive contigs that are good quality. For the false positive contigs, they generally have large e-value in the first BLASTx query, and relatively large pident & qcovs for bat sequences in the second BLASTx query results or the megaBLAST query. For the sigle contig remained for the 14-screened samples, the e-value, pident and qcovs scores are not optimal comparing to below. For the example true positive contig, it did not have any much in megaBLAST query, while had lots of CoVs hits in the second BLASTx query. The scores of its first BLASTx results are optimal.

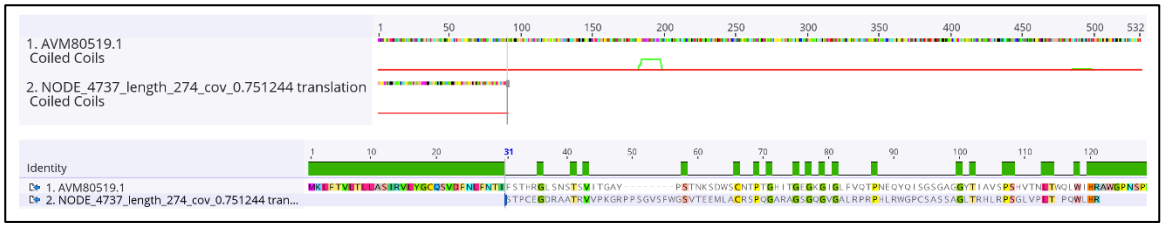

Figure 13. Alignment Result of the Poor Quality Contig of the 14-screened Samples. The contig itself is only 274bp long, which is on the short end of the other contigs. While the aligned region is even shorter. The alignment was poor as the colored regions of the first plot are the disagreements, and the colored regions of the second plot are the agreements.

Besides the 14-screened samples, the remaining 216 samples yielded totally 55 usable contigs that belong to 27 samples, from the first BLASTx query. After the entire filtering, 23 contigs remained to be true positive.

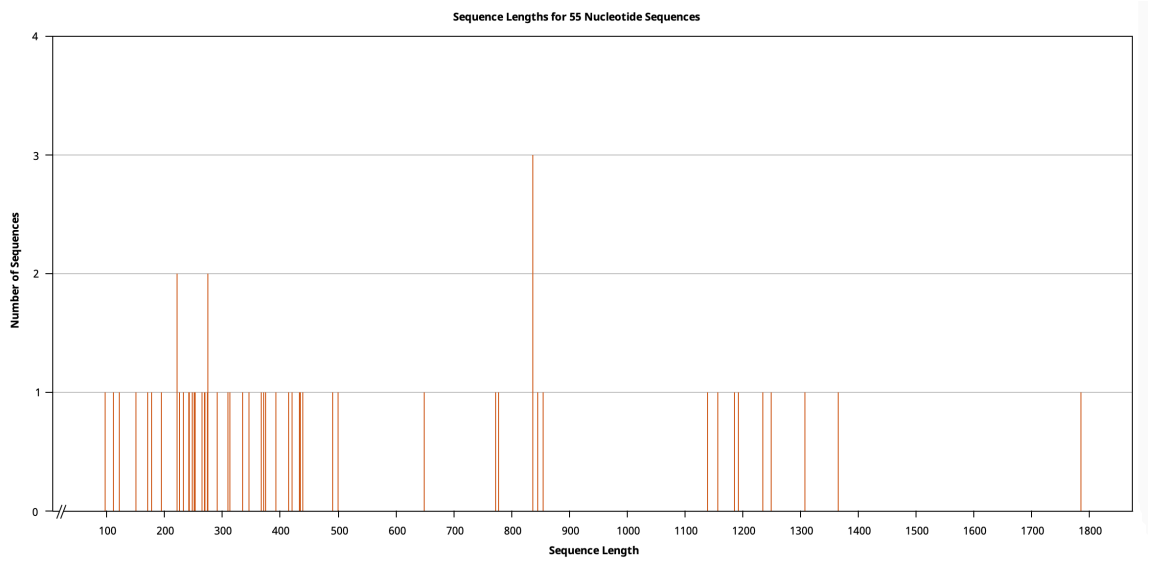

Figure 14. The Overview of the 55 Contigs Showing Various Length.

Table 2

Overview of the remaining true positive contigs from the 216 samples.

Sample ID	Contig #	Total Length (aa)	Matching Region
I91185 (BZ399)	265	124	ORF1ab Total: 1404 aa
	473	73	
	37	385	
	30	414	
	252	130	
	68	278	
	474	73 (only match 1-41)	Spike Total: 484 aa
	32	411 (only match 182-393)	
	39	379 (only match 185-352)	ORF7
Total	9 contigs	2267 aa	3 regions

Sample ID	Contig #	Length (aa)	Matching Region
I91217 (BZ773)	4010	163	N

After mapping the raw reads back to the concatenated contigs, the reads generally matched the regions that are also the 1st BLASTx hit regions on the contigs, proving the accountability of the concatenating step.

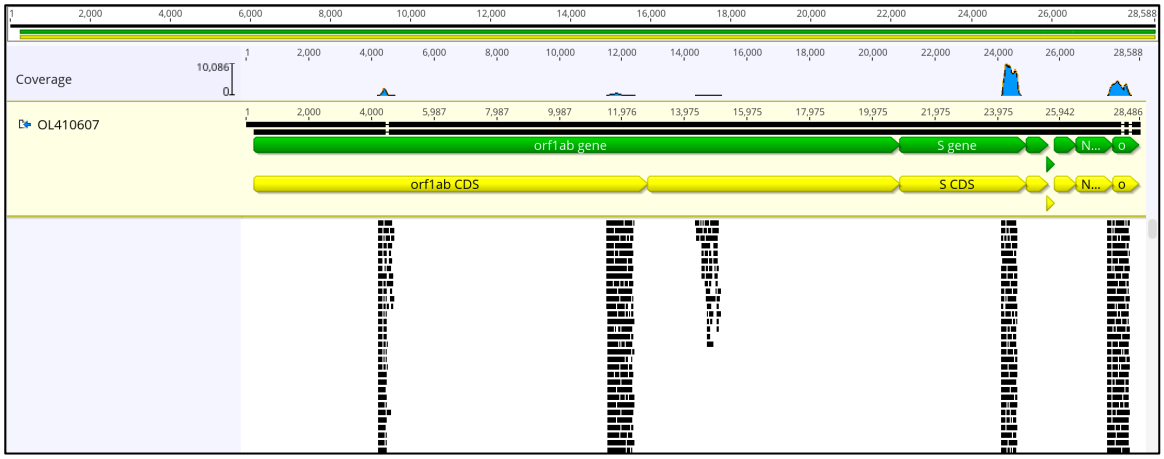

Figure 15. An Example of Mapping QC-ed Reads Back to the Concatenated Contigs. The raw reads showed in the figure is from sample I91185, and the reference genome is Eptesicus Bat Coronavirus, the most frequent target of this sample. The matching regions are similar to the first BLASTx query results. Other samples that have true positive contigs have the similar mapping patterns.

Phylogenetic Analyses

The tree-building results show that the potential novel CoV fragments are Alpha-CoVs. The trees built with different tools for the same concatenated contig are similar, and the phylogenetic relationships are similar. The results of bootstrapping at 100 and 1000 were identical, the figures below show the results with bootstrapping=100.

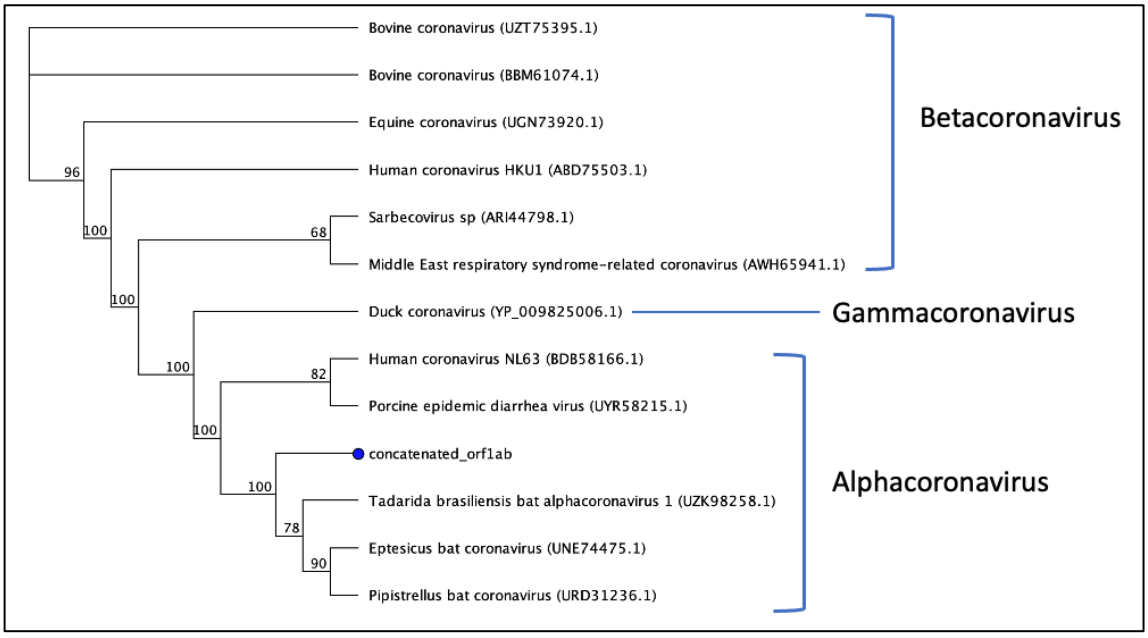

Figure 16. PhyML Tree of the Concatenated Contig at the ORF1ab Region.

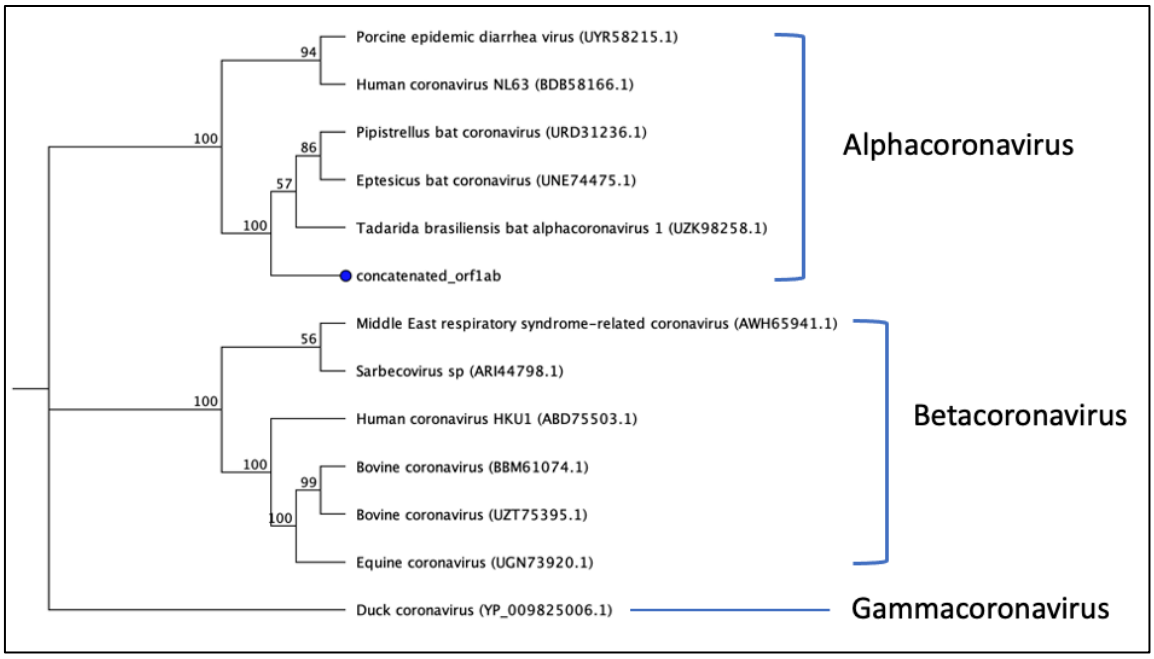

Figure 17. RAxML Tree of the Concatenated Contig at the ORF1ab Region. The phylogenetic relationship of this contig and other viruses between these two trees are very similar.

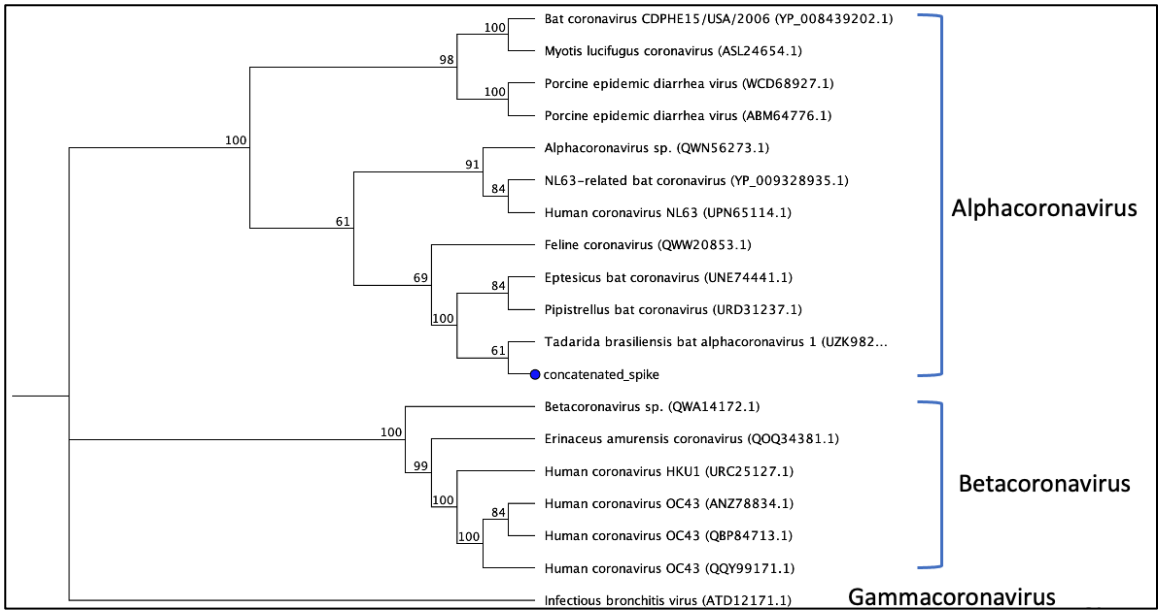

Figure 18. RAxML Tree of the Concatenated Contig at the Spike Region. No PhyML tree was constructed for this contig since it contains stop codons.

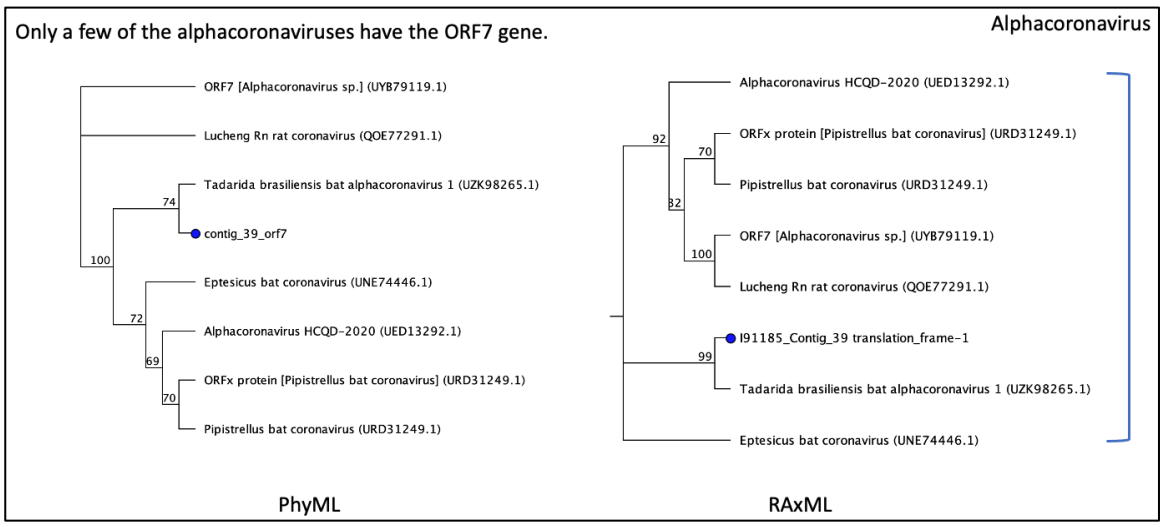

Figure 19. PhyML and RAxML Trees of the Concatenated Contig at the ORF7 Region. The phylogenetic relationship of this contig and other viruses between these two trees are very similar. ORF7 gene is not presented in every alpha-CoV, so the reference CoVs in these two trees are less than other trees.

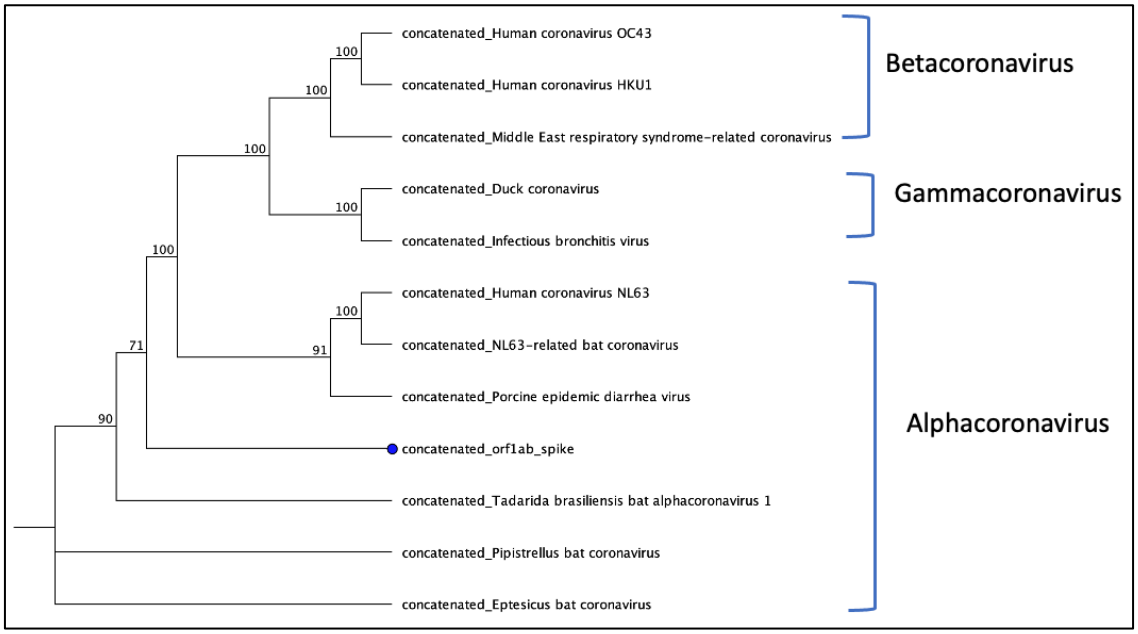

Figure 20. RAxML Tree of the Concatenated Contig at the ORF1ab and Spike Regions. This contig was generated by further concatenating the ORF1ab and the Spike contigs together. No PhyML tree was constructed for this contig since it contains stop codons. The reference CoV genomes are also cut and concatenated together to keep only the ORF1ab and the Spike regions.

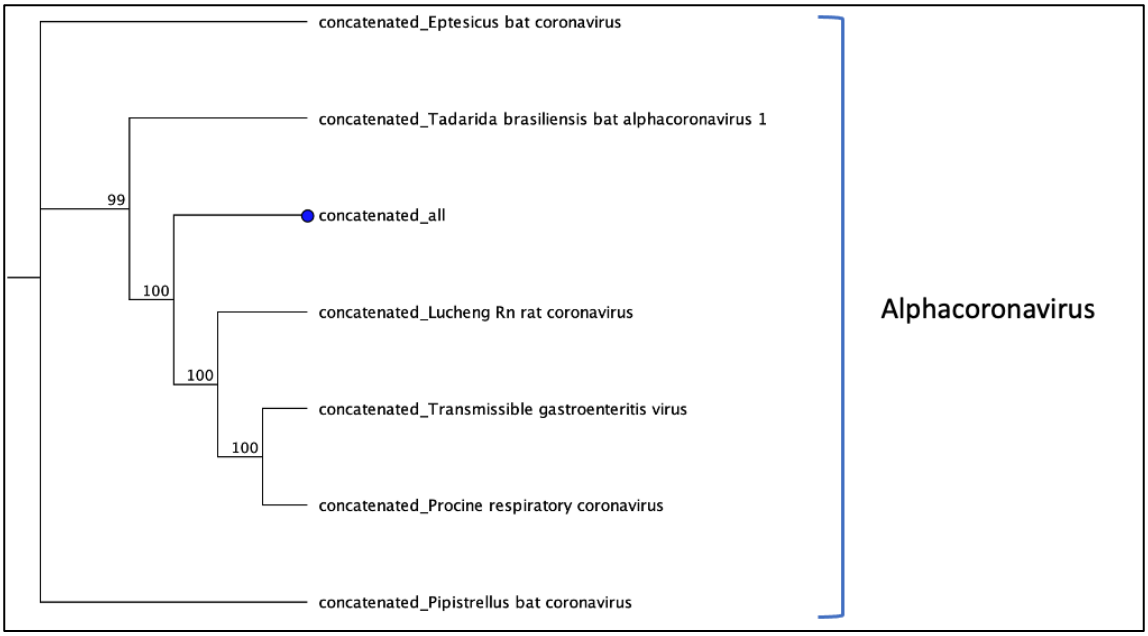

Figure 21. RAxML Tree of the Concatenated Contig at the ORF1ab, S, ORF7 Regions. This contig was generated by further concatenating the ORF1ab, the Spike and the ORF7 contig together. No PhyML tree was constructed for this contig since it contains stop codons. The reference CoV genomes are also cut and concatenated together to keep only the ORF1ab, Spike and ORF7 regions.

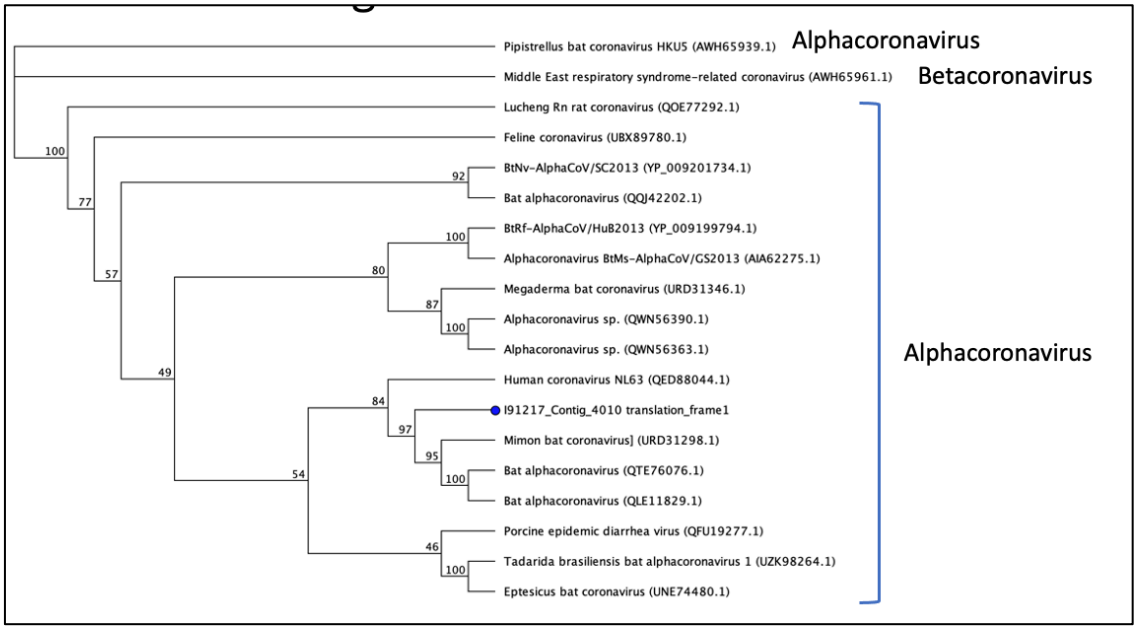

Figure 22. PhyML Tree of the Concatenated Contig at the N-gene Region.

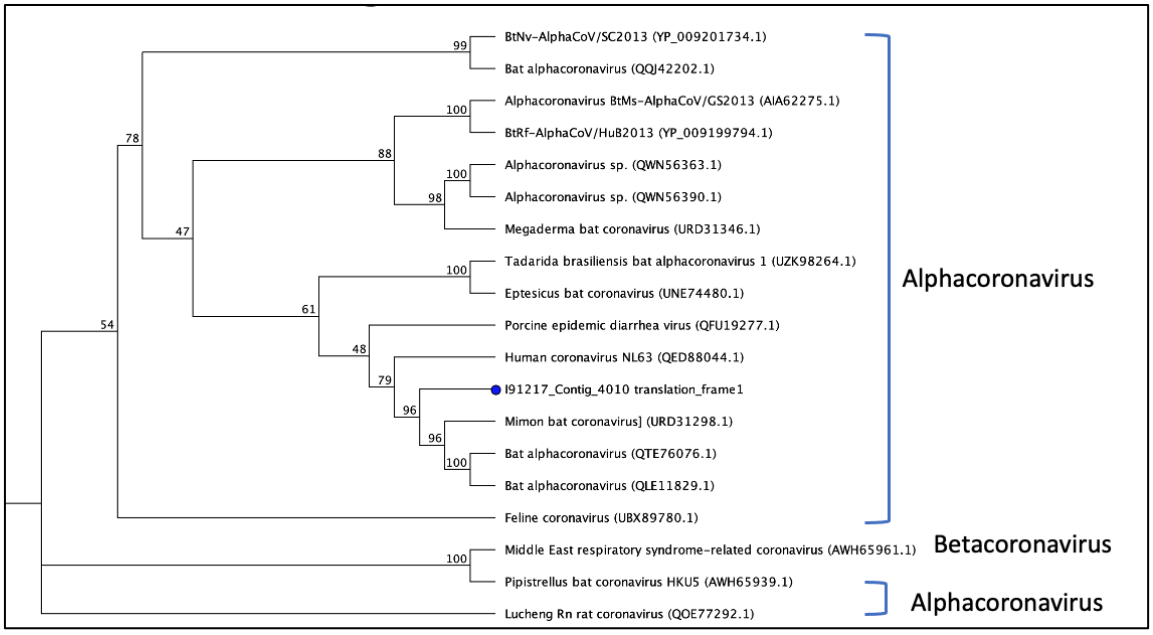

Figure 23. RAxML Tree of the Concatenated Contig at the N-gene Region. The phylogenetic relationship of this contig and other viruses between these two trees are very similar. There was no further concatenating since this sample only have one true positive contig.

Positive Control

The positive control sample was sequenced at Lim Lab previously with the same sequencing strategy, QC, and contigs building methods. The contigs went through the same workflow, using a customized *Picornaviridae* protein database. The results show that a long Rhinovirus B14 sequence was assembled, which is the content of the positive control sample.

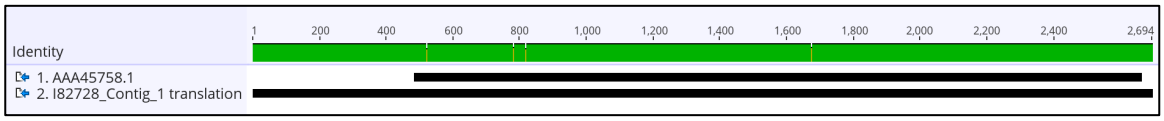

Figure 24. Long Contig of the Positive Control Sample. Its alignment result with the *Rhinovirus B14* virus genome have both high pairwise identity score and long matching region, showing the sequence of the positive control sample was generated from the workflow correctly.

Comparison with A Similar Workflow

A similar workflow was performed on the 216 samples without the host genome removal step in the QC and the customized protein database querying step. Other steps are the same. This workflow only found two contigs considered CoVs, included in the ten contigs described above. This comparison means the proposed workflow can catch more potential novel CoVs, reducing the false negative rate, thus, being more sensitive. The reason is that in megaBLAST, the CoV hits sometimes may be outside the top 100 or so results and may be discarded due to their lower rank. However, their query scores, such as e-value, may not be significantly lower than the top 100 hits. With a customized CoV protein database, these hits can be preserved beforehand.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This study analyzed 230 bat samples to explore the presence of novel CoVs. A custom bioinformatics workflow was developed, incorporating parallel computing for optimized performance. Among the samples, 14 had previously tested positive for CoVs using a pan-coronavirus qPCR. Illumina Next-Generation Sequencing generated shotgun readings.

The newly devised bioinformatics pipeline processed sequencing reads from each bat sample and positive control, generating longer viral contigs. These contigs underwent a BLASTx search against a tailored Coronavirus database, followed by further filtering with BLASTx and megaBLAST against the NCBI nr_nt database. Confirmed coronavirus contigs were then used to construct bootstrapped phylogenetic trees with representative Alpha, Beta, and Gamma-CoVs genomes.

Excitingly, two bat samples revealed novel CoV fragments, predominantly matching the ORF1ab, ORF7, and N gene regions. Phylogenetic analysis identified these fragments as Alpha-CoVs, closely related to Eptesicus Bat Coronavirus, Pipistrellus Bat Coronavirus, and Tadarida Brasiliensis Bat Alphacoronavirus 1.

In the future, the bats that produced the samples containing the novel CoVs can be sampled again and analyzed again, and longer CoV fragments or even the whole genome may be found. The novel CoVs can be further analyzed. Also, the workflow can be adapted to investigate other kinds of viruses for other or more bat samples. The manual processes and the steps involving Geneious Prime can be scripted for maximum automation.

This project highlights the utility of advanced sequencing techniques and bioinformatics workflows in uncovering novel virus variants within bat populations, advancing our understanding of virus evolution and potential zoonotic threats.

REFERENCES

Amoros, J. L. (2023, July 18). *The agile development process for mobile apps*. Krasamo. https://www.krasamo.com/agile-development-process/

Babraham Bioinformatics. (n.d.). *FastQC*. Babraham Bioinformatics - FastQC a quality control tool for high throughput sequence data. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Becker, D. J., Lei, G.-S., Janech, M. G., Brand, A. M., Fenton, M. B., Simmons, N. B., Relich, R. F., & Neely, B. A. (2022). *Serum Proteomics Identifies Immune Pathways and Candidate Biomarkers of Coronavirus Infection in Wild Vampire Bats*. https://doi.org/10.1101/2022.01.26.477790

Behjati, S., & Tarpey, P. S. (2013). What is next generation sequencing? *Archives of Disease in Childhood - Education & Practice Edition*, *98*(6), 236–238. https://doi.org/10.1136/archdischild-2013-304340

Bushnell. (2022, February 16). *BBTools*. DOE Joint Genome Institute. https://jgi.doe.gov/data-and-tools/software-tools/bbtools/

Biopython (n.d.). *Split large file*. Biopython wiki. https://biopython.org/wiki/Split_large_file

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., & Madden, T. L. (2009). BLAST+: Architecture and applications. *BMC Bioinformatics*, *10*(1). https://doi.org/10.1186/1471-2105-10-421

Edgar, R. C., & Batzoglou, S. (2006). Multiple sequence alignment. *Current Opinion in Structural Biology*, *16*(3), 368–373. https://doi.org/10.1016/j.sbi.2006.04.004

Ellis, P., Somogyvári, F., Virok, D. P., Noseda, M., & McLean, G. R. (2021). Decoding covid-19 with the SARS-COV-2 genome. *Current Genetic Medicine Reports*, *9*(1), 1–12. https://doi.org/10.1007/s40142-020-00197-5

Fehr, A. R., & Perlman, S. (2015). Coronaviruses: An overview of their replication and pathogenesis. *Coronaviruses*, 1–23. https://doi.org/10.1007/978-1-4939-2438-7_1

Frutos, R., Serra-Cobo, J., Pinault, L., Lopez Roig, M., & Devaux, C. A. (2021). Emergence of bat-related betacoronaviruses: Hazard and risks. *Frontiers in Microbiology*, *12*. https://doi.org/10.3389/fmicb.2021.591535

Galtier, N., & Gouy, M. (1998, July 1). *Inferring pattern and process: Maximum-likelihood implementation of a nonhomogeneous model of DNA sequence evolution*

for phylogenetic analysis. Molecular biology and evolution. https://pubmed.ncbi.nlm.nih.gov/9656487/

Geneious Prime 2023.0.4. (n.d.). Geneious Prime: Bioinformatics software for Sequence Data Analysis. http://www.geneious.com/

Geneious Prime. (2020). Which maximum likelihood tree builder should I use? – Geneious Prime. https://help.geneious.com/hc/en-us/articles/360045071571-Which-maximum-likelihood-tree-builder-should-I-use-

Guindon, S., Dufayard, J.-F., Lefort, V., Anisimova, M., Hordijk, W., & Gascuel, O. (2010). New algorithms and methods to estimate maximum-likelihood phylogenies: Assessing the performance of phyml 3.0. *Systematic Biology*, *59*(3), 307–321. https://doi.org/10.1093/sysbio/syq010

Guo, J., Bolduc, B., Zayed, A. A., Varsani, A., Dominguez-Huerta, G., Delmont, T. O., Pratama, A. A., Gazitúa, M. C., Vik, D., Sullivan, M. B., & Roux, S. (2021). Virsorter2: A multi-classifier, expert-guided approach to detect diverse DNA and RNA viruses. *Microbiome*, *9*(1). https://doi.org/10.1186/s40168-020-00990-y

Illumina DNA Prep Reference Guide (1000000025416). (n.d.-c). https://support.illumina.com/content/dam/illumina-support/documents/documentation/chemistry_documentation/illumina_prep/illumina-dna-prep-reference-guide-1000000025416-10.pdf

Illumina. (2023). *The future is created by you. experience Illumina's NextSeq 2000.* NextSeq 1000 and NextSeq 2000 Sequencing Systems | Mid-throughput benchtop sequencing. https://www.illumina.com/systems/sequencing-platforms/nextseq-1000-2000.html

Kreuze, J. F., Perez, A., Untiveros, M., Quispe, D., Fuentes, S., Barker, I., & Simon, R. (2009). Complete viral genome sequence and discovery of novel viruses by deep sequencing of small RNAS: A generic method for diagnosis, discovery and sequencing of viruses. *Virology*, *388*(1), 1–7. https://doi.org/10.1016/j.virol.2009.03.024

Krishnamurthy, S. R., & Wang, D. (2017). Origins and challenges of viral dark matter. *Virus Research*, *239*, 136–142. https://doi.org/10.1016/j.virusres.2017.02.002

Kulski, J. K. (2016). Next-generation sequencing — an overview of the history, tools, and "Omic" applications. *Next Generation Sequencing - Advances, Applications and Challenges*. https://doi.org/10.5772/61964

Li, W., & Godzik, A. (2006). CD-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, *22*(13), 1658–1659. https://doi.org/10.1093/bioinformatics/btl158

Liang, G., Zhao, C., Zhang, H., Mattei, L., Sherrill-Mix, S., Bittinger, K., Kessler, L. R., Wu, G. D., Baldassano, R. N., DeRusso, P., Ford, E., Elovitz, M. A., Kelly, M. S., Patel, M. Z., Mazhani, T., Gerber, J. S., Kelly, A., Zemel, B. S., & Bushman, F. D. (2020). The stepwise assembly of the neonatal virome is modulated by breastfeeding. *Nature*, *581*(7809), 470–474. https://doi.org/10.1038/s41586-020-2192-1

Maier, H. J., & Bickerton, E. (2021). *Coronaviruses: Methods and protocols*. Springer.

Martin, M. (2011). CUTADAPT removes adapter sequences from high-throughput sequencing reads. *EMBnet.Journal*, *17*(1), 10. https://doi.org/10.14806/ej.17.1.200

Meleshko, D., Hajirasouliha, I., & Korobeynikov, A. (2021). Coronaspades: From biosynthetic gene clusters to RNA viral assemblies. *Bioinformatics*, *38*(1), 1–8. https://doi.org/10.1093/bioinformatics/btab597

Morgulis, A., Coulouris, G., Raytselis, Y., Madden, T. L., Agarwala, R., & Schäffer, A. A. (2008). Database indexing for production MEGABLAST SEARCHES. *Bioinformatics*, *24*(24), 2942–2942. https://doi.org/10.1093/bioinformatics/btn554

National Center for Biotechnology Information. (n.d.) Appendices - BLAST® command line applications user Manual - NCBI Bookshelf. https://www.ncbi.nlm.nih.gov/books/NBK279684/

National Center for Biotechnology Information. (n.d.-b) BLAST® Command Line Applications user Manual - NCBI Bookshelf. https://www.ncbi.nlm.nih.gov/books/NBK279690/

National Library of Medicine (US). (1988). National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/

National Library of Medicine (US). (1988). *Nucleotide*. National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/nucleotide/

National Library of Medicine (US). (1988). *Protein*. National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/protein/

National Library of Medicine (US). (n.d.-a). *Genome*. National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/genome/

National Library of Medicine (US). (n.d.-c). *Genome assembly ASM294091v3*. National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_002940915.2/

National Library of Medicine (US). (n.d.-b). *Identical Protein Groups*. National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/ipg/

National Library of Medicine (US). (n.d.-c). *Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome*. National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/nuccore/1798174254

Nurk, S., Meleshko, D., Korobeynikov, A., & Pevzner, P. A. (2017). Metaspades: A new versatile metagenomic assembler. *Genome Research*, *27*(5), 824–834. https://doi.org/10.1101/gr.213959.116

Pascal, R. (2014, February 10). *How to correctly speed up blast using num_threads*. voorloopnul. https://voorloopnul.com/blog/how-to-correctly-speed-up-blast-using-num_threads/

Plyusnin, I., Kant, R., Jääskeläinen, A. J., Sironen, T., Holm, L., Vapalahti, O., & Smura, T. (2020). Novel NGS pipeline for virus discovery from a wide spectrum of hosts and sample types. *Virus Evolution*, *6*(2). https://doi.org/10.1093/ve/veaa091

Ren, J., Ahlgren, N. A., Lu, Y. Y., Fuhrman, J. A., & Sun, F. (2017). Virfinder: A novel K-mer based tool for identifying viral sequences from assembled metagenomic data. *Microbiome*, *5*(1). https://doi.org/10.1186/s40168-017-0283-5

Roux, S., Enault, F., Hurwitz, B. L., & Sullivan, M. B. (2015). Virsorter: Mining viral signal from microbial genomic data. *PeerJ*, *3*. https://doi.org/10.7717/peerj.985

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., Thompson, J. D., & Higgins, D. G. (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular Systems Biology*, *7*(1). https://doi.org/10.1038/msb.2011.75

Slurm Workload Manager. (n.d.). Slurm Workload Manager - Job Array Support. https://slurm.schedmd.com/job_array.html

SPAdes 3.15.4 Manual. (n.d.). https://cab.spbu.ru/files/release3.15.4/manual.html#bgc

Sridhar, S., To, K. K. W., Chan, J. F. W., Lau, S. K. P., Woo, P. C. Y., & Yuen, K.-Y. (2015). A systematic approach to novel virus discovery in emerging infectious disease outbreaks. *The Journal of Molecular Diagnostics*, *17*(3), 230–241. https://doi.org/10.1016/j.jmoldx.2014.12.002

Stamatakis, A. (2006). RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, *22*(21), 2688–2690. https://doi.org/10.1093/bioinformatics/btl446

Tange, O. (2023, May 22). *GNU parallel 20230522 ('charles') [stable]*. Zenodo. https://doi.org/10.5281/zenodo.7958356

Wang, Y., Grunewald, M., & Perlman, S. (2020). Coronaviruses: An updated overview of their replication and pathogenesis. *Coronaviruses*, 1–29. https://doi.org/10.1007/978-1-0716-0900-2_1

Wommack, K. E., Bhavsar, J., Polson, S. W., Chen, J., Dumas, M., Srinivasiah, S., Furman, M., Jamindar, S., & Nasko, D. J. (2012). Virome: A standard operating procedure for analysis of viral metagenome sequences. *Standards in Genomic Sciences*, *6*(3), 427–439. https://doi.org/10.4056/sigs.2945050

Zhao, G., Wu, G., Lim, E. S., Droit, L., Krishnamurthy, S., Barouch, D. H., Virgin, H. W., & Wang, D. (2017). VirusSeeker, a computational pipeline for virus discovery and Virome composition analysis. *Virology*, *503*, 21–30. https://doi.org/10.1016/j.virol.2017.01.005