Carlos Arevalo

Professor David Bernick, Ph.D.

BME160 Final Project

Spring 2020

Genomic and transcriptome analysis of SARS-CoV-2 from multiple genomes

Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) is responsible for Coronavirus Disease 19 (COVID19). SARS-CoV-2 has infected more than 6 million people worldwide and has taken the lives of more than 300,000 people up to date. The lack of understanding of the genetic footprints and etiology of SARS-CoV-2 has challenged us to develop efficient treatments and vaccines to further prevent the spread of the virus. Here, I described the development of a basic workflow pipeline approach to better understand the genomic footprints of SARS-CoV-2. I studied the genomic and subgenomic RNAs in SARS-CoV-2 with the purpose to compare and understand the evolution, structural proteins and pathogenicity of this deadly virus.

To develop a program that analyzes the genomic footprints of SARS-CoV-2, I redeveloped and improved the *findORFs.py* program from one of the BME160 assignments. I used the program to identify all possible open reading frames in SARS-CoV-2 genome. In addition, I created *fastaFinder.py* to get the fasta sequences for each possible open reading frame using a reference genome of SARS-CoV-2 from the NCBI National Library of Medicine Database. To accomplish this, I wrote *fastaFinder.py* using python available modules such as *pybedtools*. The *pybeedtools* module reads over the two files one formatted as a bed file containing the ORFs coordinates from the *findORFs.py* output and the other fasta file containing the reference genome (Program requires to change both files internally). Next, I created *getCodingSeq.py*, a program that gets the putative sequences or coding sequences in each ORF. Then, I created *filterORFs.py* uses dictionaries for containing the structural and nonstructural protein encoding names for each gene. The program outputs files containing putative proteins only. It was necessary to translate the putative sequences from nucleotides to amino acids to study the evolution of each protein. To accomplish these, I created *seqTranslator.py*, a program that translates a fasta file containing multiple sequences to amino acid sequences.

To study the evolution of SARS-CoV-2 genomic and subgenomic RNAs, I used multiple genomes from viruses related to SARS-CoV-2 available at the NCBI National Library of Medicine. I obtained a file containing putative protein sequences from each genome. Then, I performed multiple sequence alignment using Clustalw Omega. Finally, I used the alignment results to perform data analysis in Jupyter Notebook and build phylogenetic trees for common proteins in each virus genome .

This workflow pipeline provides a formal and basic approach to study the genomic architecture of the genome. In addition, it allows us to deeply study and predict putative proteins and trace their evolution. I tested the program in other genomes to identify their genome footprints and validated its results using reliable programs available on the internet such as the NCBI National Library of Medicine, and publications. If the workflow pipeline programs are correct, this study revealed that some genomic architectures and structural proteins such as ORF S surface glycoprotein (spike) do not share a close evolution such as other members of the family of coronaviruses share among themselves. A further characterization and study of protein function is necessary to further understand the pathogenicity of SARS-CoV-2.

A.

Distance Matrix

==========

SARSCoV2_S	0			
BtRsBetaCoV_S	0.9036993540810334	0		
Rs4231_S	0.9072225484439225	0.14503816793893132	0	
CoV_U_S	0.9060481503229595	0.14856136230182027	0.12448620082207873	0
	SARSCoV2_S	BtRsBetaCoV_S	Rs4231_S	CoV_U_S

B.

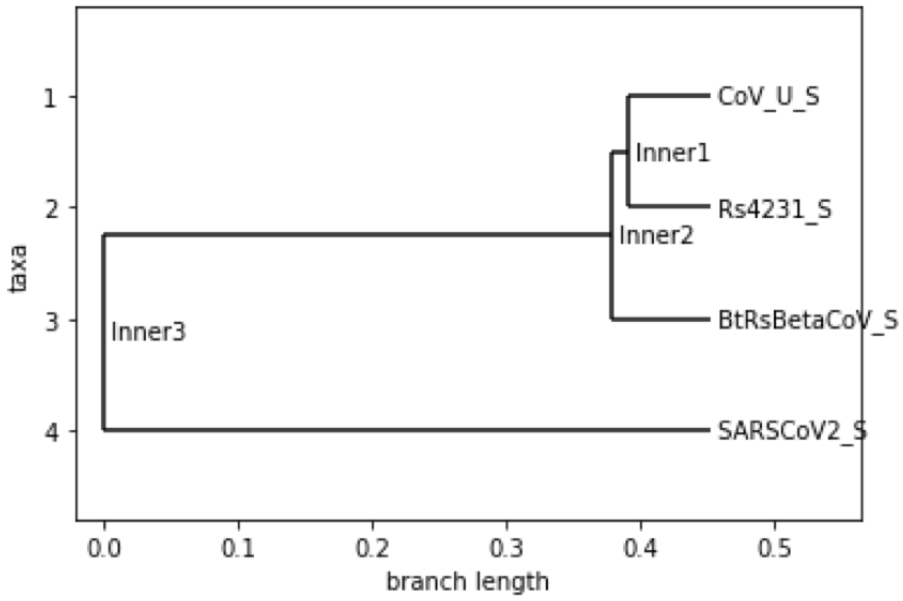


Figure 1. Data analysis and phylogenetic tree of ORFS spike protein encoded by a subgenomic RNA. A) Distance Matrix results from sequence alignment on Clustalw. **B)** Phylogenetic three of ORF S surface glycoprotein evolution in SARS-CoV-2, Bat SARS-like coronavirus Rs4231, Bat coronavirus RaTG13 and Coronavirus BtRs-BetaCoV genomes in Phylo.

REFERENCES

1. Kim, D. et al. (2020). The architecture of SARS-CoV-2 transcriptome. Cell 181, 914–921. https://doi.org/10.1016/j.cell.2020.04.011
2. Andersen, K.G., Rambaut, A., Lipkin, W.I. *et al.* The proximal origin of SARS-CoV-2. *Nat Med* 26, 450–452 (2020). https://doi.org/10.1038/s41591-020-0820-9

EXTENDED FIGURES AND WORK

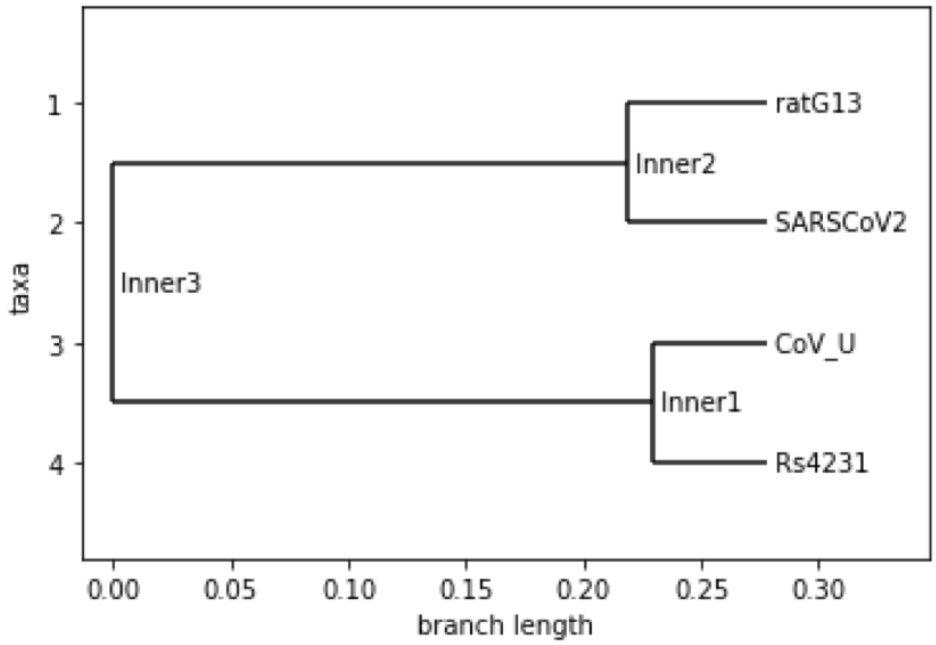
A.

Distance Matrix

==========

Rs4231	0			
CoV_U	0.09642193308550184	0		
SARSCoV2	0.5485594795539033	0.5587825278810409	0	
ratG13	0.5652881040892193	0.5518122676579926	0.1189591078066915	0
	Rs4231	CoV_U	SARSCoV2	ratG13

B.



Extended Figure 1. Data analysis and phylogenetic tree of ORF1a polyprotein encoded by a genomic RNA. A) Distance Matrix results from sequence alignment on Clustalw. **B)** Phylogenetic three of ORF1a polyprotein sequences evolution in SARS-CoV-2, Bat SARS-like coronavirus Rs4231, Bat coronavirus RaTG13 and Coronavirus BtRs-BetaCoV genomes in Phylo.

The data to perform this study will come from the NCBI National Library of Medicine, Global Initiative on Sharing All Influenza Data (GISAID), and the UCSC Genome Browser. In addition, these data sources and recent studies will be used to validate the results.

The viruses genomes used in this study were chosen with the goal of tracing the evolution of SARS-CoV-2 around the world and comparing its evolution with other viral species. The other viral species were chosen randomly from the direction of Andersen, K.G., Rambaut, A., Lipkin, W.I. *et al* 2020, and taking into account the pathogenicity of the virus.

SARS-Coronavirus Related Genomes

Species	Genome File
Severe Acute Respiratory Syndrome (SARS) coronavirus 2	SARSCoV2.fa
Bat SARS-like coronavirus Rs4231	sarsRs4231.fa
SARS coronavirus Urbani	coronavirusUrbani.fa
Bat coronavirus RaTG13	ratG13.fa
Coronavirus BtRs-BetaCoV	coronavirusBtRs.fa

Required Modules

> pybedtools.py
> biopython.py
> Clustalw

Program Workflow

STEP 1: findORFs.py

Find ORFs in genome
Program get genomic and subgenomic RNAs
In this study, I focused in subgenomic RNAs

> python findORFs.py -lG -s "ATG" -mG 100< SARSCov2.fa > sars2.bed
> python findORFs.py -lG -s "ATG" -mG 100< ratG13.fa > ratG13.bed
> python findORFs.py -lG -s "ATG" -mG 100< coronavirusUrbani.fa > coronovirusUrbani.bed
> python findORFs.py -lG -s "ATG" -mG 100< sarsRs4231.fa > sarsRs4231.bed
> python findORFs.py -lG -s "ATG" -mG 100< coronavirusBtRs.fa > coronavirusBtRs.bed

STEP 2 : fastaFinder.py

Get fasta sequences for each ORFs using a reference genome
If a novel genome was sequence , it has to be used as reference to get original ORFs fasta
sequences in that new genome
Assign a name associated with the virus name for output file to keep track
You have to change the reference genome in, every time when running the program

> python fastaFinder.py > sars2Seq.fa
> python fastaFinder.py > sarsG13Seq.fa
> python fastaFinder.py > coronavirusUrbaniSeq.fa
> python fastaFinder.py > sarsRs4231Seq.fa
> python fastaFinder.py > coronavirusBtRsSeq.fa

STEP 3: getCondingSeq.py

Clean ORFs fasta sequences and returns proteins coding sequences (putative sequences)

> python getCodingSeq.py < sars2Seq.fa > sars2PutativeSeq.fa
> python getCodingSeq.py < coronavirusBtRsSeq.fa > btRsPutativeSeq.fa
> python getCodingSeq.py < sarsG13Seq.fa > sarsG13PutativeSeq.fa
> python getCodingSeq.py < sarsRs4231Seq.fa > sarsRs4231PutativeSeq.fa
> python getCodingSeq.py < coronavirusUrbaniSeq.fa > cvUrbaniPutativeSeq.fa

STEP 4: filterORFs.py

Return only protein sequences I am interested in studying
Concatenate protein sequences from each genomes to a single fasta file to build a
phylogenetic three and keep track the evolution of the virus' protein

> python filterORFs.py < sars2Seq.fa > sars2FilProt.fa
> python filterORFs.py < sars2PutativeSeq.fa > sars2FilProt.fa
> python filterORFs.py < sarsG13PutativeSeq.fa > sarsG13FilProt.fa
> python filterORFs.py < sarsRs4231PutativeSeq.fa > sars4231FilProt.fa
> python filterORFs.py < btRsPutativeSeq.fa > btRsFiltProt.fa
> python filterORFs.py < cvUrbaniPutativeSeq.fa > cvUrbaniFilProt.fa

STEP 5: seqTranslator.py

translate dna coding sequences to single letter amino acid sequences

> python seqTranslator.py < sars2FilProt.fa > sars2ProteinSeq.fa
> python seqTranslator.py < btRsFiltProt.fa > btRsProteinSeq.fa

> python seqTranslator.py < sars4231FilProt.fa > sars4231ProteinSeq.fa
> python seqTranslator.py < sarsG13FilProt.fa > sarsG13ProteinSeq.fa
> python seqTranslator.py < cvUrbaniFilProt.fa > coronavirusUProteinSeq.fa

STEP 6: Protein Alignments

Align similar proteins from each genome and build a phylo three
Protein alignments were performed in Clustal Omega Multiple Sequence Alignment: https://www.ebi.ac.uk/Tools/msa/clustalo/

STEP 7: Data Analysis in Jupyter Notebook