Population Genetics and Evolutionary Biology Project

Alasia Miller

Introduction: Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) is the virus that causes coronavirus disease 2019 (COVID-19). The COVID-19 outbreak originated in Wuhan, Hubei province, China. It has since developed into a pandemic virus that nearly 124.0 Million people have contracted, and 6.6 million people have died from(1). Severe acute respiratory syndrome is the virus that caused the 2002–2004 outbreak of SARS. In total, 8,098 people were infected within nearly 30 countries, and there were approximately 774 deaths worldwide(2). The first known case of SARS was discovered in Foshan, Guangdong, China, in mid-November 2002. Both viruses are from a closely related family of coronarviruses. In this project, I identified orthologous genes and determined evolutionary rates of each gene in the SARS and Covid-19 coronavirus genomes. dN, dS, and dN/dS are measures of evolutionary rates. dN is the number of nonsynonymous (i.e., amino acid altering) changes between two sequences per nonsynonymous site. dS is the number of synonymous (i.e., silent) changes between two sequences per synonymous site. dS is often assumed to represent the neutral rate of substitution. dN/dS represents the ratio of non-synonymous mutations per non-synonymous site (dN) to synonymous mutations per synonymous site (dS), and is the rate of protein evolution.

Methods: My project was executed using whole genome data from SARS and Covid-19 at the National Center for Biotechnology Information (NCBI) RefSeq genome collection. The software I used was R (version 4.2.2), Blast+ (version 2.2.31+), Argtable(version 2.13), Clustal Omega ("clustalo")(version clustal-omega-1.2.4.), KaKs_Calculator (version 1.2), biomartr (version 1.0.2), and Ubuntu (version 22.04.1). Using Ubuntu as the operating system on Linux by which I executed all of my commands, I first installed R into my system. To install BLAST+, I extracted the compiled version of BLAST I retrieved from a web server, and copied BLAST files to `usr/local/bin`. Following this, I retrieved the argtable from a server, extracted the file from the folders then ran the system from inside argtable2-13 folder and installed argtable libraries in /usr/local/lib. I repeated the same steps I used for argtable for my clustal omega download. Then, I retrieved the KaKs_Calculator file from a web servers, unzipped the file, installed it, and opened the folder. While inside this folder, I used vim base.h to edit "include<string.h>" into KaKs_Calculator. I finally saved and quit the file. For my next step, I downloaded several dependencies ("libfreetype6-dev", "libpng-dev","libtiff5-dev", "libjpeg-dev") for devtools to run. I loaded R, then installed devtools using install.packages("devtools"). To download orthologr, I Installed package dependencies ("Biostrings", "GenomicRanges", "GenomicFeatures", "Rsamtools", "rtracklayer","IRanges"). Next, I installed metablastr from GitHub and orthologr from GitHub. After, I installed a core Bioconductor package "BiocManager", installed package dependencies which were "Biostrings" and "biomaRt", and finally installed "biomartr". Subsequently, I loaded orthologr and biomartr library. I used the biomartr package to download the CDS sequences for SARS and Covid-19. I created a function to compute dN/dS values for covid-19 versus SARS. In this function, I first labeled the query species (SARS) and a subject species (covid-19). Then, I used the orthologr R package to identify orthologs using reciprocal best hit ("RBH") approach, aligned my multiple alignments with Clustal Omega, ran pal2nal to convert protein alignments back to nucleotide alignments, estimated dN, dS and dN/dS using Comeron's method from multiple sequence alignments, and finally set my multi-core processing comp_cores to equal 1. Lastly, I stored the results in an Excel readable .csv file.

To identify which gene in my analysis corresponds to the Spike protein, I utilized NCBI website. To find the websites for these two reference genomes, I went to NCBI (https://www.ncbi.nlm.nih.gov/) and searched "All

Databases" for "GCF_009858895.2" (Covid-19 reference genome) and "GCF_000864885.1" (SARS reference genome). I then clicked on the main link in the "Genome" section at the top of the search result. To find the gene names, I searched the GenBank formatted files for each genome to get detailed gene and other information. On the main genome assembly page for Covid-19, I found the GenBank formatted files under the "Chromosomes" section of the genome pages and clicking on the link under "RefSeq".

Results: 1. The total number of CDS sequences in both Covid-19 and SARS genomes is 22.

2. The total number of proteins meeting reciprocal best hit (RBH) approach to ortholog identification is 11.

3. A table with query id, subject id, dN, dS, dN/dS and percent identity for orthologous gene pairs

	A	B	C	D	E	F
1	query_id	subject_id	dN	dS	dNdS	perc_identity
2	lcl|NC_004718.3_cds_NP_828849.7_1	lcl|NC_045512.2_cds_YP_009724389.1_1	0.08863	0.9927	0.08928	86.16
3	lcl|NC_004718.3_cds_YP_009825051.1_3	lcl|NC_045512.2_cds_YP_009724390.1_3	0.155	1.368	0.1133	75.98
4	lcl|NC_004718.3_cds_YP_009825052.1_4	lcl|NC_045512.2_cds_YP_009724391.1_4	0.1693	0.9228	0.1834	72.46
5	lcl|NC_004718.3_cds_YP_009825054.1_6	lcl|NC_045512.2_cds_YP_009724392.1_5	0.02235	0.1522	0.1468	94.81
6	lcl|NC_004718.3_cds_YP_009825055.1_7	lcl|NC_045512.2_cds_YP_009724393.1_6	0.06771	0.6276	0.1079	90.58
7	lcl|NC_004718.3_cds_YP_009825056.1_8	lcl|NC_045512.2_cds_YP_009724394.1_7	0.1775	0.9235	0.1922	68.85
8	lcl|NC_004718.3_cds_YP_009825057.1_9	lcl|NC_045512.2_cds_YP_009724395.1_8	0.08593	0.675	0.1273	85.37
9	lcl|NC_004718.3_cds_YP_009825058.1_10	lcl|NC_045512.2_cds_YP_009725318.1_9	0.1488	0.6588	0.2259	85.37
10	lcl|NC_004718.3_cds_YP_009825060.1_12	lcl|NC_045512.2_cds_YP_009724396.1_10	0.9873	NA	NA	40.48
11	lcl|NC_004718.3_cds_YP_009825061.1_13	lcl|NC_045512.2_cds_YP_009724397.2_11	0.05848	0.3576	0.1635	90.54
12	lcl|NC_004718.3_cds_YP_009944365.1_2	lcl|NC_045512.2_cds_YP_009725295.1_2	0.128	1.21	0.1058	80.44

Figure 1: Query Table.

Discussion: The predominant mode of selection acting on viral genes in my analysis was purifying selection. Purifying selection is a type of selection that reduces the probability of fixation of a deleterious allele. The low dN/dS value (< 1) for the orthologous gene pairs from the Covid-19 and SARS genome indicates that these genes are under strong purifying selection. The Covid-19 subject_id result displayed in column B in the Figure 1 Query Table, cl|NC_045512.2_cds_YP_009724390.1_3, is the surface spike glycoprotein in Covid-19. The Spike glycoprotein is what regulates fusion to the host membrane, and releases the virus to the cytoplasm using intracellular receptors(3). The dN is 0.155, dS is 1.368 and dN/dS value is 0.1133 for this critical gene. $dS > 1$ means that on average, more than one mutation has accumulated at the same site, and the orthologous gene pairs from the Covid-19 and SARS genome that show $dS > 1$ are the surface Spike glycoprotein and the ORF1a polyprotein. Thus, we can ascertain that one reason the Spike glycoprotein and ORF1a polyprotein genes differ in evolutionary rate in comparison to other protein coding genes, is because purifying selection is eliminating non-synonymous mutations at a faster rate than synonymous mutations.

Works Cited

1) World Health Organization. (2022, December 19). "WHO Coronavirus (COVID-19) Dashboard." Retrieved from https://covid19.who.int/
2) Lam WK, Zhong NS, Tan WC. Overview on SARS in Asia and the world. Respirology. 2003 Nov;8 Suppl(Suppl 1):S2-5. doi: 10.1046/j.1440-1843.2003.00516.x. PMID: 15018125; PMCID: PMC7159403.
3) Huang, Y., Yang, C., Xu, Xf. et al. Structural and functional properties of SARS-CoV-2 spike protein: potential antivirus drug development for COVID-19. Acta Pharmacol Sin 41, 1141–1149 (2020). https://doi.org/10.1038/s41401-020-0485-4

Appendix

R

```
sudo apt-get update #To install the complete R system
sudo apt-get install r-base

Output:
R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
```

blast

```
# download BLAST+ version 2.2.31
wget ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/2.2.31/ncbi-blast-2.2.31+-x64-linux.tar.gz

# extract the compiled version of BLAST
tar zxvpf ncbi-blast-2.2.31+-x64-linux.tar.gz

# copy BLAST files to `usr/local/bin`
sudo -i cp -r ~/ncbi-blast-2.2.31+/bin/* /usr/local/bin


#calling R, checking if BLAST can be executed from R
$ R
> system("blastp -version")
blastp: 2.2.31+
Package: blast 2.2.31, build Jun  2 2015 10:20:04
```

Argtable/clustalo(clustal omega)

```
#download argtable: https://packages.ubuntu.com/bionic/libargtable2-dev #now in bionic...
~$ wget http://archive.ubuntu.com/ubuntu/pool/universe/a/argtable2/argtable2_13.orig.tar.gz

# extract folders from argtable2_13.orig.tar.gz
tar zxvpf argtable2_13.orig.tar.gz

#run inside folder: #~/argtable2-13
cd argtable2-13
./configure

make

make check

sudo make install
#Libraries have been installed in: /usr/local/lib


#download clustal omega
wget http://www.clustal.org/omega/clustal-omega-1.2.4.tar.gz

# extract folders from argtable2_13.orig.tar.gz
tar zxvpf clustal-omega-1.2.4.tar.gz

#run inside folder: #~/clustal-omega-1.2.4
cd clustal-omega-1.2.4
```

```
./configure

make

sudo apt install make

#Libraries have been installed in: /usr/local/lib
```

KaKs_Calculator (version 1.2)

```
# download KaKs_Calculator 1.0 ,follow base.h instructions from here https://www.shengweihou.com/blog/i

wget https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/kaks-calculator/Ka

# unzip
gzip -d KaKs_Calculator1.2.tar.gz
tar -xf KaKs_Calculator1.2.tar

# install, all commands run inside ~/KaKs_Calculator1.2/src folder
cd KaKs_Calculator1.2/src
vim base.h #writing the line below
#.
#.
#.
#/* Stanard lib of C++ */
#include<string>
#include<iostream>
#include<sstream>
#include<fstream>
#include<vector>
#include<stdlib.h>
#include<math.h>
#include<time.h>
#include<string.h>
#.
#.
#.
#hit esc, type :wq
sudo make
sudo cp KaKs_Calculator /usr/local/bin


#running r and checking for
$ cd ~
$ R
> system("KaKs_Calculator -h")

#output:
**********************************************************************
  Program: KaKs_Calculator
  Version: 1.2, Apr. 2006
  Description: Calculate Ka and Ks through model selection and model averaging.
**********************************************************************
```

devtools

```
sudo apt install libfreetype6-dev libpng-dev libtiff5-dev libjpeg-dev (Debian, Ubuntu, etc) #set up dep
$ R
>install.packages("devtools")
>library(devtools)

#output:
Loading required package: usethis
```

Orthologr

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install()

# Install package dependencies
BiocManager::install(c("Biostrings", "GenomicRanges", "GenomicFeatures", "Rsamtools", "rtracklayer","IR

# install metablastr from GitHub
devtools::install_github("HajkD/metablastr", force = TRUE)


# install orthologr from GitHub
devtools::install_github("HajkD/orthologr", force = TRUE)
system("orthologr -version")
```

biomartr

```
# Install core Bioconductor packages
if (!requireNamespace("BiocManager"))
    install.packages("BiocManager")
BiocManager::install()

# Install package dependencies
BiocManager::install("Biostrings")
BiocManager::install("biomaRt")


# install biomartr 1.0.2
install.packages("biomartr", dependencies = TRUE)
system("biomartr -version")
```

```
library(orthologr)
library(biomartr)
#"GCF_009858895.2" for Covid-19
#"GCF_000864885.1" for SARS

# download all coding sequences for covid-19
covid19_file <- biomartr::getCDS(organism = "GCF_009858895.2", path = getwd())
#The genomic CDS of 'GCF_009858895.2' has been downloaded to 'home/Alasi' and has been named 'GCF_00985

# download all coding sequences for SARS
```

```
SARS_file <- biomartr::getCDS(organism = "GCF_000864885.1", path = getwd())
#The genomic CDS of 'GCF_000864885.1' has been downloaded to 'home/Alasi' and has been named 'GCF_00086

# compute dN/dS values for covid-19 versus SARS
cd19_vs_sar_dNdS <- dNdS(query_file = SARS_file,
       subject_file    = covid19_file,
       delete_corrupt_cds = FALSE,
       ortho_detection = "RBH", # perform BLAST best reciprocal hit orthology inference
       aa_aln_type     = "multiple",  # perform multiple alignments of AA seqs
       aa_aln_tool     = "clustalo", #using clustal omega
       codon_aln_tool  = "pal2nal", # perform codon alignments using the tool Pal2Nal
       dnds_est.method = "Comeron", # use Comeron's method for dN/dS inference
       comp_cores      = 1) #multi-core processing 'comp_cores = 1'
# store result in Excel readable csv file
install.packages("readr")
readr::write_excel_csv(cd19_vs_sar_dNdS, "alas_cd19_vs_sar_dNdS.csv")

#checking to see if .csv file was written correctly
check_file <- read.csv("alas_cd19_vs_sar_dNdS.csv")
head(check_file)
```