

---

# REPLICATION OF SARS-CoV-2 MUTATION ANALYSIS SUGGESTS DIFFERENCES IN PER-PROTEIN MUTATION CHARACTERISTICS

---

A PREPRINT

**William Svoboda \***  
Computer Science  
Princeton University  
Princeton, NJ  
wsvoboda@princeton.edu

**Brendan McManamon**  
Electrical and Computer Engineering  
Princeton University  
Princeton, NJ  
bm18@princeton.edu

**Sara Schwartz**  
Computer Science  
Princeton University  
Princeton, NJ  
sarats@Princeton.edu

December 23, 2021

## Abstract

The increasing spread of COVID-19, caused by the virus SARS-CoV-2, raises concerns about the extent to which mutations have occurred across the viral genome. We present a partial replication of an earlier 2021 study by Wang, R. *et al.* that determined the presence of four substrains and eleven top mutations in the United States. We analyze a portion of the authors' data set in order to recreate Figure S1 from the paper, recapitulating the same features observed in the original figure. We further generate a summary of mutation characteristics for each of the 26 named proteins and confirm the significance of the spike protein at roughly 24% of all recorded mutations. Our analysis suggests that additional factors may affect per-protein mutation rate besides protein length.

**Keywords** COVID-19 · SARS-CoV-2 · coronavirus · mutation

## 1 Background

The spread of coronavirus disease 2019 (COVID-19) continues to be of scientific and public health interest in the context of the ongoing global pandemic. Of particular concern is whether the underlying virus, severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), has become more infectious as a result of new mutations.

A 2021 paper by Wang, R. *et al.*<sup>1</sup> analyzed 45,494 complete SARS-CoV-2 genome sequences in order to analyze their mutations. Of the 12,754 sequences from the United States, the authors determined the presence of four substrains and eleven top mutations. Furthermore, the study concluded that two of these substrains had become more infectious given the mutations found on the spike protein.

We present a partial replication of the original study in order to validate its findings. A subset of the data used in the paper was processed in order to recreate selected figures (Figure 1). Further analysis was also performed in order to summarize mutation characteristics for each protein across the entire SARS-CoV-2 genome. Finally, we discuss the implications of these findings on our current understanding of the SARS-CoV-2 viral makeup.

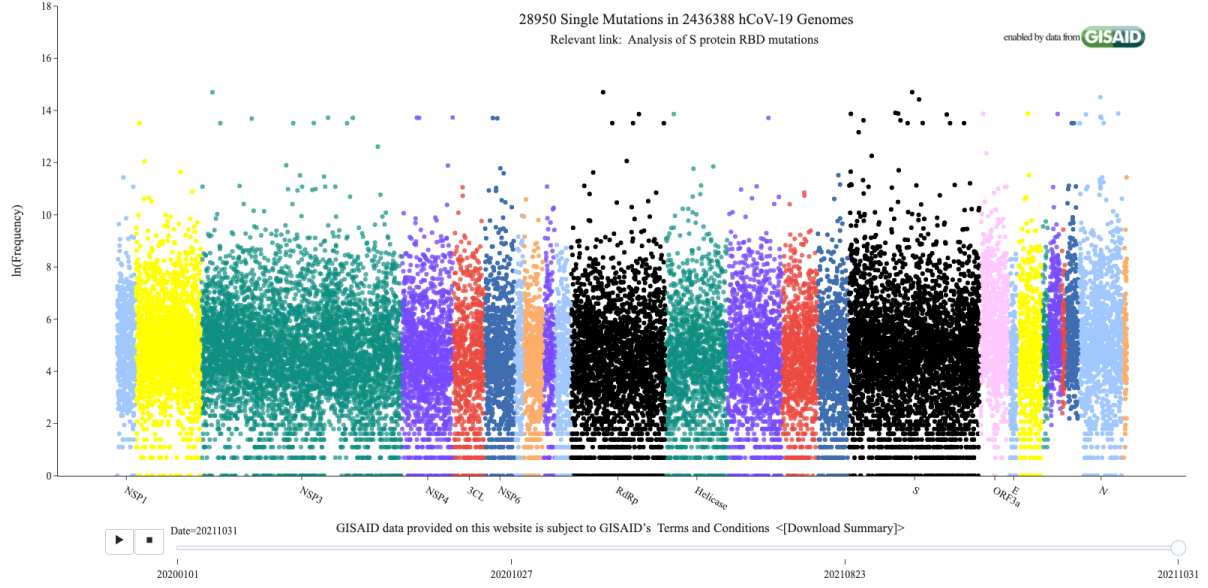


Figure 1: Screenshot of the mutation tracker website showing the distribution of mutations across the entire SARS-CoV-2 genome.

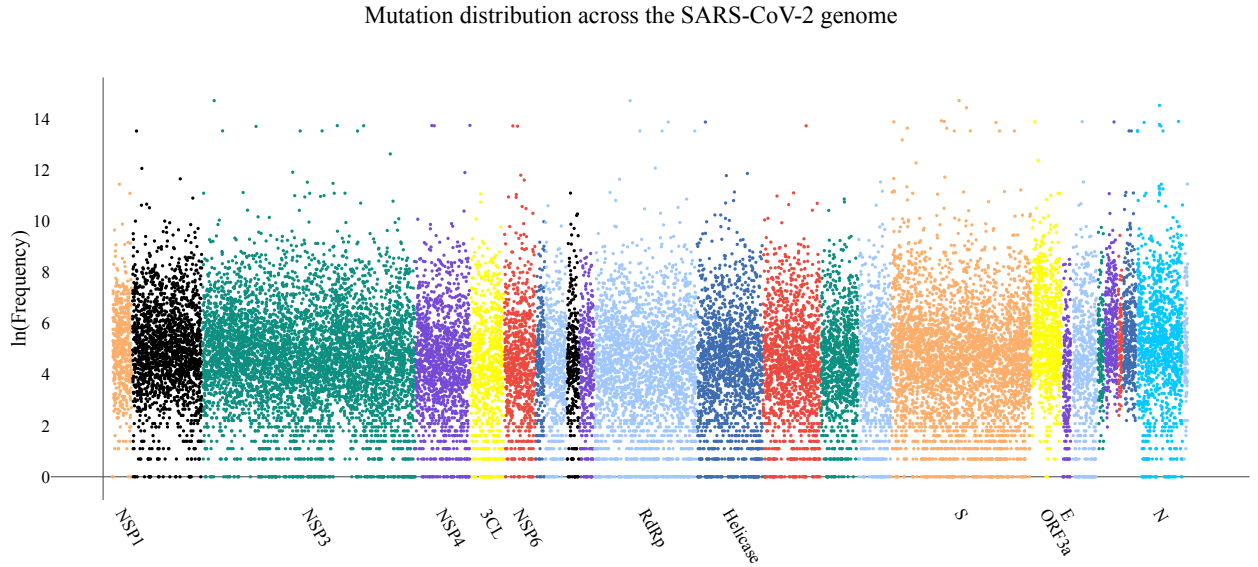


Figure 2: Genome-wide SARS-CoV-2 mutation distribution as of 31 October 2021. As in the original distribution figure, the natural log frequency of mutations on specific nucleotide positions is plotted on the  $y$ -axis. The positions of certain landmark proteins are also annotated on the  $x$ -axis. (author W.S.)

Table 1: Summary statistics for each SARS-CoV-2 protein (author W.S.)

protein	total mutations	% total mutations	protein length	mutations/nucleotide
Spike	17,134,885	23.785	3,821	4,484.398
NSP3	12,191,479	16.923	5,834	2,089.729
Nucleocapsid	11,760,145	16.324	1,259	9,340.862
RdRp	7,193,224	9.985	2,794	2,574.525
NSP4	3,612,766	5.015	1,499	2,410.117
NSP6	2,735,245	3.797	869	3,147.578
ORF3a	2,490,822	3.457	827	3,011.877
NSP2	2,392,844	3.321	1,913	1,250.833
Helicase	2,319,519	3.220	1,802	1,287.191
ORF8	2,072,786	2.877	365	5,678.866
Exonuclease	1,851,994	2.571	1,580	1,172.148
Membrane	1,606,101	2.229	668	2,404.343
ORF7a	1,462,425	2.030	365	4,006.644
2'-O-ribose MTases	564,099	0.783	893	631.690
endoRNase	544,308	0.756	1,037	524.887
3CL	484,768	0.673	917	528.646
NSP1	433,783	0.602	539	804.792
NSP9	302,133	0.419	338	893.885
NSP8	254,185	0.353	593	428.642
ORF10	163,272	0.227	116	1,407.517
NSP7	125,480	0.174	248	505.968
NSP10	122,023	0.169	416	293.325
ORF6	109,947	0.153	185	594.308
ORF7b	56,865	0.079	131	434.084
Envelope	53,802	0.075	227	237.013
NSP11	2,578	0.004	38	67.842

## 2 Results

### 2.1 Mutation Tracker

As part of the original study<sup>1</sup>, a website called [Mutation Tracker](#) was created to report the distribution of unique single mutations across the entire SARS-CoV-2 genome. A screen shot of this tracker is included in the paper's<sup>1</sup> supplementary information (Figure S1, included as [Figure 1](#) in this report). We downloaded the summary data provided from the website, itself generated by Wang, R. *et al.* using sequence data from GISAID (<https://www.gisaid.org/>). The summary data is updated automatically by the mutation website, with the totals from 31 October 2021 being used in this paper. Split among 26 different files for each named protein, the summary data was reprocessed into a single data frame representing the whole viral genome.

As [Figure 2](#) shows, we replicated the original distribution graph using this updated data. It is evident that NSP3 and the spike protein are both among the largest proteins and have experienced a high frequency of unique mutations across their domains. We also observe that ORF7a, ORF7b, and ORF8 all have relatively high mutation frequencies, in spite of their short length.

### 2.2 Mutation Summary Statistics

To quantify our interpretations, we computed further statistics to summarize the mutation distribution. [Table 1](#) lists this data for each protein ordered by total mutation frequency. The spike protein, NSP3, and the nucleocapsid are the top three proteins, with their respective frequencies (17,134,885, 12,191,479, and 11,760,145 mutations) being a full order of magnitude greater than the rest. Interestingly, we note that

\*website: <https://will.cx>, email: [wcsvoboda@gmail.com](mailto:wcsvoboda@gmail.com)

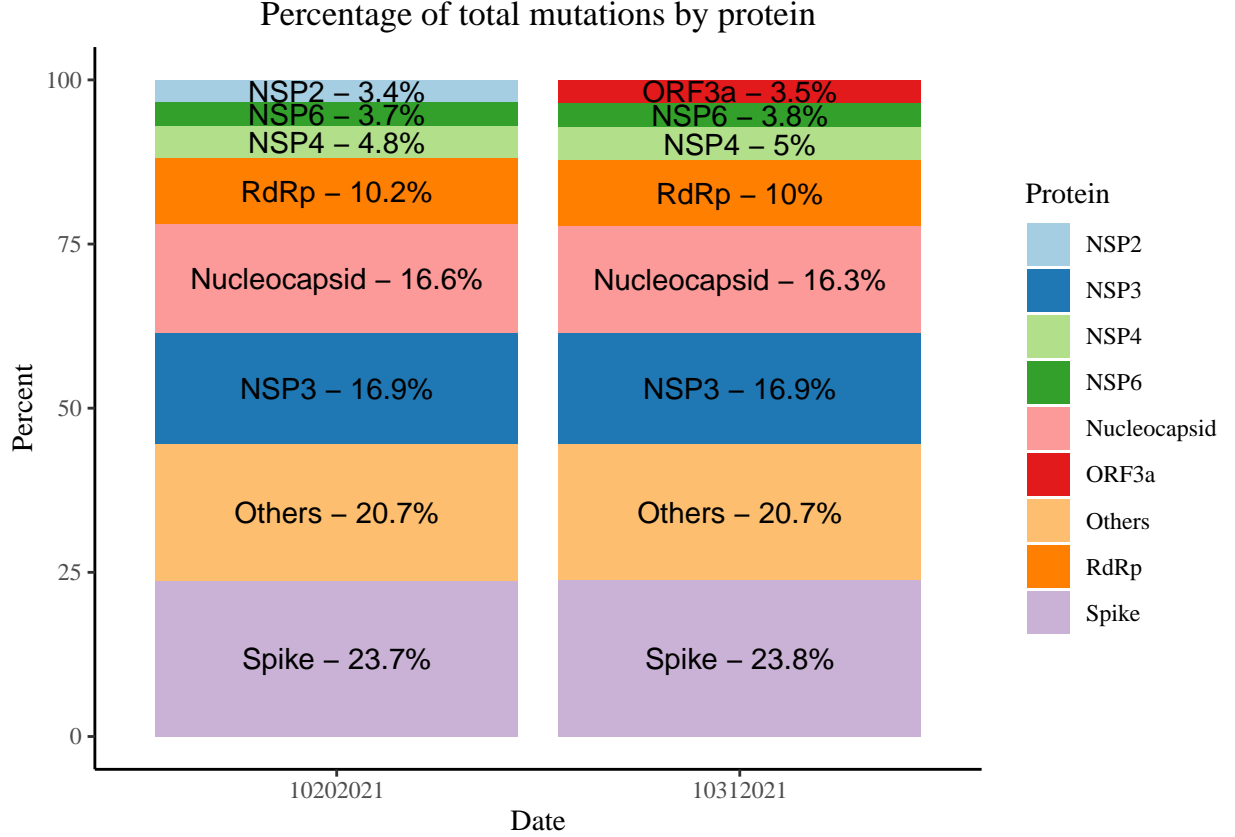


Figure 3: Comparison of total mutation percentages for two collected dates (listed in MMDDYYYY format). For each bar, only the 7 proteins with the largest percentages are named. (author W.S.)

ORF8, ORF7a, and NSP6 have an unusually high number of mutations with respect to their length. This corroborates our previous observations of the mutation distribution graph provided by Figure 2.

### 2.3 Mutation Percentage Comparison

From our summary statistics, we generated a graphical breakdown of the total mutation percentage. Figure 3 displays this data for the top 7 proteins between 20 October 2021 and 31 October 2021. Following from our analysis of the summary statistics, the spike protein made up roughly 24% of the total recorded mutation frequency. The graph changes little between the 10 days comprising each time point. Of note is that a full 13 proteins make up less than 1% each of the total.

## 3 Discussion

Our analysis successfully replicated the original plot of the mutation distribution represented by Figure 1. Our version, Figure 2, makes use of the updated data provided from the Mutation Tracker website and recapitulates the features observed there. Along with the summary statistics from Table 1, we corroborated the significance of the spike protein in the current pattern of SARS-CoV-2 mutation.

Our approach is limited by the scope of the data used in the study. The Mutation Tracker website only allows downloading summary data from the most recent time point. While we had preserved an earlier archive from 20 October 2021, a more thorough comparison than that seen in Figure 3 would use a greater range of dates. Further study into the change in mutation over time, either by sampling the tracker or by directly processing the source data from GISAID, could provide more insight.

The summary statistics from Table 1 suggest differences in mutation characteristics between the 26 studied proteins. Given our observation of relatively more mutations in shorter proteins, a conformational analysis could further assess the influence of a protein’s size and 3D structure on the number of mutations it experiences.

There is also interest in determining the extent, if any, of evolutionary pressure on mutation rate. Some of the proteins comprising the lowest percentages of the total mutation frequency, such as NSP11 and the envelope protein, are known to have important roles in the viral life cycle<sup>2,3</sup>. Future work could determine if proteins with functions in viral replication versus internal regulation experience more mutations.

## 4 Conclusion

In this paper, we presented a partial replication of an earlier study by Wang, R. *et al.*<sup>1</sup>. We recapitulated their visualization of the genome-wide SARS-CoV-2 mutation distribution. We further analyzed the underlying data in order to summarize the influence of each protein on the overall mutation frequencies. The current global pandemic remains one of the most significant challenges to public health in recent memory. In light of emerging variants of SARS-CoV-2, it is even more important to understand how and where the virus mutates.

## 5 Methods

### 5.1 Data Collection and Preprocessing

The original paper<sup>1</sup> used 45,494 complete genome sequences of SARS-CoV-2 strains from infected individuals. These sequences were sourced from the GISAID (<https://www.gisaid.org/>) database up to 11 September 2020. For this analysis, we used the 31 October 2021 summary data from the Mutation Tracker website. The tracker is continually updated with the latest data from GISAID.

Data preprocessing was accomplished using R<sup>4</sup> and the dplyr<sup>5</sup> package. The 26 .csv files for each named protein were parsed and combined into a single data frame in order of position on the SARS-CoV-2 genome. Mutation frequencies were mapped to the corresponding protein name using nucleotide positions sourced from the NIH (<https://www.ncbi.nlm.nih.gov/gene/>).

### 5.2 Plots and Figures

The plot of the genome-wide SARS-CoV-2 mutation distribution (Figure 2) was generated using the Plotly package for R<sup>6</sup>. The stacked bar chart comparing per-protein mutation percentages (Figure 3) was generated using the ggplot2 package<sup>7</sup> using the same data as the mutation distribution plot (summarized for each protein).

## Declarations

### Data and Code Availability

This paper uses the summary data from the Mutation Tracker website up to 31 October 2021. This data was originally sourced from GISAID (<https://www.gisaid.org/>).

All code for this paper is attached in the appendix, and will be uploaded to the authors’ GitHub pending approval.

### Acknowledgements

The authors are grateful to Rui Wang and Guo-Wei Wei for their correspondence providing the color values used in the Mutation Tracker and describing the approach used to clean the summary data.

### Author Contributions

W.S. downloaded and preprocessed the summary data from the Mutation Tracker website. W.S. plotted the mutation frequencies from the summary data to replicate Figure S1 from the original paper. W.S. conducted further analysis of the dataset to summarize the mutation data for each protein.

B.M. downloaded and translated reference genomes for related coronaviruses into individual proteins. B.M. conducted multiple sequence alignment for each protein and calculated percent similarity in order to replicate Figures S5, S7, S8, S10, and S13. Additionally, B.M. added a table summarizing substitution mutation frequencies.

S.S. conducted multiple sequence alignment and single-nucleotide polymorphism (SNP) calling on the complete genome sequences to create SNP profiles for each genome, and used a Jaccard distance-based representation of the SNP variants as input features into a K-means clustering algorithm. S.S. plotted the distributions of the resulting clusters on a U.S. map to replicate Figure 1 from the original paper.

All authors have read and agreed to the published version of the manuscript.

## Competing Interests

The authors declare no competing interests.

## References

1. Wang, R. *et al.* [Analysis of SARS-CoV-2 mutations in the united states suggests presence of four substrains and novel variants](#). *Communications Biology* **4**, 228 (2021).
2. Zhang, M. *et al.* [Structural biology of the arterivirus nsp11 endoribonucleases](#). *Journal of virology* **91**, e01309–16 (2016).
3. Schoeman, D. & Fielding, B. C. [Coronavirus envelope protein: Current knowledge](#). *Virology Journal* **16**, 69 (2019).
4. R Core Team. [R: A language and environment for statistical computing](#). (R Foundation for Statistical Computing, 2021).
5. Wickham, H., François, R., Henry, L. & Müller, K. [Dplyr: A grammar of data manipulation](#). (2021).
6. Sievert, C. *et al.* [Plotly: Create interactive web graphics via plotly.js](#). (2021).
7. Wickham, H. *et al.* [ggplot2: Create elegant data visualisations using the grammar of graphics](#). (2021).
8. Xie, Y. [Bookdown: Authoring books and technical documents with r markdown](#). (2021).
9. Xie, Y. [formatR: Format r code automatically](#). (2021).
10. Xie, Y. [Knitr: A general-purpose package for dynamic report generation in r](#). (2021).
11. Ushey, K., Allaire, J. & Tang, Y. [Reticulate: Interface to python](#). (2021).
12. Allaire, J. *et al.* [Rticles: Article formats for r markdown](#). (2021).
13. Xie, Y. [Bookdown: Authoring books and technical documents with R markdown](#). (Chapman; Hall/CRC, 2016).
14. Wickham, H. [ggplot2: Elegant graphics for data analysis](#). (Springer-Verlag New York, 2016).
15. Xie, Y. [Dynamic documents with R and knitr](#). (Chapman; Hall/CRC, 2015).
16. Xie, Y. [Knitr: A comprehensive tool for reproducible research in R](#). in *Implementing reproducible computational research* (eds. Stodden, V., Leisch, F. & Peng, R. D.) (Chapman; Hall/CRC, 2014).

17. Sievert, C. *Interactive web-based data visualization with r, plotly, and shiny*. (Chapman; Hall/CRC, 2020).

## A All Code

```
# Required packages
library(knitr, quietly = TRUE)
library(formatR, quietly = TRUE)
library(rtables, quietly = TRUE)
library(dplyr, quietly = TRUE)
library(reticulate, quietly = TRUE)
library(plotly, quietly = TRUE)
library(bookdown, quietly = TRUE)

# Wrap R code in final document
knitr::opts_chunk$set(cache = TRUE, echo = FALSE, message = FALSE, error = FALSE,
  fig.align = "center", out.width = "\\textwidth", tidy.opts = list(width.cutoff = 80),
  tidy = TRUE)
knitr::include_graphics("./figures/original_distribution.png")
# Read latest mutation summary data into a combined data frame Data retrieved
# from https://users.math.msu.edu/users/weig/SARS-CoV-2_Mutation_Tracker.html
path <- "./data/MutationSummary_10312021/"
files <- dir(path)
mutation_summaries <- do.call(rbind, lapply(paste0(path, files), read.csv))

# Add column for raw nucleotide position
mutation_summaries <- mutation_summaries %>%
  mutate(position = as.integer(gsub("[^0-9]", "", mutation_site))) %>%
  arrange(position)

# Create new data frame to track the combined frequencies of each unique
# mutation
mutation_summaries_combined <- mutation_summaries %>%
  group_by(position) %>%
  summarise(frequency = sum(Total_frequency))

# Use nucleotide locations from https://www.ncbi.nlm.nih.gov/gene/ to clean
# data Protein name mapping adapted from original paper (Python code): def
# mapDict(self): keys = ['266:805', '[806:2719]', '[2720:8554]',
# '[8555:10054]', '[10055:10972]', '[10973:11842]', '[11843:12091]',
# '[12092:12685]', '[12686:13024]', '[13025:13441]', '[13442:16236]',
# '[16237:18039]', '[18040:19620]', '[19621:20658]', '[20659:21552]',
# '[13442:13480]', '[21563:25384]', '[25393:26220]', '[26245:26472]',
# '[26523:27191]', '[27202:27387]', '[27394:27759]', '[27756:27887]',
# '[27894:28259]', '[28274:29533]', '[29558:29674]'] values = ['NSP1', 'NSP2',
# 'NSP3', 'NSP4', '3CL', 'NSP6', 'NSP7', 'NSP8', 'NSP9', 'NSP10', 'RdRp',
# 'Helicase', 'Exonuclease', 'endoRNase', '2'-O-ribose MTases',
# 'NSP11', 'Spike', 'ORF3a', 'Envelope', 'Membrane', 'ORF6', 'ORF7a', 'ORF7b',
# 'ORF8', 'Nucleocapsid', 'ORF10'] nameMap = dict(zip(keys,values)) return
# nameMap
name_map <- rep("Others", 30000)
name_map[266:805] <- "NSP1"
name_map[806:2719] <- "NSP2"
name_map[2720:8554] <- "NSP3"
name_map[8555:10054] <- "NSP4"
name_map[10055:10972] <- "3CL"
name_map[10973:11842] <- "NSP6"
```



```

name_map[11843:12091] <- "NSP7"
name_map[12092:12685] <- "NSP8"
name_map[12686:13024] <- "NSP9"
name_map[13025:13441] <- "NSP10"
name_map[13442:16236] <- "RdRp"
name_map[16237:18039] <- "Helicase"
name_map[18040:19620] <- "Exonuclease"
name_map[19621:20658] <- "endoRNase"
name_map[20659:21552] <- "2'-O-ribose MTases"
name_map[13442:13480] <- "NSP11"
name_map[21563:25384] <- "Spike"
name_map[25393:26220] <- "ORF3a"
name_map[26245:26472] <- "Envelope"
name_map[26523:27191] <- "Membrane"
name_map[27202:27387] <- "ORF6"
name_map[27394:27759] <- "ORF7a"
name_map[27756:27887] <- "ORF7b"
name_map[27894:28259] <- "ORF8"
name_map[28274:29533] <- "Nucleocapsid"
name_map[29558:29674] <- "ORF10"
mutation_summaries_combined$protein <- name_map[mutation_summaries_combined$position]

# Remove rows with default protein name
mutation_summaries_combined <- subset(mutation_summaries_combined, protein != "Others")
# Color values adapted from settings used in original paper (Python code):
# color_continuous_scale=[ 'rgb(160,200,255)', 'yellow', 'rgba(13,143,129,0.7)',
# 'rgba(119,74,1750,0.8)', 'rgba(235,74,64,0.8)', 'rgb(62,109,178)',
# 'rgb(160,200,255)', 'rgb(253,174,107)', 'rgba(119,74,1750,0.8)',
# 'rgb(160,200,255)', 'black', 'rgba(13,143,129,0.7)', 'rgba(119,74,1750,0.8)',
# 'rgba(235,74,64,0.8)', 'rgb(62,109,178)', 'rgb(160,200,255)', 'black',
# 'rgb(260,200,255)', 'rgb(160,200,255)', 'yellow', 'rgba(13,143,129,0.7)',
# 'rgba(119,74,1750,0.8)', 'rgba(235,74,64,0.8)', 'rgb(62,109,178)',
# 'rgb(160,200,255)', 'rgb(253,174,107)']
scatter_palette <- c("#a0c8ff", "yellow", "#0d8f81", "#774ad6", "#eb4a40", "#3e6db2",
  "#a0c8ff", "#fdae6b", "#774ad6", "#a0c8ff", "black", "#0d8f81", "#774ad6", "#eb4a40",
  "#3e6db2", "#a0c8ff", "black", "#04c8ff", "#a0c8ff", "yellow", "#0d8f81", "#774ad6",
  "#eb4a40", "#3e6db2", "#a0c8ff", "#fdae6b")

# Create scatter plot of mutations
scatter_fig <- plot_ly(data = mutation_summaries_combined, x = ~position, y =
  ~log(frequency),
  type = "scatter", mode = "markers", color = ~protein, colors = scatter_palette)
scatter_fig <- scatter_fig %>%
  layout(title = "Mutation distribution across the SARS-CoV-2 genome", xaxis =
    ~list(title = "",
    showgrid = FALSE, tickangle = 60, tickmode = "array", nticks = 11, tickvals =
    ~c(536,
    5638, 9305, 10514, 11408, 14839, 17138, 23474, 25807, 26359, 28904),
    ticktext = c("NSP1", "NSP3", "NSP4", "3CL", "NSP6", "RdRp", "Helicase", "S",
    "ORF3a", "E", "N")), yaxis = list(title = "ln(Frequency)", showgrid = FALSE),
    font = list(family = "Serif", size = 32, color = "black"), showlegend = FALSE,
    margin = list(t = 200))
save_image(scatter_fig, "./figures/updated_distribution.pdf", width = 2304, height =
  ~1152)
knitr::include_graphics("./figures/updated_distribution.pdf")
# Generate mutation statistics by protein
aggregate_summaries <- mutation_summaries_combined %>%
  group_by(protein) %>%

```



```

summarise(total_mutations = sum(frequency)) %>%
mutate(raw_percent = (total_mutations/sum(total_mutations)) * 100, nucleotide_length
↪ = sapply(protein,
  switch, NSP1 = {
    805 - 266
  }, NSP2 = {
    2719 - 806
  }, NSP3 = {
    8554 - 2720
  }, NSP4 = {
    10054 - 8555
  }, `3CL` = {
    10972 - 10055
  }, NSP6 = {
    11842 - 10973
  }, NSP7 = {
    12091 - 11843
  }, NSP8 = {
    12685 - 12092
  }, NSP9 = {
    13024 - 12686
  }, NSP10 = {
    13441 - 13025
  }, RdRp = {
    16236 - 13442
  }, Helicase = {
    18039 - 16237
  }, Exonuclease = {
    19620 - 18040
  }, endoRNase = {
    20658 - 19621
  }, `2'-O-ribose MTases` = {
    21552 - 20659
  }, NSP11 = {
    13480 - 13442
  }, Spike = {
    25384 - 21563
  }, ORF3a = {
    26220 - 25393
  }, Envelope = {
    26472 - 26245
  }, Membrane = {
    27191 - 26523
  }, ORF6 = {
    27387 - 27202
  }, ORF7a = {
    27759 - 27394
  }, ORF7b = {
    27887 - 27756
  }, ORF8 = {
    28259 - 27894
  }, Nucleocapsid = {
    29533 - 28274
  }, ORF10 = {
    29674 - 29558
  }
), mutations_per_nucleotide = total_mutations/nucleotide_length) %>%
  arrange(desc(total_mutations))
knitr::kable(aggregate_summaries, format = "latex", booktabs = TRUE, digits = 3,

```

```

col.names = c("protein", "total mutations", "% total mutations", "protein length",
  "mutations/nucleotide"), caption = "Summary statistics for each SARS-CoV-2
  ↪ protein (author W.S.)",
  format.args = list(big.mark = ","))
# Read mutation summary data for 10202021
path <- "./data/MutationSummary_10202021/"
files <- dir(path)
mutation_summaries_10202021 <- do.call(rbind, lapply(paste0(path, files), read.csv))

# Find percent mutations by protein for 10202021
bar_data_10202021 <- mutation_summaries_10202021 %>%
  mutate(position = as.integer(gsub("[^0-9]", "", mutation_site))) %>%
  group_by(position) %>%
  summarise(frequency = sum(Total_frequency))
bar_data_10202021$protein <- name_map[bar_data_10202021$position]
bar_data_10202021 <- subset(bar_data_10202021, protein != "Others")
bar_data_10202021 <- bar_data_10202021 %>%
  group_by(protein) %>%
  summarise(total_mutations = sum(frequency)) %>%
  mutate(raw_percent = (total_mutations/sum(total_mutations)) * 100) %>%
  arrange(desc(total_mutations)) %>%
  select(protein, raw_percent) %>%
  head(7)
others <- list("Others", 100 - sum(bar_data_10202021$raw_percent))
bar_data_10202021 <- bar_data_10202021 %>%
  rbind(others) %>%
  arrange(raw_percent) %>%
  mutate(date = "10202021")

# Find percent mutations by protein for latest data (10312021)
bar_data_10312021 <- aggregate_summaries %>%
  select(protein, raw_percent) %>%
  head(7)
others <- list("Others", 100 - sum(bar_data_10312021$raw_percent))
bar_data_10312021 <- bar_data_10312021 %>%
  rbind(others) %>%
  arrange(raw_percent) %>%
  mutate(date = "10312021")

# Combine data into single data frame for plotting
bar_data <- rbind(bar_data_10202021, bar_data_10312021)

# Created stacked bar chart for each date ordered by percent
stacked_bar <- ggplot(bar_data, aes(x = date, y = raw_percent, group = raw_percent)) +
  geom_col(aes(fill = protein), width = 0.9) + scale_fill_brewer(type = "qual",
  palette = "Paired") + geom_text(aes(label = paste0(protein, " - ", round(raw_percent,
  digits = 1), "%")), position = position_stack(vjust = 0.5)) + ggtitle("Percentage of
  ↪ total mutations by protein") +
  xlab("Date") + ylab("Percent") + labs(fill = "Protein") + theme_classic(base_family =
  ↪ "serif") +
  theme(plot.title = element_text(hjust = 0.5))
ggsave("./figures/stacked_bar.pdf")
knitr::include_graphics("./figures/stacked_bar.pdf")
# Cite packages
knitr::write_bib(c(.packages(), "bookdown"), "packages.bib")
sessionInfo(package = NULL)

```

## B Session Info

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] bookdown_0.24  plotly_4.10.0  ggplot2_3.3.5  reticulate_1.22
## [5] dplyr_1.0.7    rtables_0.21   formatR_1.11   knitr_1.36
##
## loaded via a namespace (and not attached):
## [1] tidyselect_1.1.1  xfun_0.27      purrr_0.3.4    lattice_0.20-45
## [5] colorspace_2.0-2  vctrs_0.3.8    generics_0.1.0  htmltools_0.5.2
## [9] viridisLite_0.4.0  yaml_2.2.1     utf8_1.2.2     rlang_0.4.12
## [13] pillar_1.6.4      glue_1.4.2     withr_2.4.2     DBI_1.1.1
## [17] RColorBrewer_1.1-2 lifecycle_1.0.1 stringr_1.4.0   munsell_0.5.0
## [21] gtable_0.3.0      htmlwidgets_1.5.4 codetools_0.2-18 evaluate_0.14
## [25] labeling_0.4.2     fastmap_1.1.0   crosstalk_1.2.0 fansi_0.5.0
## [29] Rcpp_1.0.7         scales_1.1.1    jsonlite_1.7.2  farver_2.1.0
## [33] png_0.1-7          digest_0.6.28   stringi_1.7.5   grid_4.1.1
## [37] rprojroot_2.0.2    here_1.0.1      tools_4.1.1     magrittr_2.0.1
## [41] lazyeval_0.2.2     tibble_3.1.5    crayon_1.4.1    tidyr_1.1.4
## [45] pkgconfig_2.0.3    ellipsis_0.3.2  Matrix_1.3-4    data.table_1.14.2
## [49] assertthat_0.2.1   rmarkdown_2.11  httr_1.4.2      R6_2.5.1
## [53] compiler_4.1.1
```