Supplementary material of

"Persistent minimal sequences of SARS-CoV-2"

D. Pratas[1,2,3] and J. M. Silva[1,2]

[1]IEETA and [2]DETI, University of Aveiro
[3]Department of Virology, University of Helsinki

1 Pipeline of the analysis

Figure S1 depicts the pipeline used in the analysis of the results presented in the article. The input data is constituted by 93 SARS-CoV-2 whole genomes and the human reference whole genome and transcriptome. Additional EBOV and other coronaviruses have been used for comparison purposes. All the materials, including the accession numbers and automatic scripts for downloading the data, are available in Supplementary Section 8.

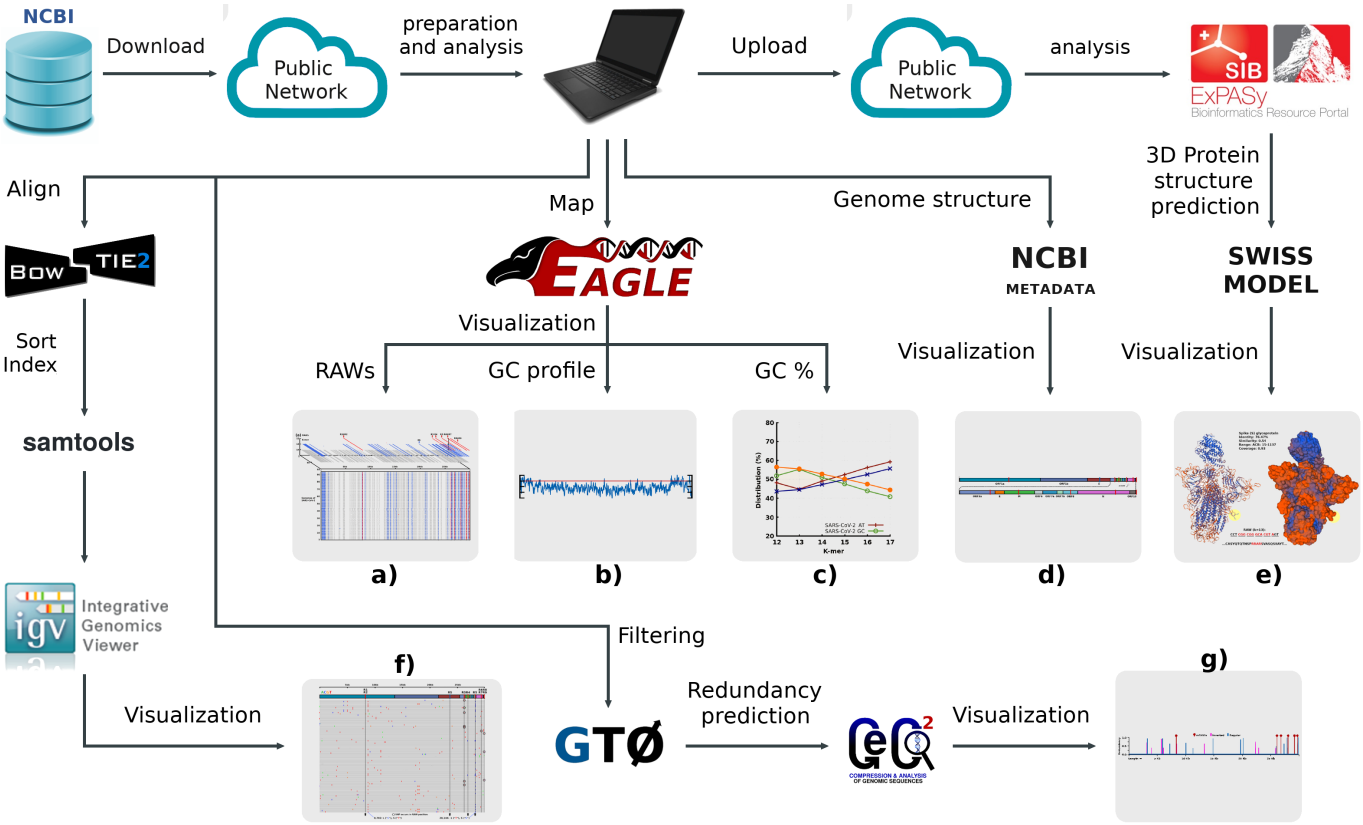

Figure S1: Pipeline of the analysis; a) visualization of the identified mRAWs by EAGLE; b) visualization of the GC average profile computed by EAGLE; c) visualization of GC percentage computed by EAGLE; d) visualization of the genome structure created with metadata from the NCBI; e) visualization of the protein structure prediction using SWISS-MODEL [1] via the ExPASy [2] and identification of the regions in the structures containing the PmRAWs; f) visualization of the alignments according to the SARS-CoV-2 reference using IGV [3] after alignment with Bowtie2 [4] and sort/index with samtools [5]; g) visualization of the redundancy profiles computed with GeCo2 [6] via the GTO [7].

After download and preparation (according to Sup. Section 8), the mRAWs and PmRAWs are identified automatically using EAGLE. Additionally, the average GC profiles and the GC percentage are computed by EAGLE using a single run, which are required to study a new pattern associated with the GC (Guanine Cytosine) content distribution in the RAWs and PmRAWs (described in the article). Detailed information about the EAGLE tool is available at Supplementary Section 6.

In order to localize the RAWs and PmRAWs in the SARS-Cov-2 genome structure, we computed a genome structure map using the NCBI metadata and aligned it according to the RAWs and PmRAWs positional maps. This methodology enables the identification of genes where the PmRAWs exist. This information is then used to predict the protein structure associated with the gene, such as the Spike protein, and, finally, to localize the region where the PmRAWs occur in the structure.

To understand the local redundancy of SARS-CoV-2, we computed the redundancy profiles. Additionally, to localize conserved regions, we computed the whole alignments of each of the 93 SARS-CoV-2 genomes against the SARS-CoV-2 reference. Both redundancy profiles and alignments incorporated the positions where the RAWs occurred. The reason for this is to confirm the conservation of the RAWs and to identify if redundant or non-conserved DNA constituted the flanking regions. The latter helps to understand if extensions to the sequence of the RAWs (or PmRAWs) can be made.

Additional information, namely the computer characteristics and materials used in the analysis, are available in Supplementary Section 7 and 8, respectively.

2 Super PmRAW

A super PmRAW is an intersection of two PmRAWs according to their positions in the genome. Let p_1 and p_2 be PmRAWs from a set of PmRAWs, P. Let also b_1 and b_2 be the positions in the sequence x where these PmRAWs occur. We say that p_1 and p_2 are a super PmRAW if $|b_1 - b_2| < k$, where k is the size of the PmRAWs. Accordingly, the super PmRAW has the length of $|b_1 - b_2| + k$ and the initial position at the genome starts at the position given by $\min\{b_1, b_2\}$.

3 Inversions of RAWs

Inversions, sometimes referred as inverted repeats (or reverse complements), are sequences that occur in the complementary chain of a DNA or RNA sequence according to the Chargaff's rules. For example, the sequence "ACGTTATCG" has the inversion of "CGATAACGT".

Since inversions can be associated with chromosomal rearrangements [8] and we use the reference human genome to map the associated RAWs at the SARS-CoV-2 genomes, a study to assess the impact of using inversions is needed. For example, if an inversion rearrangement occurs in the human genome, without considering inversions in EAGLE computations the number of RAWs will be overestimated. Notice that by default EAGLE considers inversions.

Accordingly, we run EAGLE to detect the RAWs on the SARS-CoV-2 genome using two approaches, one considers inversions while the other does not. Additionally, we include the same study using EBOV (Ebolavirus) for comparison purposes. The results are present in Supplementary Figure S2. As shown it can be seen, when the inversions are not considered, the number of RAWs is overestimated for both viral genomes.

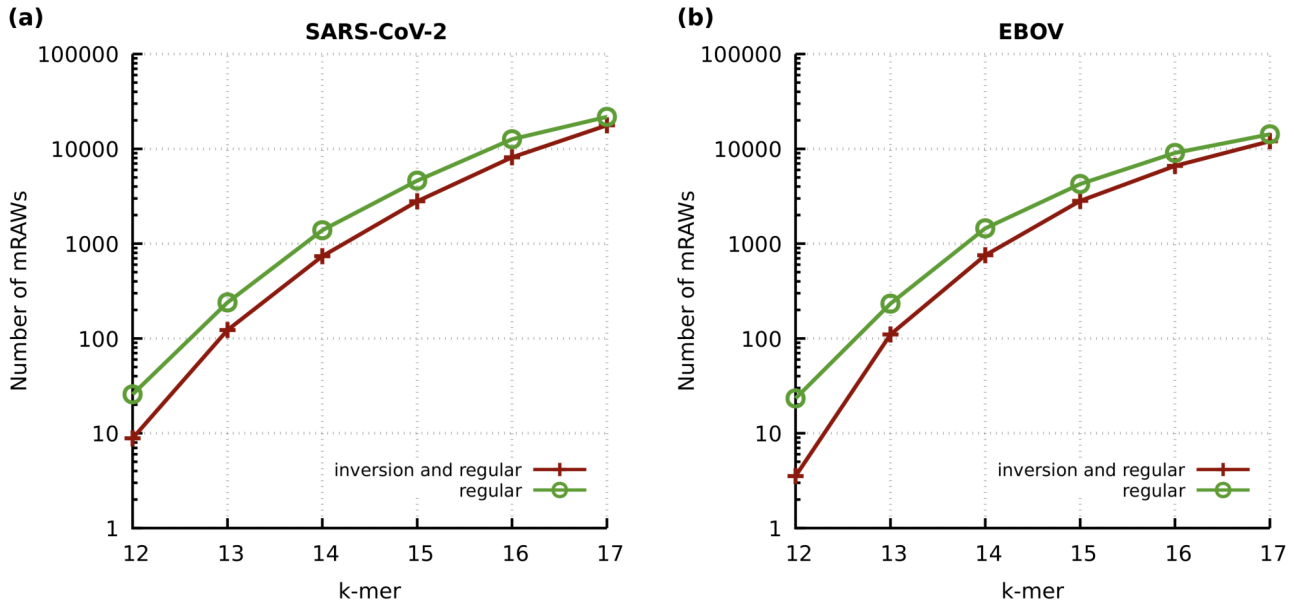

Figure S2: SARS-CoV-2 (left) an EBOV (right) minimal absent words for different k (with and without inversions) relatively to the human complete genome and transcriptome. Logarithmic scale is applied to y-axis.

4 Supplementary tables and figures for the analysis

This section includes Supplementary tables and figures to the article, namely tables 1, 2, and figures S3, S4, S5, S6, S7, S8, S9, S10.

k-mer	average	variance	std. dev.	#A	#C	#G	#T
12	8.85	0.26	0.51	2093	2289	2834	2660
13	122.61	2.13	1.46	31117	37582	44199	35341
14	736.65	5.78	2.40	222535	238188	251048	247341
15	2794.59	15.10	3.89	972377	911918	941836	1072324
16	8146.67	71.06	8.43	3275311	2615937	2700114	3530878
17	17694.37	164.90	12.84	8013026	5568721	5843853	8549192

Table S1: Average, variance, standard deviation (std. dev.), and sum of the number of symbols of each base (#A, #C, #G, #T) for the different k-mers of the SARS-CoV-2 Relative Absent Words (with inversions).

k-mer	average	variance	std. dev.	#A	#C	#G	#T
12	3.53	2.54	1.59	1629	2418	1509	1440
13	110.32	192.12	13.86	57560	69444	61764	47858
14	752.73	3374.51	58.09	452406	482396	433780	370218
15	2823.59	38597.17	196.46	1920515	1848732	1655006	1564142
16	6614.15	179618.94	423.81	5029011	4328353	3948972	4155024
17	12008.97	594252.60	770.88	10184043	7805879	7144505	8550733

Table S2: Average, variance, standard deviation (std. dev.), and sum of the number of symbols of each base (#A, #C, #G, #T) for the different k-mers of the EBOV Relative Absent Words (with inversions).

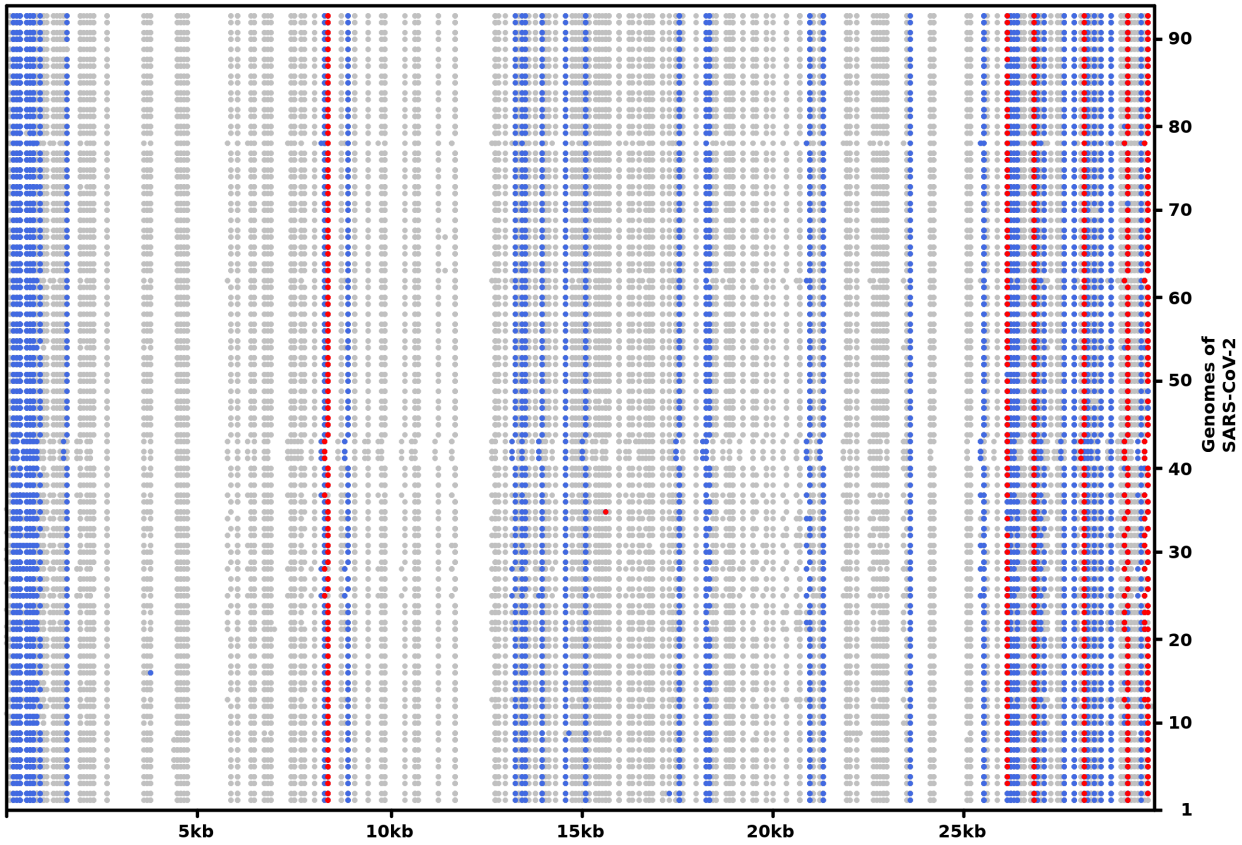

Figure S3: SARS-CoV-2 minimal absent words relatively to the human genome and transcriptome. RAWs were identified in 93 unaligned genomes from the current outbreak from different countries using EAGLE. RAWs are highlighted in red (k=12, arrows), blue (k=13) and grey (k=14). This image is the upper vertical projection of Figure 1a from the manuscript.

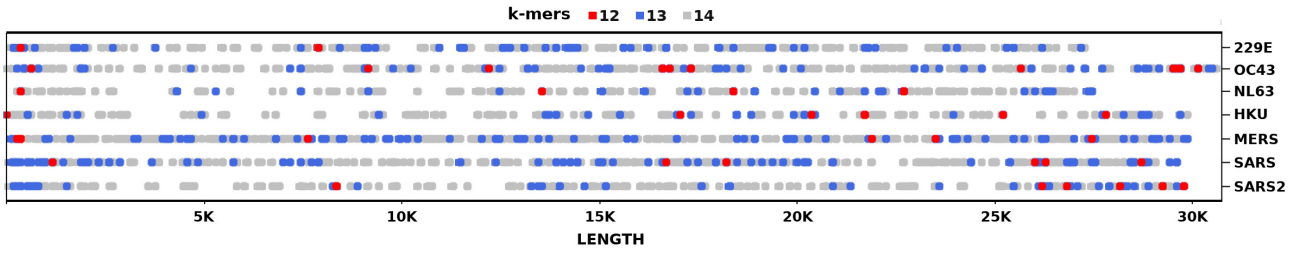

Figure S4: Minimal absent words for different k (12, 13, and 14) in several human *Coronaviruses* reference sequences, namely SARS2 (SARS-CoV-2), SARS, MERS, HKU, NL63, OC43, and 229E, relative to the human genome and transcriptome.

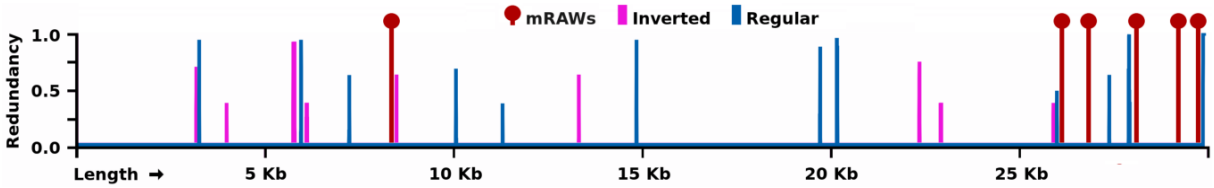

Figure S5: Redundancy profiles for regular and inverted regions according to the positions of the mRAWs. The profiles were created with GTO [7] using GeCo2 [6] with a context model and a substitution tolerant context model [9] of 13. Parameters: 13:500:2:0:0.95/3:100:0.95 with a filter window size of 2.

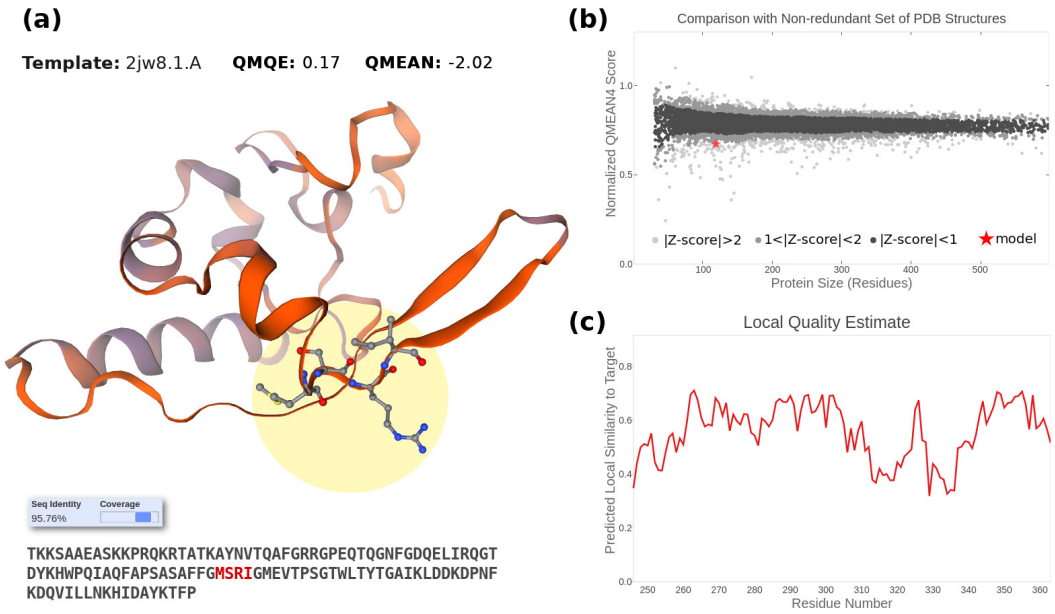

Figure S6: SARS-CoV-2 Nucleocapsid protein (N) structure simulated with SWISS-MODEL [1] via the ExPASy [2]. (a) aminoacid sequence corresponding to R6 / R7 mRAWs in red (MSRI) and a yellow background on the prediction. (b) Comparison with non-redundant set of structures (z-score). (c) local quality estimate of the model.

Figure S7: SARS-CoV-2 membrane glycoprotein (M) partial structure simulated with SWISS-MODEL [1] via the ExPASy [2]. The aminoacid sequence corresponding to the R4 mRAW is in red (ART) and a yellow background on the prediction.

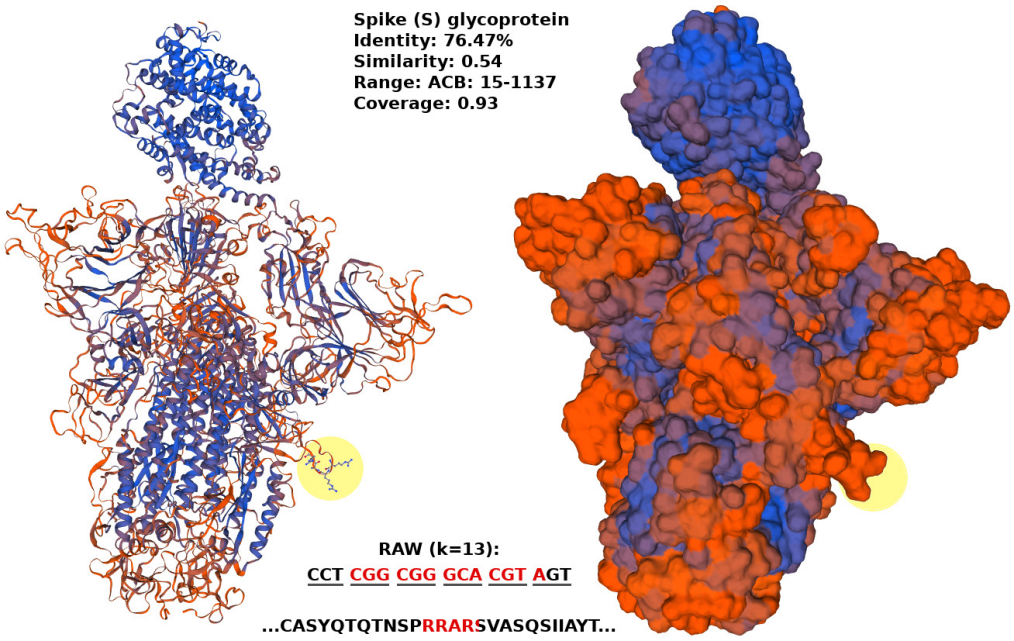

Figure S8: SARS-CoV-2 spike glycoprotein (S) partial structure simulated with SWISS-MODEL [1] via the ExPASy [2] during prefusion conformation. The aminoacid sequence corresponding to the RAW is in red (RRAR) and a yellow background on the prediction.

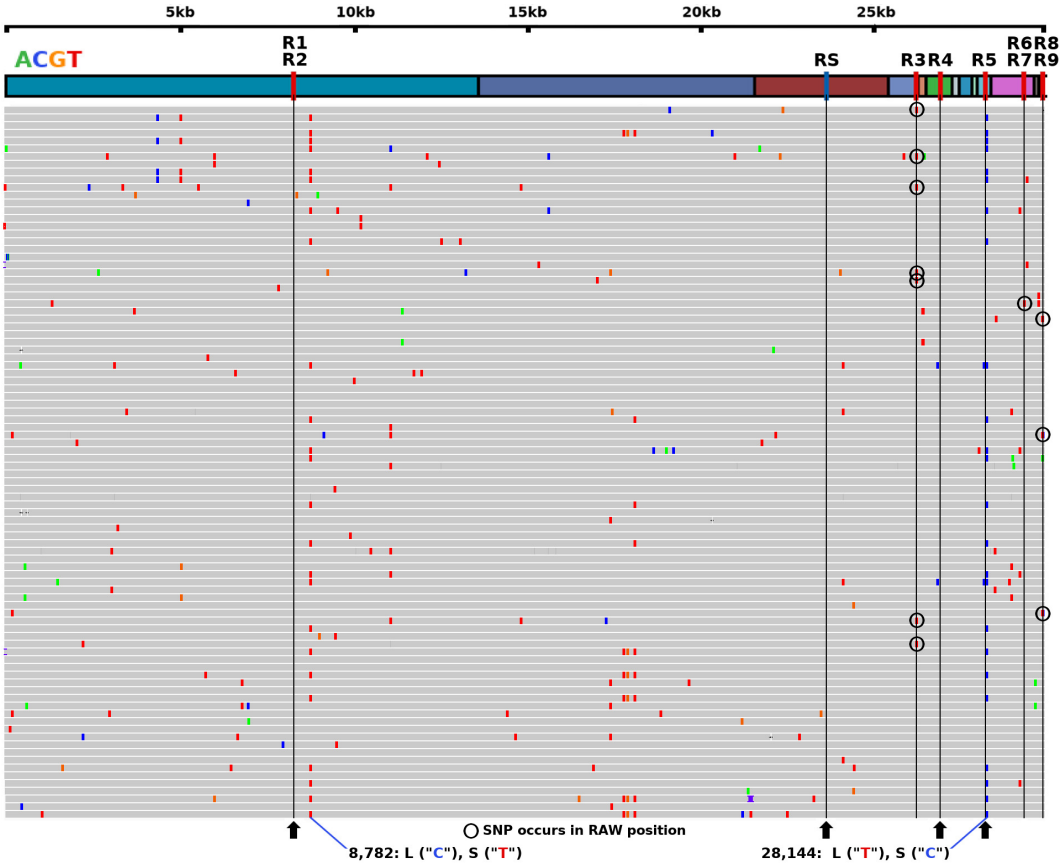

Figure S9: Whole SARS-CoV-2 genomes Single Nucleotide Polymorphisms (SNPs) aligned with minimal absent words relatively to the human complete genome and transcriptome. Length of genome is represented in the x-axis, while the order of each genome in the y-axis. Persistent RAWs are marked with an arrow. The alignments were performed with Bowtie2 [4], the indexing and sorting with Samtools [5], and the SNPs map was extracted from IGV [3] and aligned with the structure and EAGLE output. L and S stands for SARS-CoV-2 genomes with different severity characteristics that express different SNPs in the respective positions in concordance with [10]. In running commands the procedure is detailed.

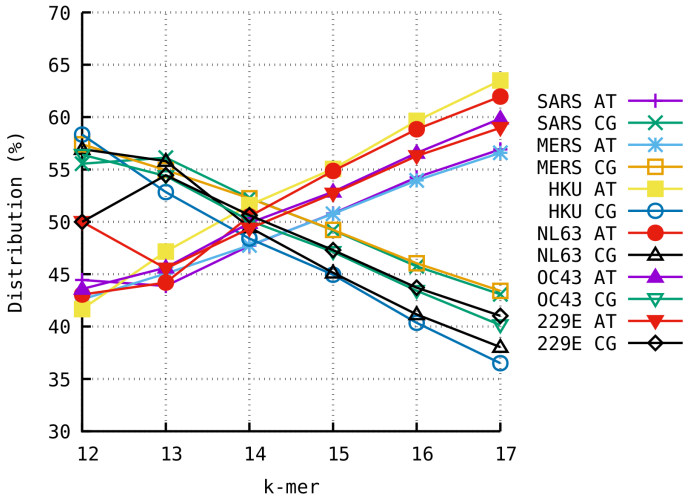

Figure S10: Distribution of GC/AT percentage of RAWs for different k in several human *Coronaviruses* reference sequences: SARS, MERS, HKU, NL63, OC43, and 229E.

5 Human coronavirus mRAWs

The following slots show the mRAWs for several human coronavirus, namely the SARS-CoV-2 and the reference sequences of SARS, MERS, HKU, NL63, OC43, and 229E.

SARS-CoV-2 ($k = 12$):

```
8296  TGCGCGTCATAT # Persistent
8297  GCGCGTCATATT # Persistent
26079 CACAATCGACGG
26776 TTGCGCGTACGC # Persistent
28061 CGATATCGGTAA # Persistent
29167 AATGTCGCGCAT
29170 GTCGCGCATTGG
29693 GTACGATCGAGT
29696 CGATCGAGTGTA
```

SARS ($k = 12$):

```
1157  GGCGTATACGCT
16629 CGCGAAGTACTC
18210 CGTCACGTTCGT
26005 CACAATCGACGG
26289 TCTACTCGCGTG
28688 GTAGTCGCGGTA
```

MERS ($k = 12$):

```
248 TCGTCCGGTGCG
313 CGCGCGGTACGT
314 GCGCGGTACGTA
315 CGCGGTACGTAT
317 CGGTACGTATCG
7628  TAACGCTACGGA
21866 CGCTACTATACG
23525 CCGTTCTACGCG
27460 AACGCGCGATTC
```

HKU ($k = 12$):

```
16   ACGTTCGTACCG
17   CGTTCGTACCGT
16989 TTACCGCGCTAC
20341 GGTAATTGCGCG
21697 GTTCGCGATACG
21698 TTCGCGATACGC
21699 TCGCGATACGCG
21700 CGCGATACGCGT
```

```
25207 CGTCGTAAACGT
27752 TACGAGCCGTAG
```

NL63 ($k = 12$):

```
363 TAGCCGTTCGCA
13563 CGTTGATCAACG
18356 TTCGCGCTAGTA
22705 TTCGTCCGCGTA
22706 TCGTCCGCGTAA
22707 CGTCCGCGTAAT
```

OC43 ($k = 12$):

```
599 GTTTCGGCGTAG
9086  ACGTATCGTGCG
12134 TGAACGCGACCG
16576 TGAGCGCGAATT
16579 GCGCGAATTGAT
16726 GTATTATCGCGC
17267 ATTCGCGCTAAG
17269 TCGCGCTAAGCA
25664 CTATAGCGGTCG
29489 GTACGGCACCGA
29561 TGTCGATCGGGA
29715 TCGCGTAGTAGA
30133 GCTATAACGGCG
```

229E ($k = 12$):

```
377 CGCCGTTATAGC
7840  GTACGATATCGT
```

6 EAGLE implementation

Consider a target sequence, x, and a reference sequence, y, both drawn from the finite alphabet Θ. In order to map the presence of RAWs in a sequence x, we compute a binary sequence reporting its presence or absence along the sequence, using the model of y exclusively. As such, a relative uniqueness profile is given by

$$U(x_i \| y), \tag{1}$$

where U does not need to respect causality. Therefore, after loading the model from y and freezing, we can access to any i.

For a biomedical application, we use a large reference sequence, namely one corresponding to the conjoint human genome and transcriptome, while the target (or targets) are usually tiny (viruses). Since $y \gg x$, for detecting MRAWs the algorithm will spend most of its time loading each k-mer of y, for $k \in \{k_1, k_2, ..., k_n\}$.

Every k_j is computed according to (1), using a k-mer model that uses a numerical index between 0 and $|\Theta|^{k_j} - 1$. Each index is updated, with the information of presence or absence of each respective word, in a simple table array for $k_j \leq 16$, or a hash table in a larger k_j. Although there is the possibility to search for a $k_j > 16$, in practice the MRAWs are used for $k_j \leq 16$. The memory, Ω, required for k is given by

$$\Omega_k = \chi \sum_{j=k_1}^{k_n} |\Theta|^j, \tag{2}$$

where χ is the precision of the memory. For a common search, having $k = \in \{11, 12, 13, 14\}$, it would be needed 340 MBytes. As it can be seen, $\Omega_k = 340$ MBytes regardless the size of y, in nowadays computers, are very low memory numbers.

On the other hand, the time resource is a demanding task in this situation, namely because there is the need to load each k_j from y (large sequence). Aware of this, we have created a method that uses parallel computing for loading each k-mer model. Therefore, for each k_j we compute it with a thread, T_j. Additional information can be seen in [11].

After loading the reference, the models are kept frozen. Here it starts the detection of RAWs for each target sequence accordingly. In this phase, there is no need for parallel computing since, for practical applications, the sizes of the targets are tiny.

After the identification of the RAWs, the tool performs two extra analysis, namely:

- Computes the GC percentage distribution for each RAW. The GC percentage is given by the number of Cytosine (C) and Guanine (G) bases in a string x with length $|x|$ according to

$$GC(x) = \frac{100}{|x|} \sum_{i=1}^{|x|} \mathcal{N}(x_i || x_i \in \Xi), \tag{3}$$

  where x_i is each symbol of x (assuming causal order), Ξ is a subset alphabet containing the symbols $\{G, C\}$ and $\mathcal{N}$ the program that counts the numbers of symbols in Ξ. Complementary, $AT(x) = 100 - GC(x)$.

- Computes the GC content. The GC content is obtained with Eq 3 using a sliding window of size 10. Then, the average is computed for all the sequences followed by a low-pass filter with a Hamming window of 20 symbols. The filter size can be set to specific numbers.

Additionally, EAGLE provides two output types. One uses a command-line interface in an ASCII format, while the other creates Gnuplot scripts to the creation of automatic plots with the EAGLE results.

7 Software and Hardware

Laptop computer running Linux Ubuntu 18.04 LTS with GCC (`https://gcc.gnu.org`), Conda (`https://docs.conda.io`) and CMake (`https://cmake.org`) installed. The hardware includes 16 GB of RAM, Intel Core i7-8650U CPU @ 1.90GHz × 8, and a 500 GB disk.

8 Reproducibility

8.1 Input data

The input data is constituted by FASTA files with SARS-CoV-2, EBOV, human coronaviruses, and conjoint human genome and transcriptome. The automatic download of the data can be performed running the Shell programs `GetSARS2.sh`, `GetHumanCorona.sh`, `GetEBOV.sh`, `GetHuman.sh`, from the `scripts` folder at the EAGLE2 repository as

```
./GetSARS2.sh
./GetHumanCorona.sh
./GetEBOV.sh
./GetHuman.sh
```

All the genome identifiers are described bellow.

8.1.1 SARS-CoV-2 genomes

```
MT007544, MT126808, NC_045512, MT121215, MT135042, MT135041, MT135044, MT135043, MT123290,
    MT123293, MT123291, MT123292, MT093631, MT049951, MT039873, MT019529, MT019532, MT019531,
    MT019533, MT019530, MN996529, MN996530, MN996531, MN996528, MN996527, MN988669, MN988668,
    MN938384, MN975262, MN908947, MT050493, MT012098, MT066156, MT072688, MT039890, MT093571,
    MT192759, MT066176, MT066175, MT192765, MT188339, MT188340, MT188341, MT184907, MT184913,
    MT184909, MT184908, MT184911, MT184910, MT184912, MT163716, MT163719, MT163718, MT163717,
    MT159721, MT159711, MT159710, MT159708, MT159712, MT159707, MT159715, MT159716, MT159722,
    MT159714, MT159713, MT159706, MT159705, MT159719, MT159709, MT159717, MT159720, MT159718,
    MT152824, MT118835, MT106052, MT106053, MT106054, MT044258, MT044257, MT039887, MT039888,
    MT027063, MT027064, MT027062, MT020880, MT020881, MN994468, MN997409, MN994467, MN988713,
    MN985325, MT192772, MT192773.
```

8.1.2 Human Corona genomes

```
NC_045512, NC_004718.3, NC_019843.3, NC_006577.2, NC_005831.2, NC_006213.1, NC_002645.1.
```

8.1.3 EBOV genomes

```
10313991, 436409439, 692112628, 436409339, 436409349, 436409389, 436409419, 33860540,
    384406804, 436409359, 436409369, 436409379, 436409399, 436409409, 436409429, 436409269,
    436409279, 436409289, 436409299, 436409309, 436409319, 436409329, 355344222, 355344232,
    685509613, 733962878, 733962903, 733962926, 674810549, 674810552, 674810554, 703773102,
    703773112, 703773128, 743615855, 724722348, 732464081, 732464090, 732464099, 732464108,
    661348595, 661348605, 661348615, 661348625, 661348635, 661348645, 661348655, 661348665,
    661348675, 661348685, 661348695, 661348705, 661348715, 661348725, 661348735, 667852489,
    667852500, 667852510, 667852521, 667852531, 667852542, 667852552, 667852562, 667852572,
    667852582, 667852592, 667852604, 667852614, 667852625, 667852635, 667852646, 667852656,
    667852666, 667852676, 667852686, 667852696, 667852706, 667852716, 667852726, 667852736,
    667852747, 667852757, 667852767, 667852778, 667852788, 667852798, 667852809, 667852819,
```

```
    667852830, 667852840, 667852851, 667852861, 667852872, 667852882, 667852893, 667852903,
    667852914, 667852924, 667852934, 667852945, 667852957, 667852967, 667852978, 667852988,
    667852999, 667853009, 667853020, 667853030, 667853041, 667853051, 667853062, 667853072,
    667853082, 667853093, 667853103, 667853113, 667853124, 667853134, 667853145, 667853155,
    667853166, 667853176, 667853186, 667853197, 667853207, 667853218, 667853228, 667853239,
    667853249, 667853259, 667853270, 667853280, 667853291, 667853301, 667853311, 667853322,
    667853332, 667853343, 667853353, 712659283, 725101076, 748403407, 208436385, 499104232,
    499104240, 499104248, 499104256, 2267162, 15823608, 253317719, 253317728, 253317737,
    440385018, 440385027, 52352969, 237900821, 448880171, 165940954, 499104197, 499104205,
    499104213, 499104221, 498541194, 399151316, 208436395.
```

8.1.4 Human genome and transcriptome

The human GRC p13 reference genome was obtained from:

```
ftp://ftp.ncbi.nlm.nih.gov/genomes/Homo_sapiens/current/GCF_000001405.39_GRCh38.p13/
    GCF_000001405.39_GRCh38.p13_genomic.fna.gz
```

while the human transcriptome from:

```
ftp://ftp.ncbi.nlm.nih.gov/refseq/H_sapiens/annotation/GRCh38_latest/refseq_identifiers/
    GRCh38_latest_rna.fna.gz
```

Both files were decompressed and merged into a single file with the name HS.fna.

8.2 Running commands

8.2.1 Installation

To install EAGLE, the following commands must be executed

```
git clone https://github.com/cobilab/eagle.git
cd eagle/src/
cmake .
make
```

To install Bowtie2, Samtools, and IGV, the following commands must be executed

```
conda install -c bioconda bowtie2 samtools igv -y
```

8.2.2 EAGLE and alignment results

To get the SARS-CoV-2 RAWs relative to the human with regular and inverted regions:

```
./EAGLE -v -t -p -min 12 -max 17 HS.fna SARS2.fa
```

EAGLE will create automatic SHELL scripts to plot the results. Therefore, first, there is the need to give execution access to the scripts by

```
chmod +x *.sh
```

Then, the scripts must be executed by

```
./kplot.sh
./AVGplot.sh
./CGplot.sh
./CGprofileplot.sh
```

Every time that EAGLE runs it will generate automatically the plot scripts, hence, for each run the scripts need also to be executed. For running SARS-CoV-2 RAWs without inversions, the following command must be applied

```
./EAGLE -v -t -i -p -min 12 -max 17 HS.fna SARS2.fa
```

For running EBOV with inversions, the following command must be applied

```
./EAGLE -v -t -p -min 12 -max 17 HS.fna EBOV.fa
```

For running EBOV without inversions, the following command must be applied

```
./EAGLE -v -i -t -p -min 12 -max 17 HS.fna EBOV.fa
```

The NCBI reference of SARS-CoV-2 (NC_045512) was extracted and all the genomes have been aligned to the reference using Bowtie2 [4]. Then, the SNPs of Supplementary Figure S9 were computed using the following Shell script

```
#!/bin/bash
THREADS=" 8 ";
rm -f index-file.i result.sam result.bam ;
bowtie2-build REFERENCE.fa IDX-REF
bowtie2 -a --threads $THREADS -x IDX-REF -f SARS2_noRef.fa -S result.sam
samtools sort --threads $THREADS result.sam > result.bam
samtools index -@ $THREADS result.bam result.bam.bai
```

where the THREADS variable can be edited for improving performance.
The detection of the RAWs, for the multiple human coronaviruses, can be computed with

```
./EAGLE -v -t -p -min 12 -max 17 HS.fna CORONAS.fa
```

The information about CG/AT content of the reference coronavirus can be computed using:

```
./EAGLE -v -t -p -min 12 -max 17 HS.fna SARS.fa # Running SARS only
mv CG-data.eg SARS-CG-data.eg
./EAGLE -v -t -p -min 12 -max 17 HS.fna MERS.fa # Running MERS only
mv CG-data.eg MERS-CG-data.eg
./EAGLE -v -t -p -min 12 -max 17 HS.fna HKU.fa  # Running HKU only
mv CG-data.eg HKU-CG-data.eg
./EAGLE -v -t -p -min 12 -max 17 HS.fna NL63.fa # Running NL63 only
mv CG-data.eg NL63-CG-data.eg
./EAGLE -v -t -p -min 12 -max 17 HS.fna OC43.fa # Running OC43 only
mv CG-data.eg OC43-CG-data.eg
./EAGLE -v -t -p -min 12 -max 17 HS.fna 229E.fa # Running 229E only
mv CG-data.eg 229E-CG-data.eg
```

Then, the CGplot.sh needs to be changed (only the last line) in order to join all the references information and, only after, the script needs to run. The following script contains all the changes and enables to create the image:

```
#!/bin/sh
echo 'reset
set terminal pdfcairo enhanced font "Monospace,12"
set output "CGplot.pdf"
set style line 11 lc rgb "#000000" lt 1 lw 2
set border 3 back ls 11
set tics nomirror
set size ratio 1
set style line 12 lc rgb "#000000" lt 0 lw 2
set grid back ls 12
set key outside vertical center right
set xlabel "k-mer"
set ylabel "Distribution (%)"
set xrange [12:17]
set yrange [30:70]
plot "SARS-CG-data.eg" u 1:2 t "SARS AT" w lp ls 1,  "SARS-CG-data.eg" u 1:3 t "SARS CG" w lp
     ls 2, "MERS-CG-data.eg" u 1:2 t "MERS AT" w lp ls 3,  "MERS-CG-data.eg" u 1:3 t "MERS CG"
     w lp ls 4, "HKU-CG-data.eg" u 1:2 t "HKU AT" w lp ls 5,  "HKU-CG-data.eg" u 1:3 t "HKU CG
    " w lp ls 6, "NL63-CG-data.eg" u 1:2 t "NL63 AT" w lp ls 7,  "NL63-CG-data.eg" u 1:3 t "
    NL63 CG" w lp ls 8, "OC43-CG-data.eg" u 1:2 t "OC43 AT" w lp ls 9, "OC43-CG-data.eg" u 1:3
     t "OC43 CG" w lp ls 10, "229E-CG-data.eg" u 1:2 t "229E AT" w lp ls 13,  "229E-CG-data.eg
    " u 1:3 t "229E CG" w lp ls 14' | gnuplot -persist
```

The following instructions need to be computed in order to replicate Fig. S2-a:

```
#!/bin/bash
./EAGLE -v -t -p -min 12 -max 17 HS.fna EBOV.fa
mv AVG-data.eg regular-AVG-data.eg
./EAGLE -v -t -i -p -min 12 -max 17 HS.fna EBOV.fa
mv AVG-data.eg inverted-AVG-data.eg
#
echo 'reset
set terminal pdfcairo enhanced font "Verdana,12"
set output "EBOV-AVGplot.pdf"
set style line 11 lc rgb "#000000" lt 1 lw 2
set border 3 back ls 11
set logscale y
set tics nomirror
set size ratio 1
set style line 12 lc rgb "#808080" lt 0 lw 2
set grid back ls 12
set style line 1 lc rgb "#8b1a0e" pt 1 ps 1 lt 1 lw 3
set style line 2 lc rgb "#5e9c36" pt 6 ps 1 lt 1 lw 3
unset key
set xlabel "k-mer"
set ylabel "Number of mRAWs"
set xrange [12:17]
```

```
set yrange [:]
plot "regular-AVG-data.eg" u 1:2 w lp ls 1, "inverted-AVG-data.eg" u 1:2 w lp ls 2' | gnuplot
     -persist
```

while to replicate Fig. S2-b, the following computations must be provided:

```
#!/bin/bash
./EAGLE -v -t -p -min 12 -max 17 HS.fna SARS2.fa
mv AVG-data.eg regular-AVG-data.eg
./EAGLE -v -t -i -p -min 12 -max 17 HS.fna SARS2.fa
mv AVG-data.eg inverted-AVG-data.eg
#
echo 'reset
set terminal pdfcairo enhanced font "Verdana,12"
set output "SARS2-AVGplot.pdf"
set style line 11 lc rgb "#000000" lt 1 lw 2
set border 3 back ls 11
set logscale y
set tics nomirror
set size ratio 1
set style line 12 lc rgb "#808080" lt 0 lw 2
set grid back ls 12
set style line 1 lc rgb "#8b1a0e" pt 1 ps 1 lt 1 lw 3
set style line 2 lc rgb "#5e9c36" pt 6 ps 1 lt 1 lw 3
unset key
set xlabel "k-mer"
set ylabel "Number of mRAWs"
set xrange [12:17]
set yrange [:]
plot "regular-AVG-data.eg" u 1:2 w lp ls 1, "inverted-AVG-data.eg" u 1:2 w lp ls 2' | gnuplot
     -persist
```

References

[1] Andrew Waterhouse, Martino Bertoni, Stefan Bienert, Gabriel Studer, Gerardo Tauriello, Rafal Gumienny, Florian T Heer, Tjaart A P de Beer, Christine Rempfer, Lorenza Bordoli, et al. SWISS-MODEL: homology modelling of protein structures and complexes. *Nucleic acids research*, 46(W1):W296–W303, 2018.

[2] Panu Artimo, Manohar Jonnalagedda, Konstantin Arnold, Delphine Baratin, Gabor Csardi, Edouard De Castro, Severine Duvaud, Volker Flegel, Arnaud Fortier, Elisabeth Gasteiger, et al. ExPASy: SIB bioinformatics resource portal. *Nucleic acids research*, 40(W1):W597–W603, 2012.

[3] James T Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S Lander, Gad Getz, and Jill P Mesirov. Integrative genomics viewer. *Nature biotechnology*, 29(1):24–26, 2011.

[4] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature methods*, 9(4):357, 2012.

[5] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[6] Diogo Pratas, Morteza Hosseini, and Armando J Pinho. GeCo2: an optimized tool for lossless compression and analysis of DNA sequences. In *International Conference on Practical Applications of Computational Biology & Bioinformatics*, pages 137–145. Springer, 2019.

[7] João R. Almeida, Armando J. Pinho, José L. Oliveira, Olga Fajarda, and Diogo Pratas. GTO: A toolkit to unify pipelines in genomic and proteomic research. *SoftwareX*, 12:100535, 2020.

[8] Morteza Hosseini, Diogo Pratas, Burkhard Morgenstern, and Armando J Pinho. Smash++: an alignment-free and memory-efficient tool to find genomic rearrangements. *GigaScience*, 9(5):giaa048, 2020.

[9] Diogo Pratas, Morteza Hosseini, and Armando J Pinho. Substitutional tolerant Markov models for relative compression of DNA sequences. In *International Conference on Practical Applications of Computational Biology & Bioinformatics*, pages 265–272. Springer, 2017.

[10] Xiaolu Tang, Changcheng Wu, Xiang Li, Yuhe Song, Xinmin Yao, Xinkai Wu, Yuange Duan, Hong Zhang, Yirong Wang, Zhaohui Qian, et al. On the origin and continuing evolution of SARS-CoV-2. *National Science Review*, 2020.

[11] Diogo Pratas. *Compression and analysis of genomic data*. PhD thesis, Universidade de Aveiro (Portugal), 2016.