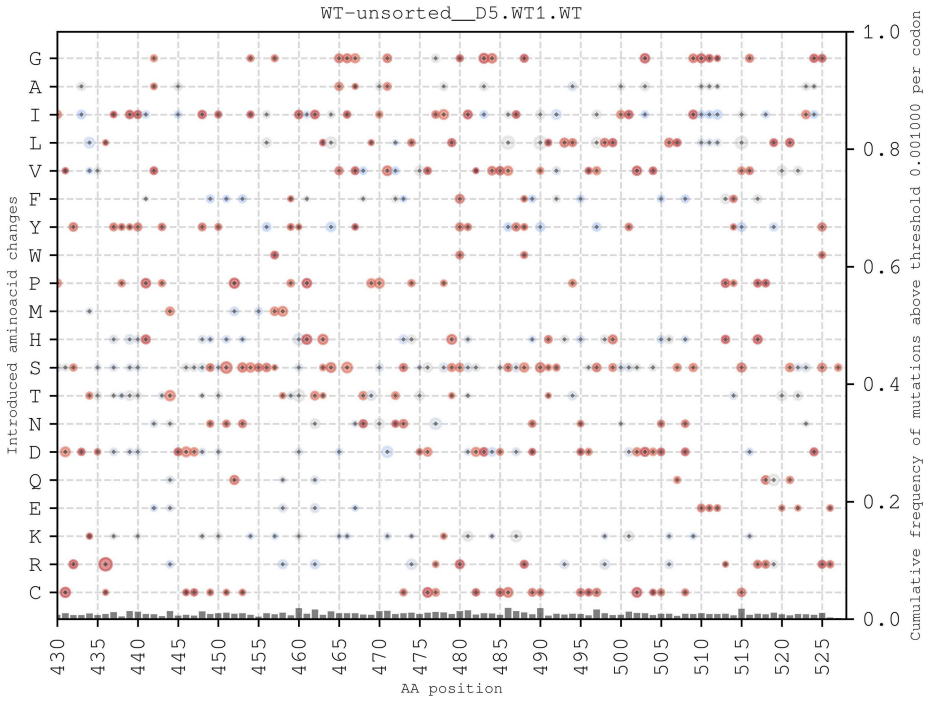
mutation_scatter_plot: Tackling codon usage analysis from a different angle

ENBIK 2025



FIRST FACULTY OF MEDICINE
Charles University

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Martin Mokrejš

martin.mokrejs@lf1.cuni.cz

Introduction

Codon usage used to unleash evolution of protein-coding regions
ratio between synonymous and non-synonymous changes

Lab assays: PCR-based mutation were introduced into an amplicon of a portion of the viral S protein, the protein was expressed on *S. cerevisiae* host cells

Raw outputs:

- 100k – 200k NGS read pairs / sample (Illumina, 2x250 or 2x300 cycles)

- Multiple sequence alignment of the Covid19 S protein from a vast set of raw NGS reads

Aim: to determine which codons out of the theoretical 64 are used in every amino acid position of the encoded protein.

Introduction

What was needed:

1) manually polish multiple sequence alignment, or
2) use **`mafft`** to re-align the already existing alignment to improve it a bit
3) find or develop a software to introduce padding gaps
    1) into either sequence due to sequencing errors
    2) for rare events
        1) into the reference sequence for INSertion events in the sample NGS reads
        2) into the sample NGS reads for DELetion events
4) we found gofasta can derive multiFASTA files back from SAM/BAM

What was needed:
1)manually polish mult
2)use `mafft` to re-alig
3)find or develop a sof
    1)into either sequ
    2)for rare events
        1)into the re
        2)into the sa
4)we found gofasta ca
5)

gofasta: command-line utilities for genomic epidemiology research

Ben Jackson

Bioinformatics, Volume 38, Issue 16, August 2022, Pages 4033–4035, https://doi.org/10.1093/bioinformatics/btac424
Published: 05 July 2022 **Article history**

PDF Split View Cite Permissions Share

Abstract

Summary

gofasta comprises a set of command-line utilities for handling alignments of short assembled genomes in a genomic epidemiology context. It was developed for processing large numbers of closely related SARS-CoV-2 viral genomes and should be useful with other densely sampled pathogen genomic datasets. It provides functions to convert sam-format pairwise alignments between assembled genomes to fasta format; to annotate mutations in multiple sequence alignments, and to extract sets of sequences by genetic distance measures for use in outbreak investigations.

Availability and implementation

gofasta is an open-source project distributed under the MIT license. Binaries are available at https://github.com/virus-evolution/gofasta, from Bioconda, and through the Go programming language's package management system. Source code and further documentation, including walkthroughs for common use cases, are available on the GitHub repository.

ads

Introduction (continued)

What was needed:

1)manually polish multiple sequence alignment, or

2)use **`mafft`** to re-align the already existing alignment to improve it a bit

3)find or develop a software to introduce padding gaps

    1)into either sequence due to sequencing errors

    2)for rare events

        1)into the reference sequence for INSertion events in the sample NGS reads

        2)into the sample NGS reads for DELetion events

4)we found gofasta can derive multiFASTA files back from SAM/BAM

- gofasta is used by PANGOLIN pipeline https://cov-lineages.org/resources/pangolin/usage.html
- gofasta does not realize INSertion event in the sample but luckily in our assays we were only after nucleotide changes
- the protein must have been produced so no in-frame STOP codon event could have lead to protein production (hence those must have been sequencing errors)

1)develop software to count codon frequencies

2)develop software to render figures

3)develop interactive figures

    1)matplotlib

    2)python Bokeh library (HTML+javascript)

Our approaches (overview)

We developed a pipeline to process the raw sequencing data but rejected some tools as not appropritate

1)sequencing of the amplicon libraries requires the sample to be diluted in the sequencing lane with some genomic DNA samples to avoid ghost wells and barcodes overflashing each other

2)PCR duplicate removal is not possible because this an an amplicon-based project

3)optical de-duplication would have to be at the sequencing center, is not possible (meaningfully) from FASTQ files received on our side anymore

4)run **`blastn -task blastn -w 4 -reward 2 -max_hsps 1 -num_alignments 1`**
to confirm which amplicon insert in the read, discard junk and NGS library prep. chimeras

5)sample demultiplexing

6)consensus derivation based on UMI tags (I failed with umi-tools and picard tools)

7)aligning to a reference (bwa or blastn)

8)convert BAM into multiFASTA alignment using gofasta https://github.com/virus-evolution/gofasta

- see github Issue 55
- it discards INSertion present in sample relative to reference, e.g. 3 nt long INSertions
- it DOES report DELetion events releative to reference

9)analysis of the multiple sequence alignment and determination of changed codons and aa residues

Sample demultiplexing

We tried multiple demultiplexing tools with varying results on our

8nt barcodes on both 5'- and 3'-ends

barcodes were designed for Hamming distance 5 (that turned to be insufficient due to InDels)

theoretical checks on barcode resiliency was done by bardcode tester https://barcode.readthedocs.io/en/latest/usage.html

•ultraplex https://github.com/ulelab/ultraplex

– for our 8nt barcodes on both ends it took way to much RAM and later did not work somehow, see e.g. github Issue 51

•demultiplex https://github.com/jfjlaros/demultiplex

•– it can search for a barcode on 5'-end and then on the 3'-end

•– DO NOT run it in a singe sweep with increased maximum distance

•– instead write a wrapper script and run it in exact match mode, then relax the thresholds and re-run on the UNKNOWN reads, the loop again

•– it iterates in multiple rounds, in each you increase the maximum distance threshold and apply it to reads in the UNKNOWN group from a previous step

•– some reads were assigned to different (contradictory) samples based on fwd_bcode versus rev_bcode, e.g. left fwd_bcode was without errors but rev_bcode matches at distance > 2

•– it was possible to fine-tune the number of rounds needed to get most of the NGS reads assigned to a sample, see e.g. github Issue 32

•– demultiplex uses a fixed window to search for a barcode

•– hence is misses incomplete barcodes and their 5'- or 3'-end !!!

1) sample demultiplexing – increasing Edit distance threshold

			demultiplex with increasing Hamming (Levenshtein) distance -m NUM due to -d argument											from all 5 levels (not used for downstream read mapping and analysis)	failed primer match (samtools ampliconstats)	Mapped reads' of those picked from '-m 0' step only
library	fwd_barcode_n	arev_barcode_n	-d -m 0 -e 16	-d -m 1 -e 16	-d -m 2 -e 16	-d -m 3 -e 16	-d -m 4 -e 16	SUM								
WW-A1 affinity	JZ190_0_F	JZ191_R	370 426	5 612	13 135	1 419	11 015	401 607	385 230	1 170	740 732					
WW-A1 affinity	JZ192_1_F	JZ191_R	246 055	4 009	30 632	1 630	23 820	306 146	270 672	772	492 033					
WW-A1 affinity	JZ193_2_F	JZ191_R	250 656	3 812	23 142	1 550	29 938	309 098	271 853	741	501 218					
WW-A1 affinity	JZ194_3_F	JZ191_R	552 163	8 204	29 903	2 646	16 110	609 026	579 622	1 720	1 104 160					
WW-A1 affinity	JZ195_4_F	JZ191_R	260 552	4 550	19 405	1 793	7 960	294 260	277 248	753	520 974					
WW-A1 affinity	JZ196_5_F	JZ191_R	378 430	5 640	18 978	1 672	9 478	414 198	396 207	1 190	756 743					
WW-A1 affinity	JZ197_6_F	JZ191_R	485 941	6 899	44 713	2 438	14 529	554 520	519 626	1 444	971 712					
WW-A1 affinity	JZ198_7_F	JZ191_R	275 698	4 996	35 446	2 133	26 509	344 782	304 270	863	551 256					
WW-A1 affinity	JZ199_8_F	JZ191_R	280 890	4 096	24 551	1 367	21 606	332 510	301 899	952	561 656					
WW-A1 affinity	JZ200_9_F	JZ191_R	432 079	5 060	39 052	2 085	17 889	496 165	461 296	1 311	863 912					
WW-A1 affinity	JZ201_10_F	JZ191_R	281 955	4 498	24 230	1 589	23 315	335 587	303 576	776	563 795					
WW-A1 affinity	JZ202_11_F	JZ191_R	354 070	5 072	33 139	1 897	23 310	417 488	380 961	952	708 003					
WW-A1 affinity	JZ203_12_F	JZ191_R	436 914	10 746	53 231	4 469	37 847	543 207	483 097	1 240	873 688					
WW-A1 affinity	JZ204_13_F	JZ191_R	509 363	9 395	54 042	3 926	17 531	594 257	551 148	1 399	1 018 533					
WW-A1 affinity	JZ205_14_F	JZ191_R	195 500	3 880	47 116	2 813	21 099	270 408	228 425	736	390 944					
WW-A1 affinity	JZ206_15_F	JZ191_R	337 398	5 789	28 083	2 912	21 553	395 735	362 338	1 097	674 706					
WW-A1 affinity	JZ207_16_F	JZ191_R	493 484	7 446	41 681	2 551	14 036	559 198	526 004	1 694	986 795					
WW-A1 affinity	JZ208_17_F	JZ210_R	346 871	8 291	36 565	2 573	26 148	420 448	379 414	1 159	693 604					

- one can seemingly assign more reads by allowing Levenshtein distance == 5 but the assignments are contradictory
- there is no scoring code to respect barcode match with the least differences in demultiplex
- instead, loop e.g. 3 times and run with “-d -m NUM” with NUM in range (0 – 2)

5% of barcodes on each amplicon end (10% total) were incomplete due to suboptimal PCR extension time and polymerase mix

- 343nt long PCR amplicon was not amplificed correctly within 16 seconds of extension time
- proof-reading polymerase can cause recessive ends and also blunt-ends
- we ended up with amplicons inserted in both plus and minus orientation
- the PCR-primers should have caused reparation of incomplete products in each cycle but somehow did not
- probably will need to interleave PCR cycles with longer extension times and use Taq polymerase in addition to proof-reading enzymes

barcode-library-specfic region -primer

```
 24 GCCCC~~t~~CTGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 24 ~~gcaacg~~TGGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 27 ~~gca~~ACGTGGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 27 GCA~~a~~CGTGGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 29 ~~ccgtg~~GGTGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 29 ~~gcaae~~GTGGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 34 ~~ccgt~~GGGTGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 35 CCGTGG~~g~~TGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 42 ~~gccccet~~CTGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 44 ~~taagt~~AGAGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 45 ~~gcaa~~CGTGGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 51 ~~ccg~~TGGGTGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 57 ~~taag~~TAGAGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 63 ~~gc~~AACGTGGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 66 ~~gccc~~CTCTGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 70 ~~nnnnnn~~GAGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
 84 ~~nnnnnnn~~AGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
104 ~~nnnnnnn~~TGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
115 ~~cc~~GTGGGTGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
122 ~~ta~~AGTAGAGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
122 ~~gc~~CCCTCTGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
143 ~~g~~CAACGTGGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
145 GCCC~~c~~TCTGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
150 ~~gcc~~CCTCTGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
150 ~~taa~~GTAGAGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
176 ~~g~~CCCCTCTGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
195 ~~c~~CGTGGGTGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
228 ~~nnnnnnnn~~GTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
256 ~~t~~AAGTAGAGTACAAGTGCTAGCCATATGGGTTGCCCTTTTG
```

Results

We developed two rather simple programs

1) calculate_codon_frequencies.py

- counts the codon
- counts amimo acid frequencies
- outputs TSV files with their frequencies

2) mutation_scatter_plot.py

- displays the frequencies as scatter plots with interactive bubbles upon mouse hover().
- the changes can be color-coded according to e.g. physicochemical properties of the amino acid residues (PAM matrices) or **their evolutionary conservation (BLOSUM matrices)** or any other color-palette. However, such efforts are a bit naïve as the weights for each amino acid are not within the same minimum-maximum range and thus are not directly comparable.

The software is available at https://github.com/host-patho-evo/mutation_scatter_plot .

Rendering evolutionary conservation of amino acid changes using raw BLOSUM62 scores

```
# Entries for the BLOSUM62 matrix at a scale of ln(2)/2.0.
   A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V  B  J  Z  X  *
A  4 -1 -2 -2  0 -1 -1  0 -2 -1 -1 -1 -1 -2 -1  1  0 -3 -2  0 -2 -1 -1 -1 -4
R -1  5  0 -2 -3  1  0 -2  0 -3 -2  2 -1 -3 -2 -1 -1 -3 -2 -3 -1 -2  0 -1 -4
N -2  0  6  1 -3  0  0  0  1 -3 -3  0 -2 -3 -2  1  0 -4 -2 -3  4 -3  0 -1 -4
D -2 -2  1  6 -3  0  2 -1 -1 -3 -4 -1 -3 -3 -1  0 -1 -4 -3 -3  4 -3  1 -1 -4
C  0 -3 -3 -3  9 -3 -4 -3 -3 -1 -1 -3 -1 -2 -3 -1 -1 -2 -2 -1 -3 -1 -3 -1 -4
Q -1  1  0  0 -3  5  2 -2  0 -3 -2  1  0 -3 -1  0 -1 -2 -1 -2  0 -2  4 -1 -4
E -1  0  0  2 -4  2  5 -2  0 -3 -3  1 -2 -3 -1  0 -1 -3 -2 -2  1 -3  4 -1 -4
G  0 -2  0 -1 -3 -2 -2  6 -2 -4 -4 -2 -3 -3 -2  0 -2 -2 -3 -3 -1 -4 -2 -1 -4
H -2  0  1 -1 -3  0  0 -2  8 -3 -3 -1 -2 -1 -2 -1 -2 -2  2 -3  0 -3  0 -1 -4
I -1 -3 -3 -3 -1 -3 -3 -4 -3  4  2 -3  1  0 -3 -2 -1 -3 -1  3 -3  3 -3 -1 -4
L -1 -2 -3 -4 -1 -2 -3 -4 -3  2  4 -2  2  0 -3 -2 -1 -2 -1  1 -4  3 -3 -1 -4
K -1  2  0 -1 -3  1  1 -2 -1 -3 -2  5 -1 -3 -1  0 -1 -3 -2 -2  0 -3  1 -1 -4
M -1 -1 -2 -3 -1  0 -2 -3 -2  1  2 -1  5  0 -2 -1 -1 -1 -1  1 -3  2 -1 -1 -4
F -2 -3 -3 -3 -2 -3 -3 -3 -1  0  0 -3  0  6 -4 -2 -2  1  3 -1 -3  0 -3 -1 -4
P -1 -2 -2 -1 -3 -1 -1 -2 -2 -3 -3 -1 -2 -4  7 -1 -1 -4 -3 -2 -2 -3 -1 -1 -4
S  1 -1  1  0 -1  0  0  0 -1 -2 -2  0 -1 -2 -1  4  1 -3 -2 -2  0 -2  0 -1 -4
T  0 -1  0 -1 -1 -1 -1 -2 -2 -1 -1 -1 -1 -2 -1  1  5 -2 -2  0 -1 -1 -1 -1 -4
W -3 -3 -4 -4 -2 -2 -3 -2 -2 -3 -2 -3 -1  1 -4 -3 -2 11  2 -3 -4 -2 -2 -1 -4
Y -2 -2 -2 -3 -2 -1 -2 -3  2 -1 -1 -2 -1  3 -3 -2 -2  2  7 -1 -3 -1 -2 -1 -4
V  0 -3 -3 -3 -1 -2 -2 -3 -3  3  1 -2  1 -1 -2 -2  0 -3 -1  4 -3  2 -2 -1 -4
B -2 -1  4  4 -3  0  1 -1  0 -3 -4  0 -3 -3 -2  0 -1 -4 -3 -3  4 -3  0 -1 -4
J -1 -2 -3 -3 -1 -2 -3 -4 -3  3  3 -3  2  0 -3 -2 -1 -2 -1  2 -3  3 -3 -1 -4
Z -1  0  0  1 -3  4  4 -2  0 -3 -3  1 -1 -3 -1  0 -1 -2 -2 -2  0 -3  4 -1 -4
X -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -4
* -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4  1
```

https://www.ncbi.nlm.nih.gov/IEB/ToolBox/C_DOC/lxr/source/data/BLOSUM62

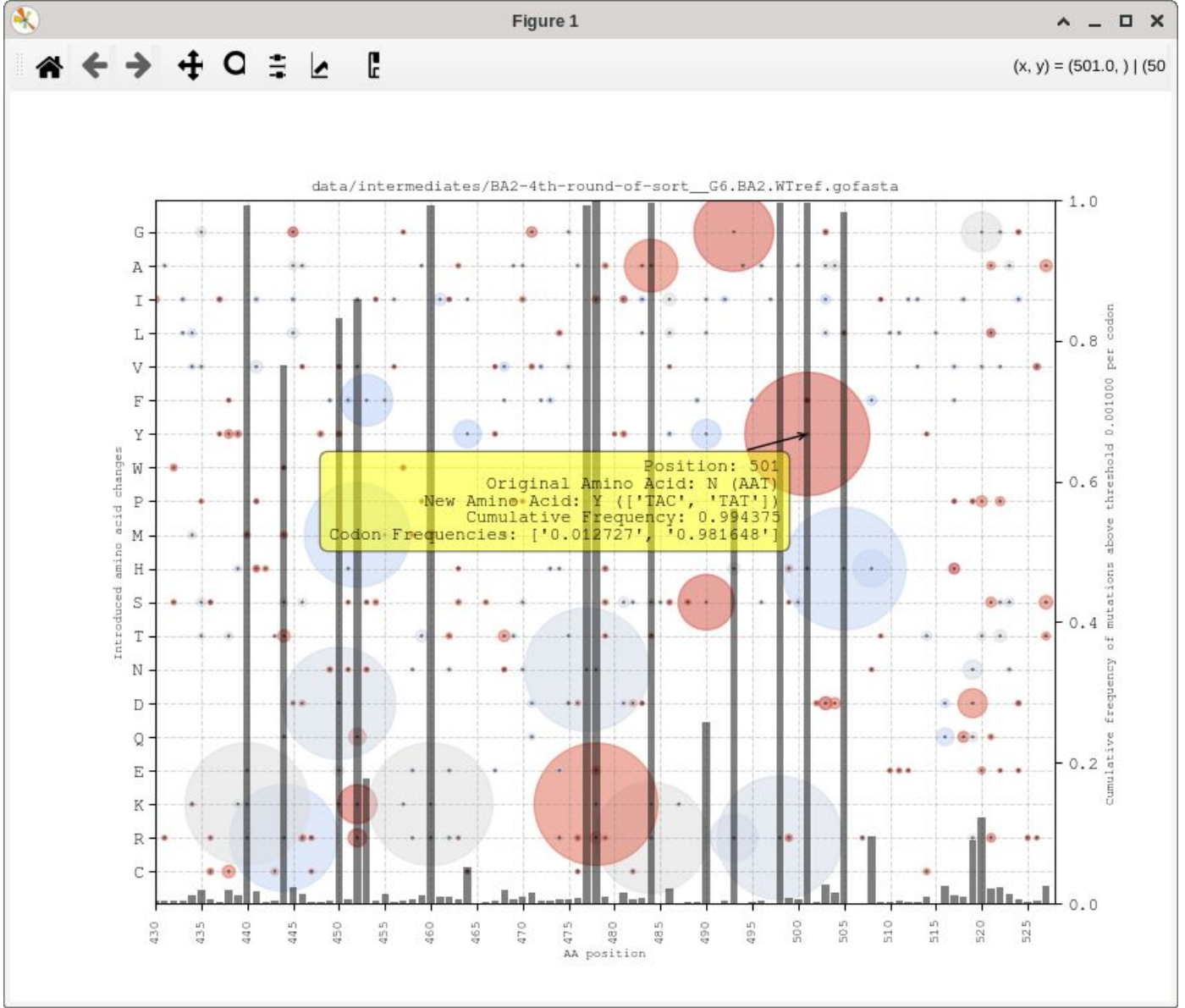
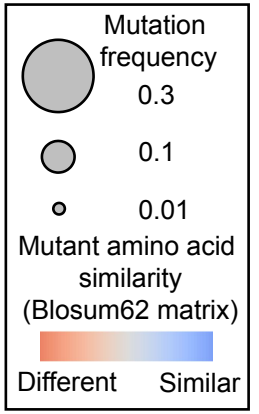
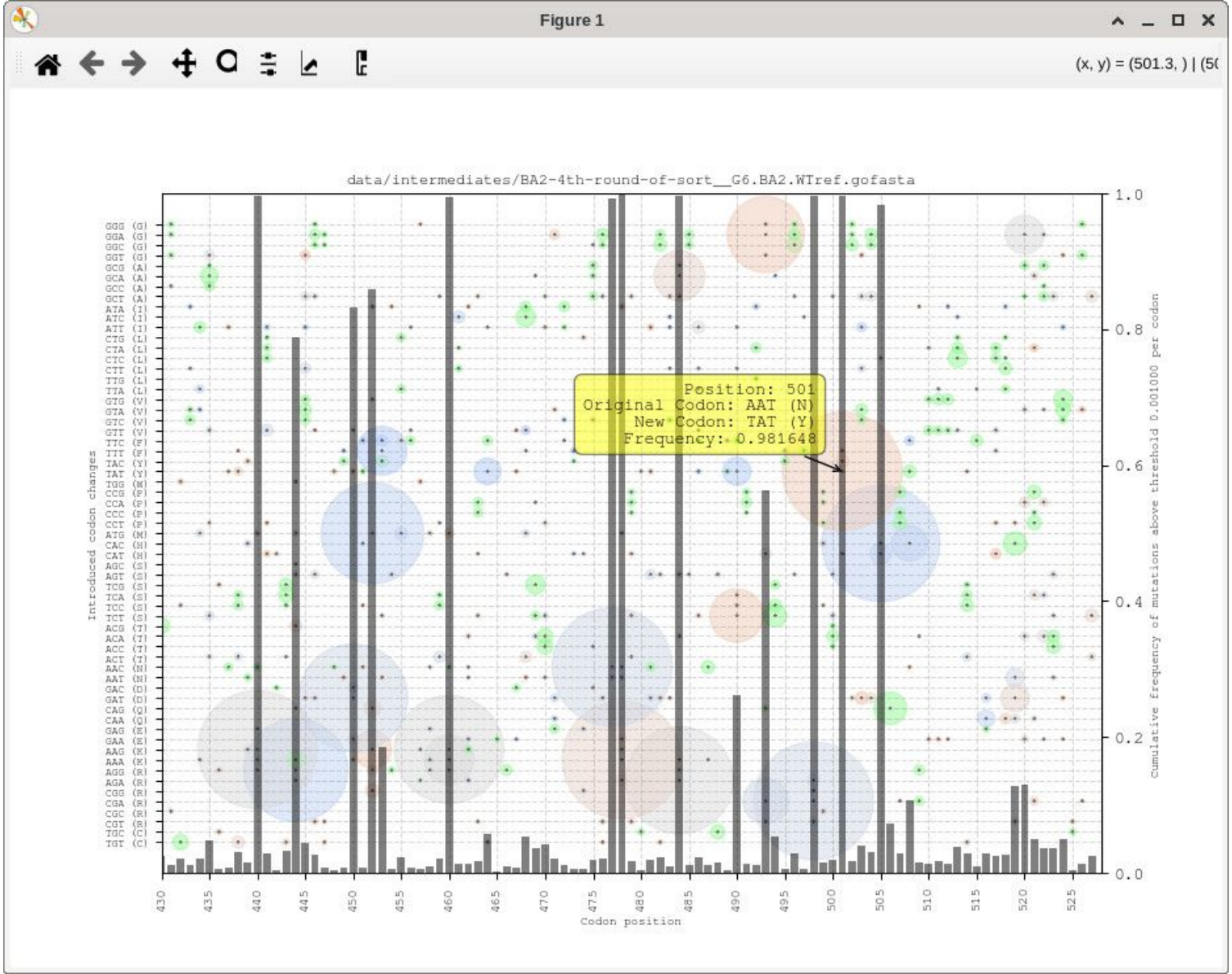


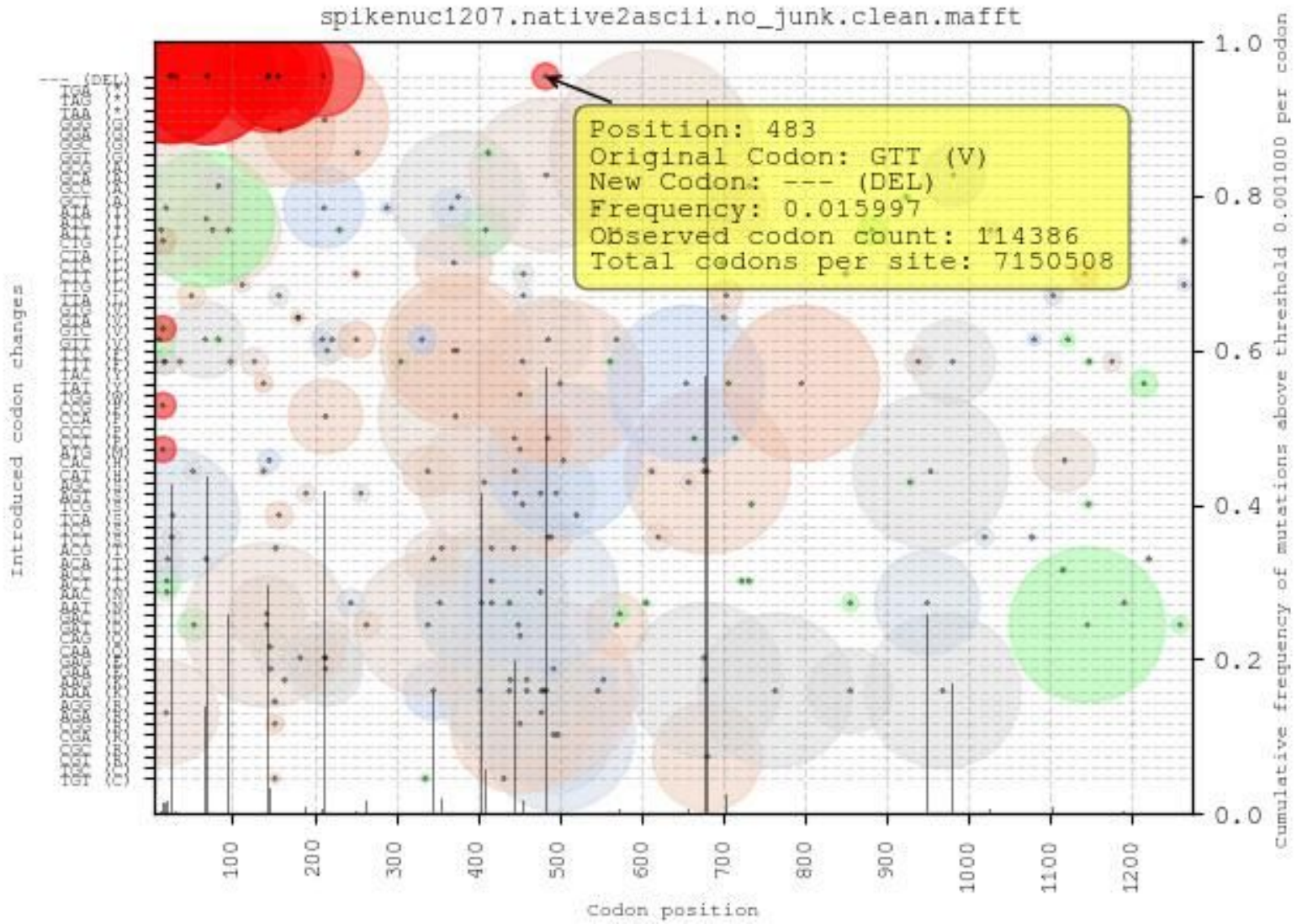
Figure 1
(x, y) = (501.0,) | (50
data/intermediates/BA2-4th-round-of-sort__G6.BA2.WTref.gofasta
Introduced amino acid changes
G
A
I
L
V
F
Y
W
P
M
H
S
T
N
D
Q
E
K
R
C
Position: 501
Original Amino Acid: N (AAT)
New Amino Acid: Y (['TAC', 'TAT'])
Cumulative Frequency: 0.994375
Codon Frequencies: ['0.012727', '0.981648']
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
AA position
1.0
0.8
0.6
0.4
0.2
0.0
Cumulative frequency of mutations above threshold 0.001000 per codon


Mutation frequency
0.3
0.1
0.01
Mutant amino acid similarity (Blosum62 matrix)
Different
Similar

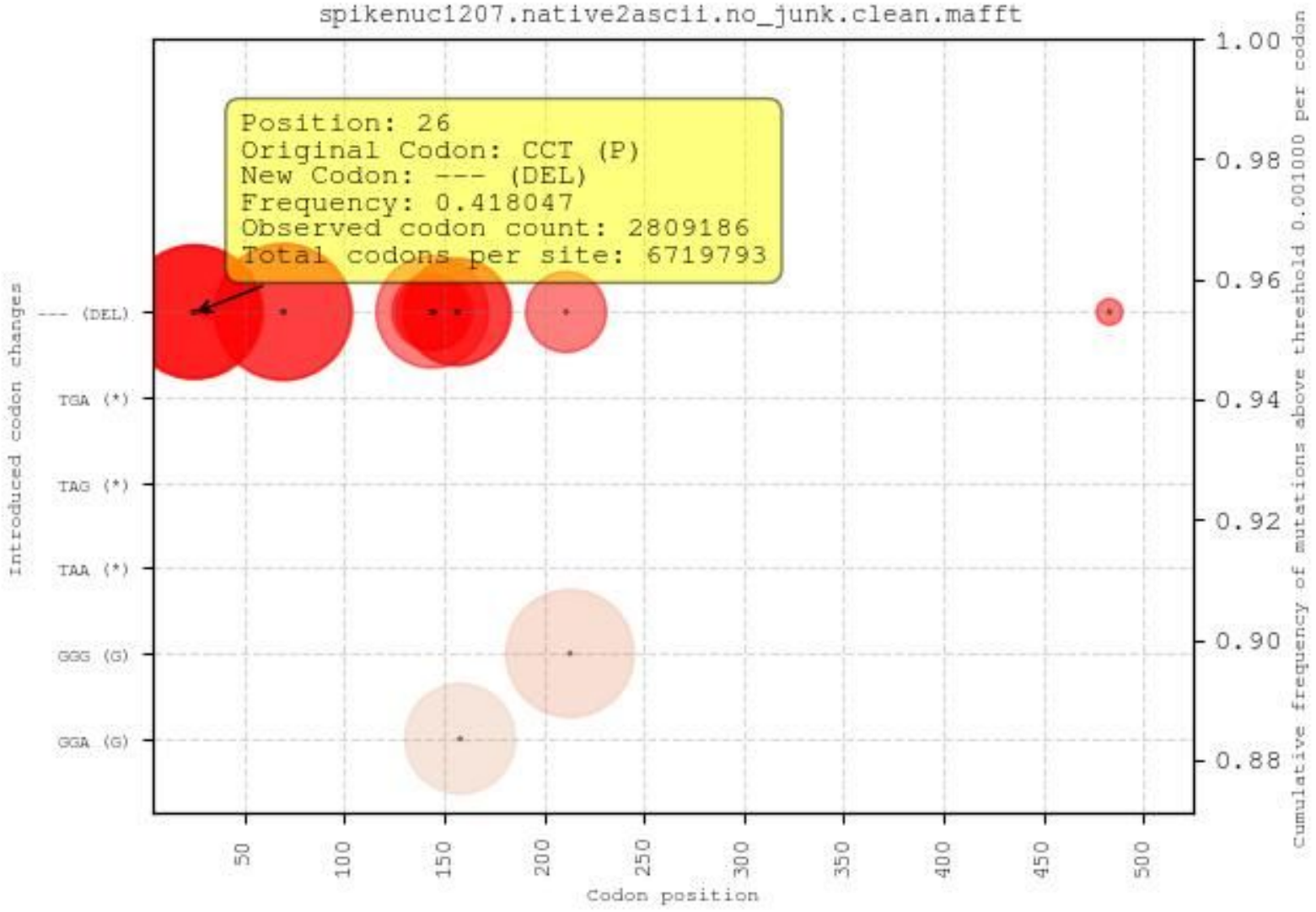


green – synonymous changes

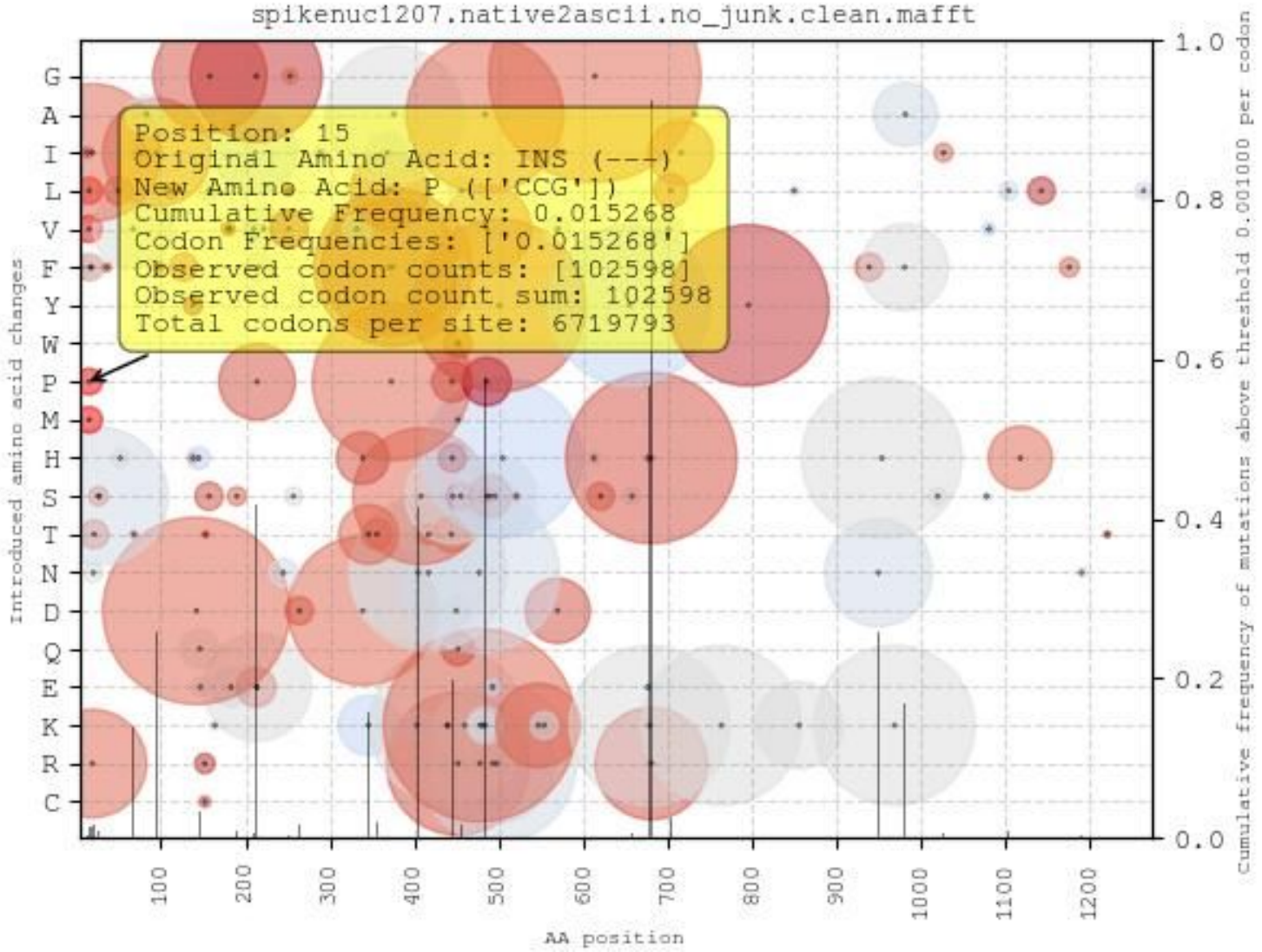
GISAID data with BA.2.86 DELetion at position 483



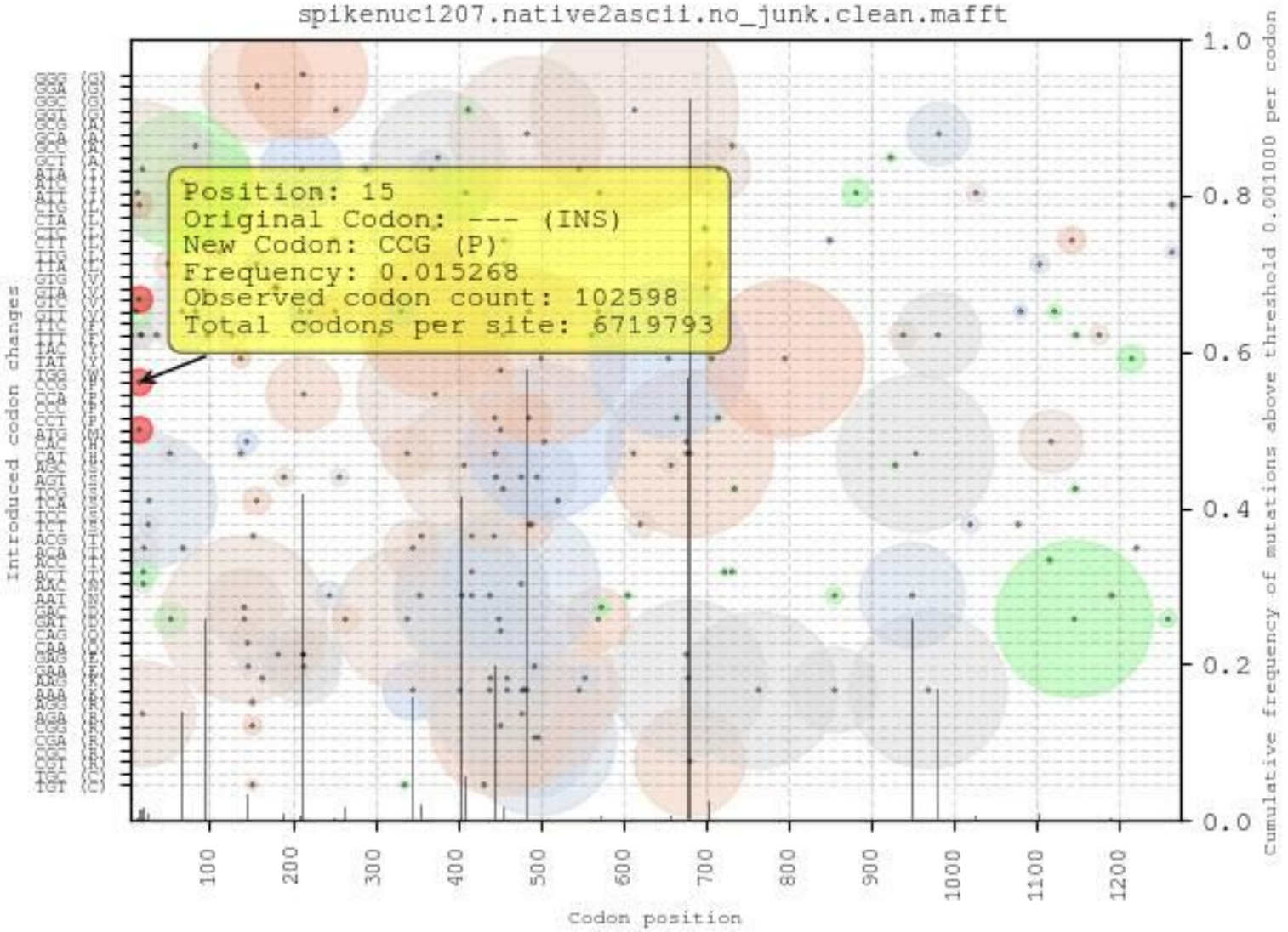
GISAID data with BA.2.86 DELetion at position 483



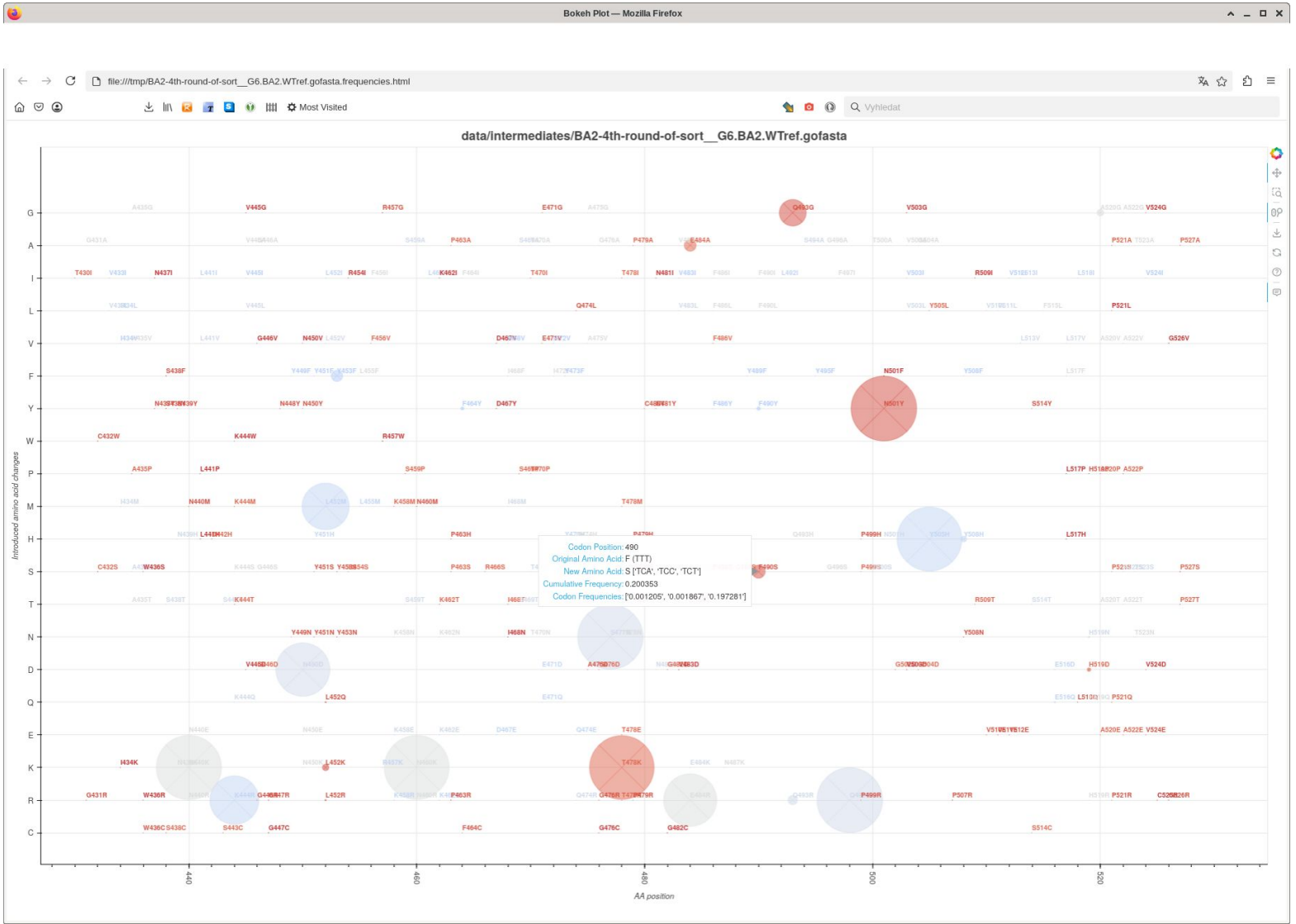
GISAID data with lots of drastic changes



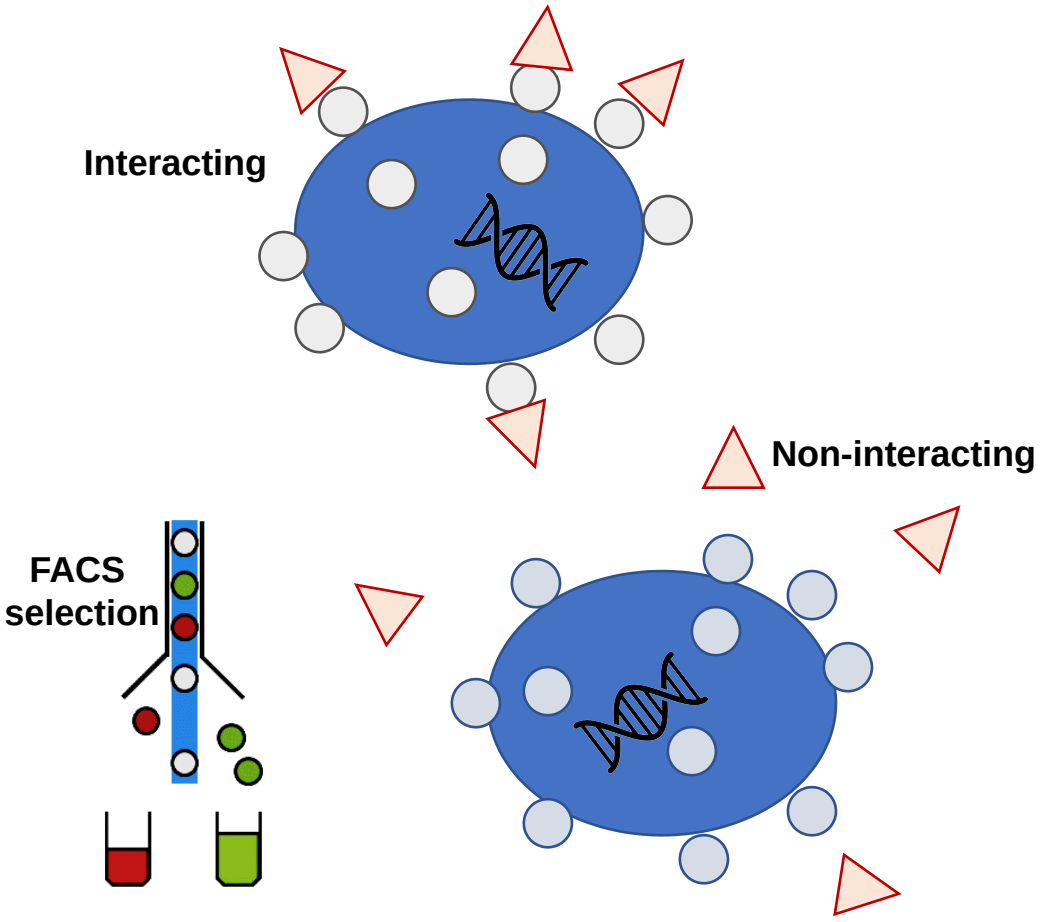
GISAID data with lots of drastic changes



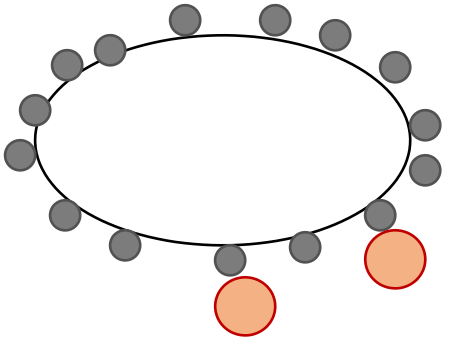
GISAID data with lots of drastic changes (red) rendered by Bokeh into HTML+javascript



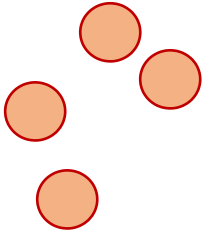
Introduction (biology - Yeast display in vitro evolution)



Yeast cell with RBD on its surface

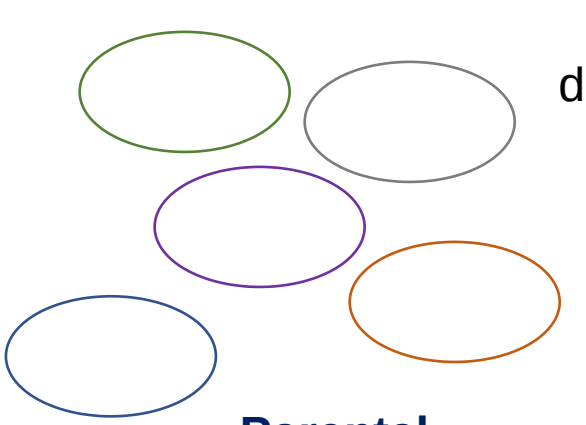


Labeled soluble ACE2 receptor

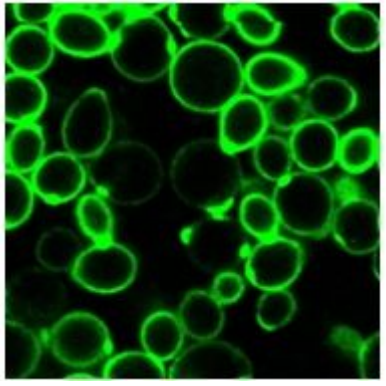


Zahradnik, J. et al. (**2021**) A protein engineered, enhanced yeast display platform for rapid evolution of challenging targets. ***ACS Synthet. Biol.*** 9 Nov. 2021, 10.1021/acssynbio.1c00395

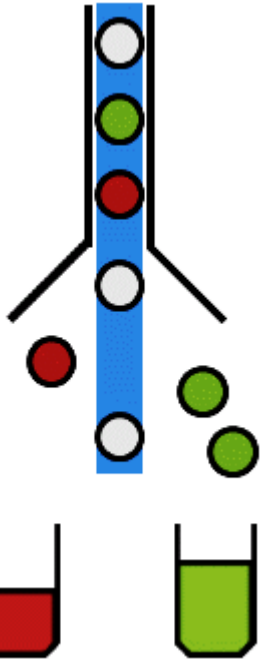
Introduction (biology – selection process)



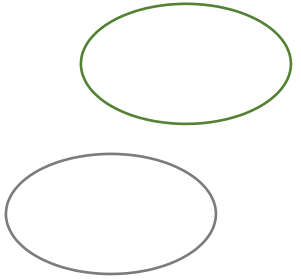
Parental population



Yeast library displaying RBD variants on surface



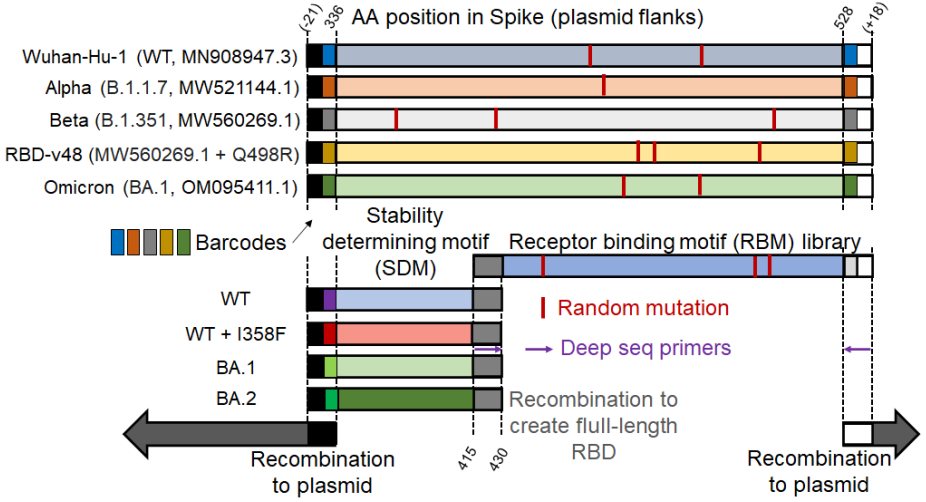
FACS Selection - ACE2 binding at given concentration



Selected yeast cells with RBD proteins of desired properties

Selected population

Results (biology)

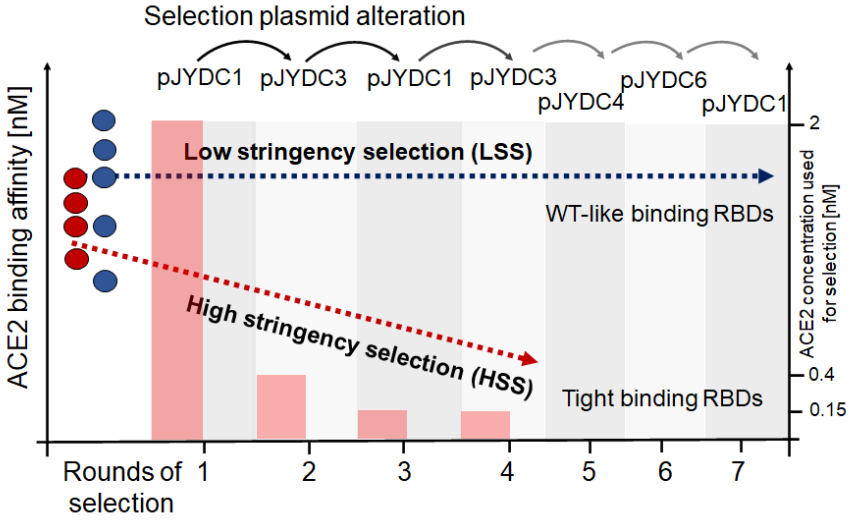


We used error-prone PCR conditions to introduce mutations into the RBD genes – similar to viral evolution

Nine different starting points

Two selection strategies

Over 100 selection steps



Low stringency selection (LSS)

Broader permissiveness
High concentration of ACE2 used

Stringent selection (HSS)

High affinity and stability
Low pM concentration of ACE2

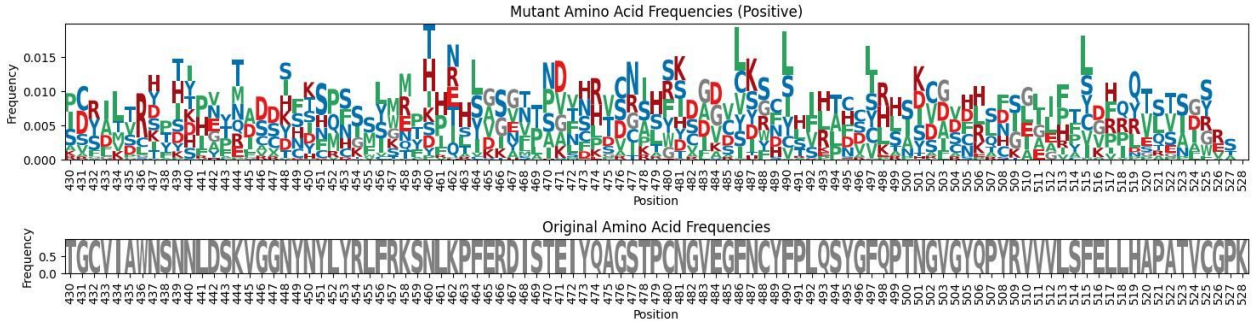
Results (BA1 evolution space)

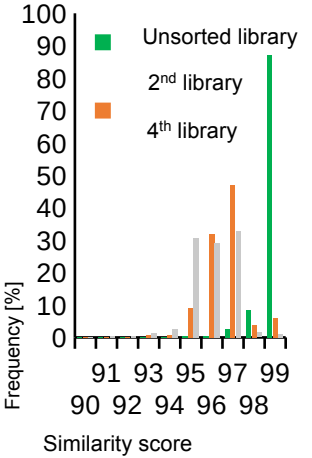
under high-stringency (HSS) conditions (aka low concentration of ACE2 available) we observed much more codon changes and hence amino acid changes

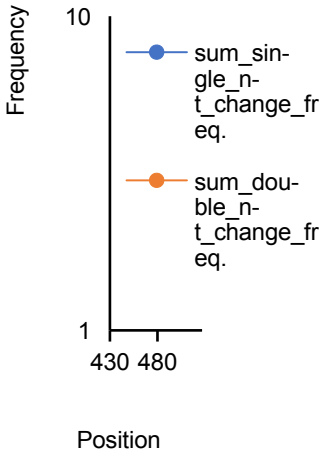
some changes are co-variant

to achieve “same” amount of changes much more rounds under low-stringency (LSS) would need to be done (when plenty of ACE2 is available)

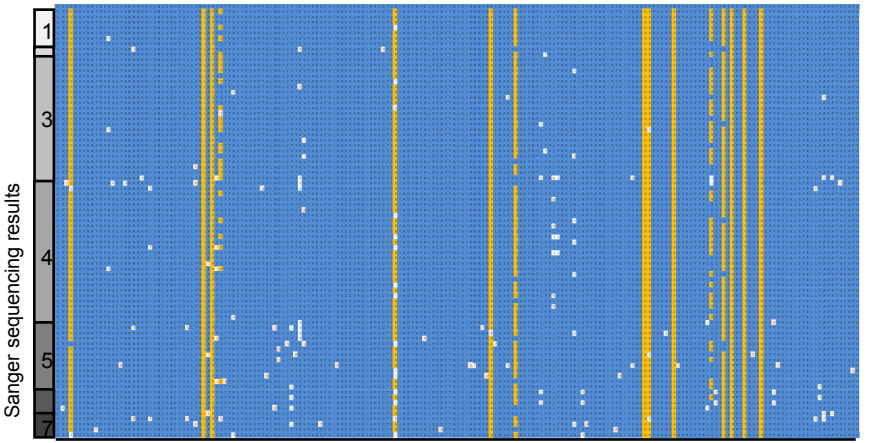
Analysis of error-prone libraries – before sorting and … (see bottom right)





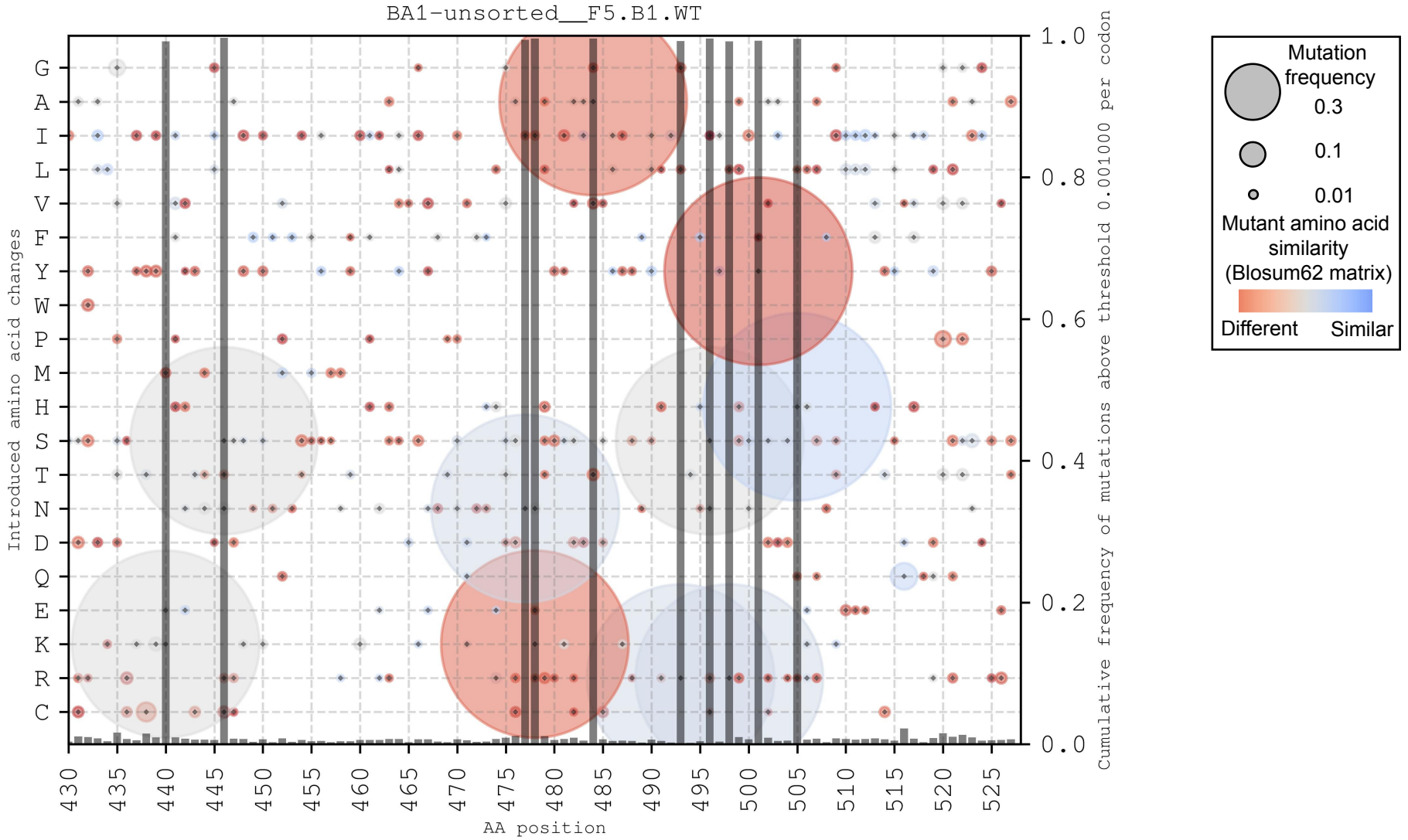


after sorting

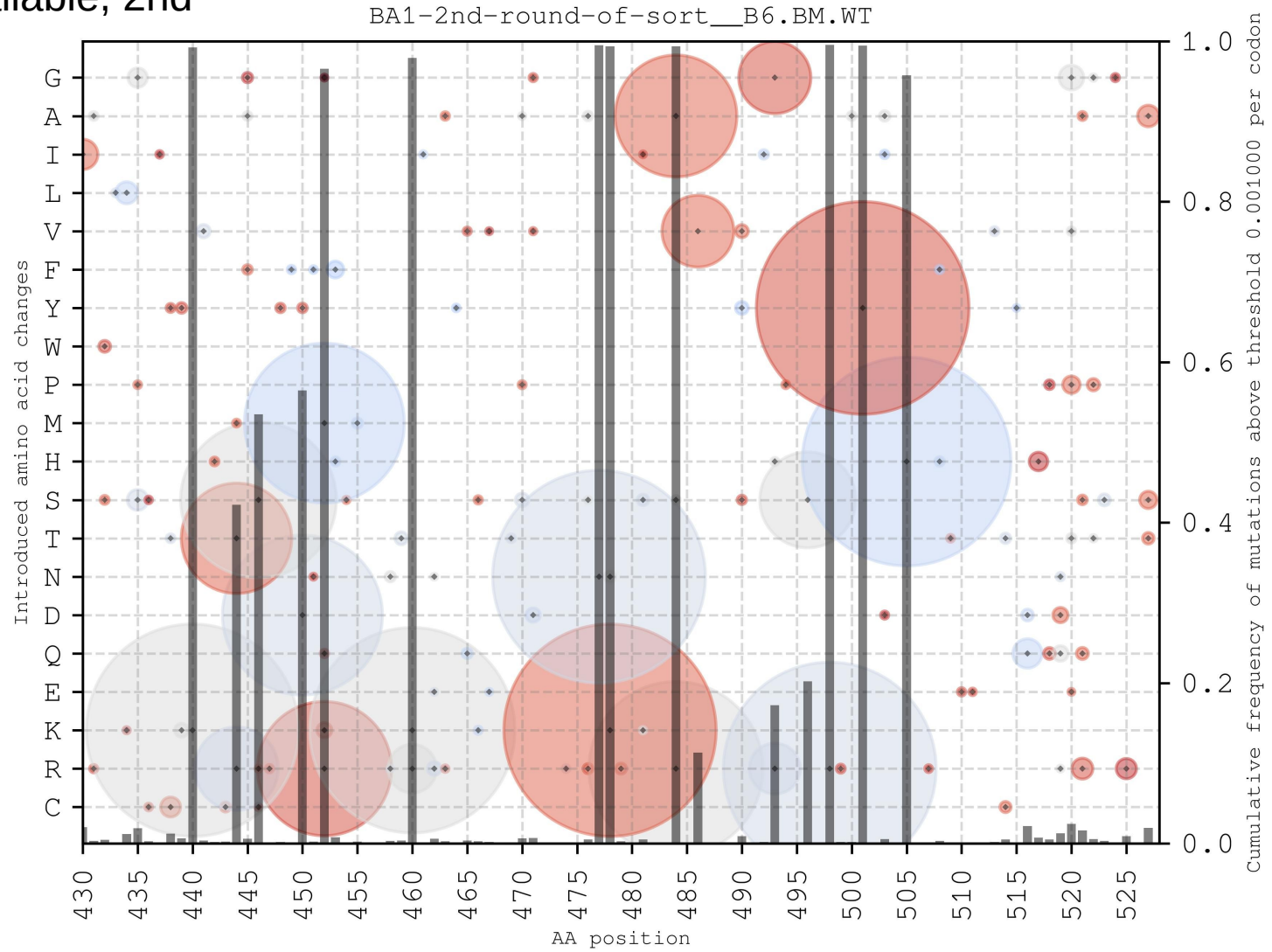


Aviv Shoshany

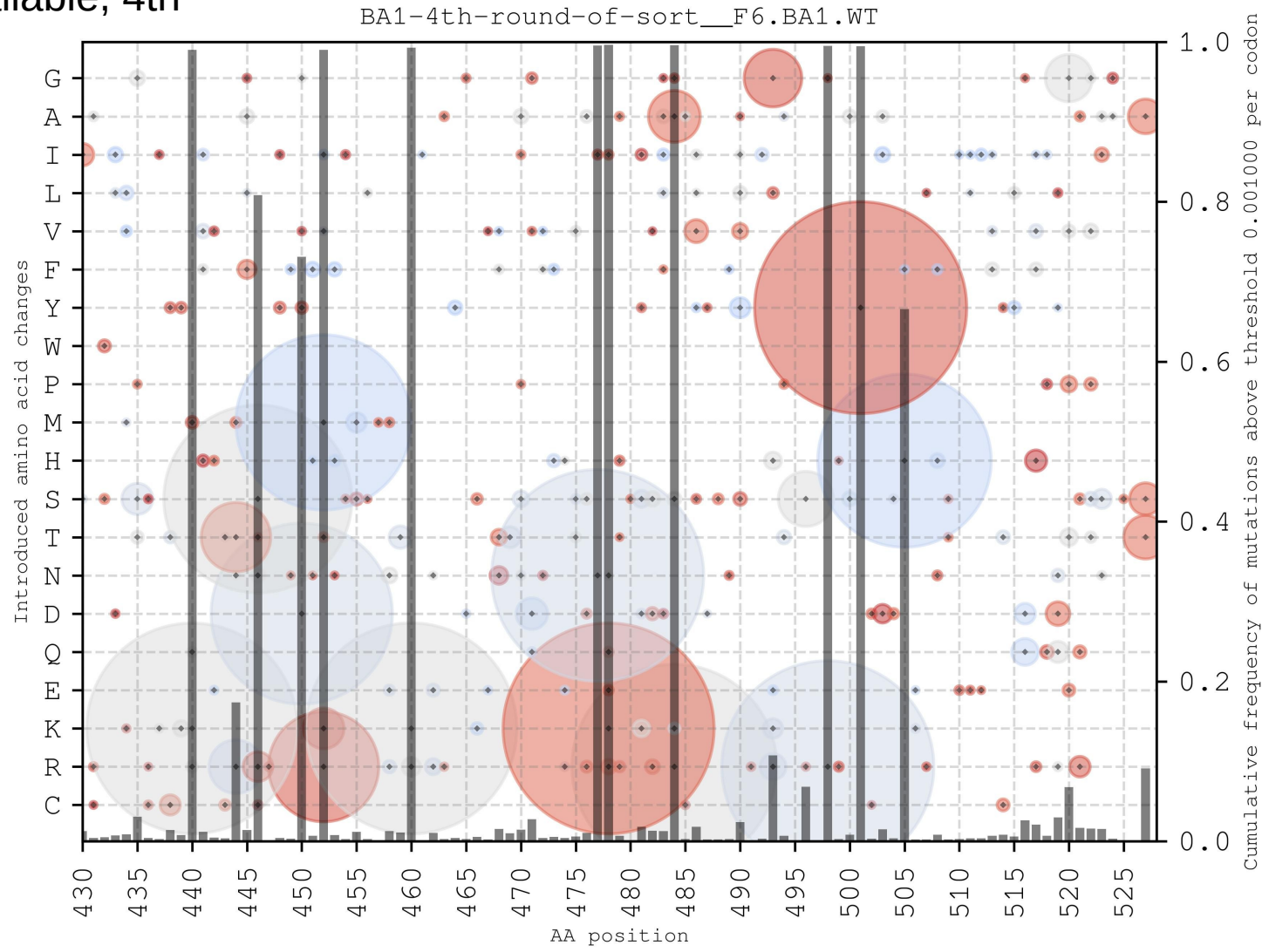
Starting parental BA.1 library (WT as reference)



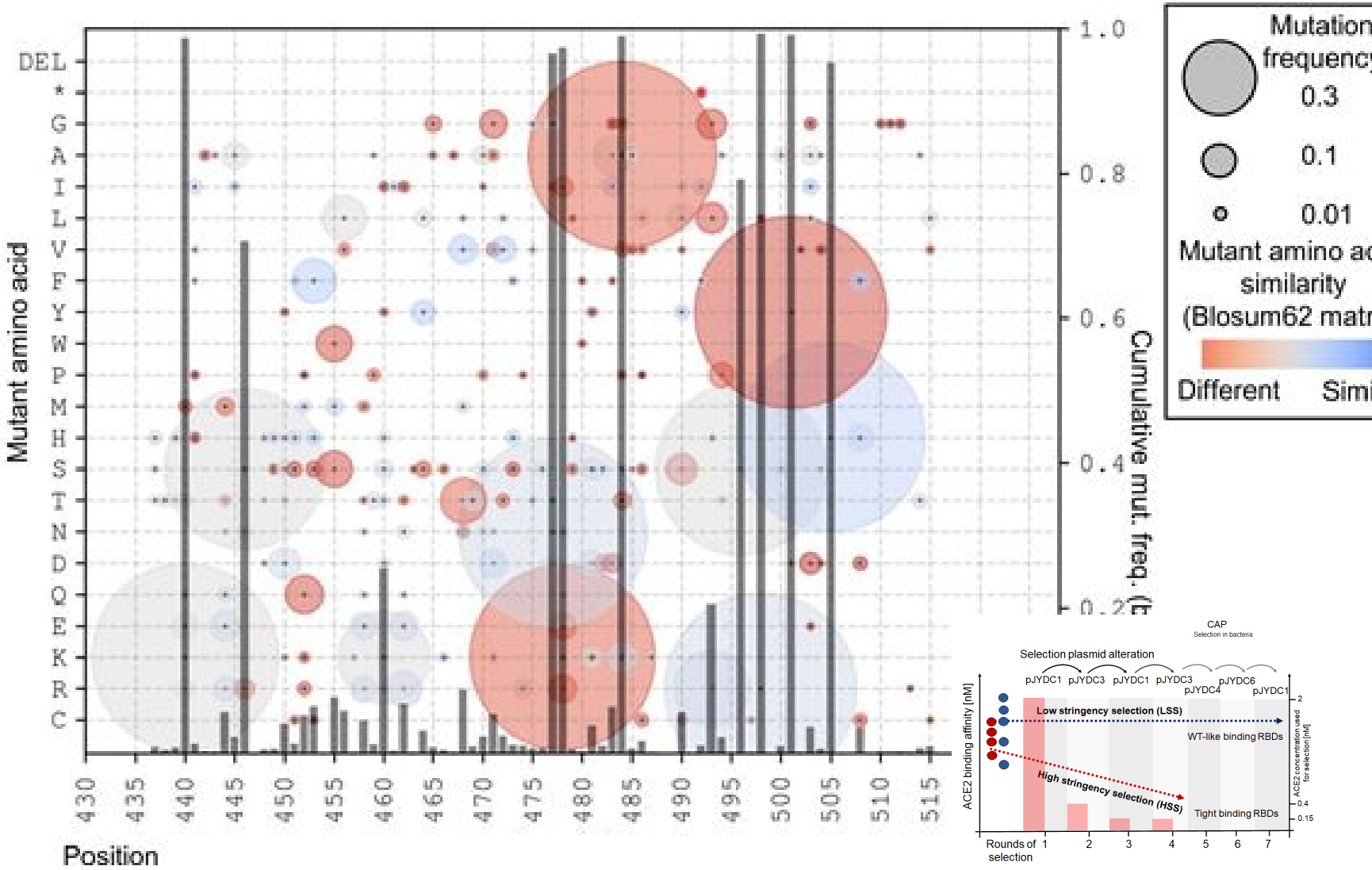
high-stringency conditions (limited ACE2 available, 2nd round)



high-stringency conditions (limited ACE2 available, 4th round)

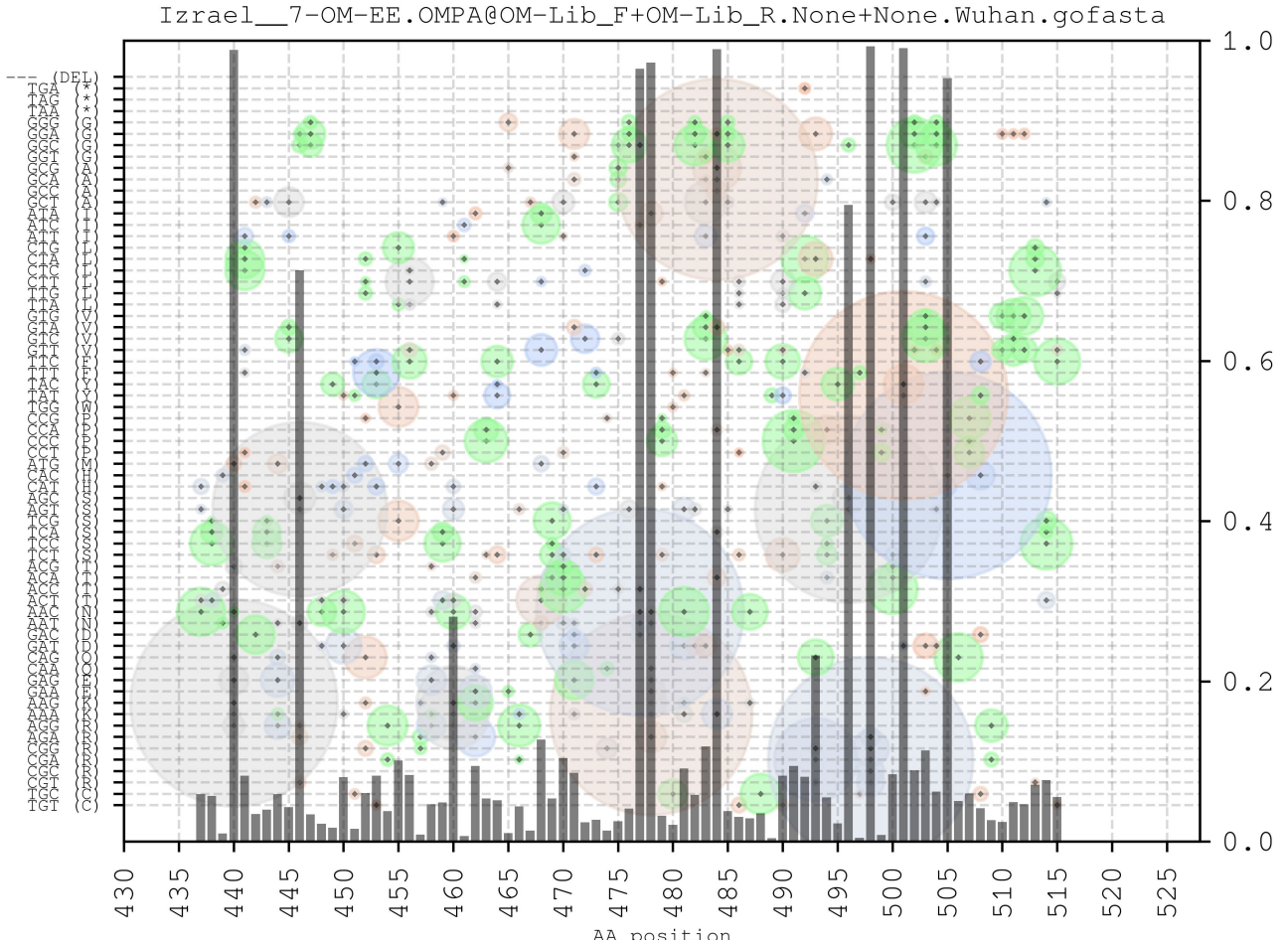


low-stringency conditions, LSS, 7th round high availability of ACE2

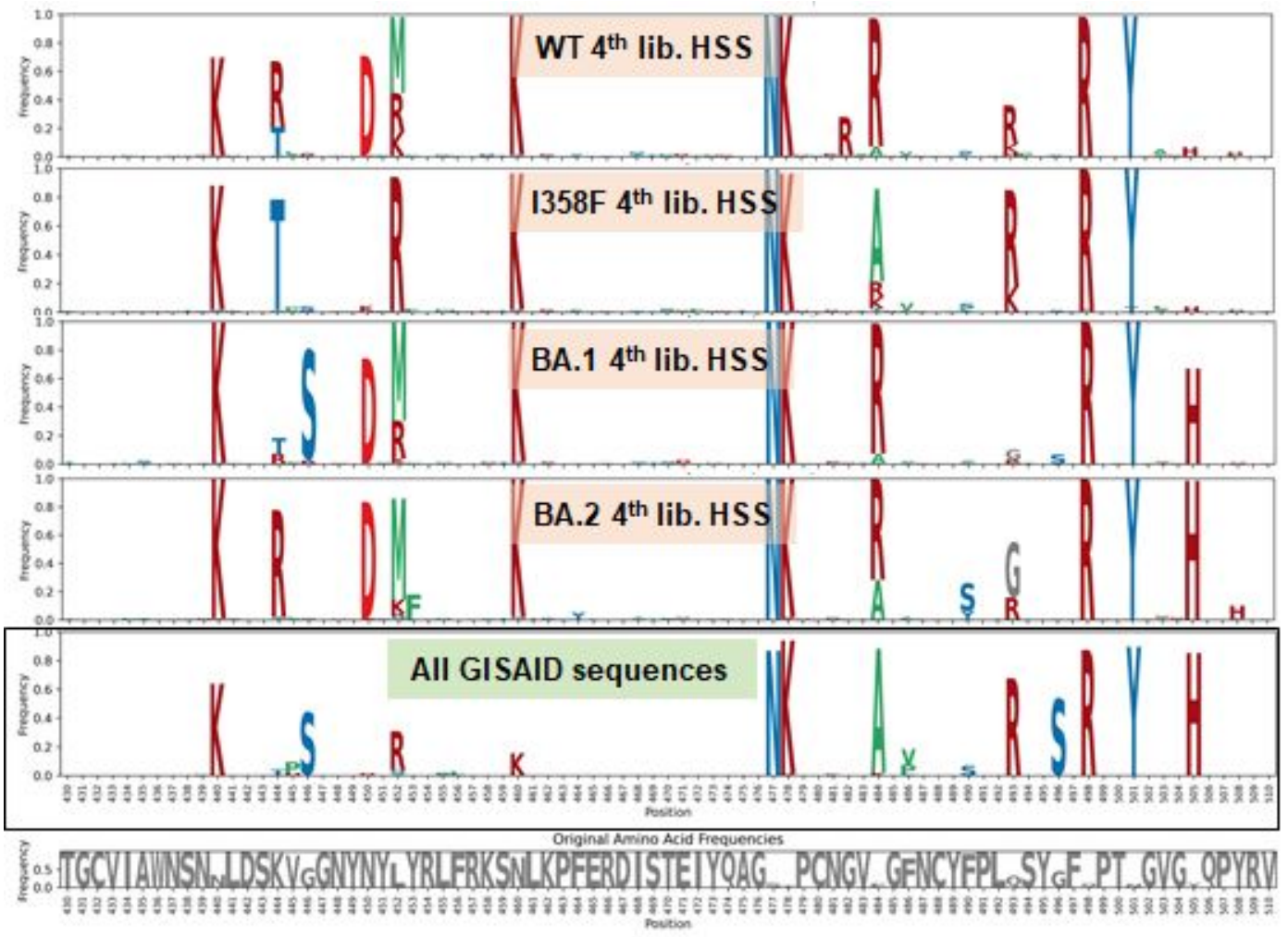


low-stringency conditions, LSS, high availability of ACE2

same as previous but rendered with codon frequencies



Stringent selection results (Logo plots) compared to real world - GISAID



Stringent Affinity Selection Drives Convergence in the RBD Towards the Mutation Profile of Omicron Lineages

Results (biology)

Deletion region around position 483 must be re-aligned manually to be codon-wise and to respect biological reality.

Upcoming: Similarly other regions inferred from GISAID we now re-align using UNIX sed (just moving the dashes around).

Upcoming: We derive multiFASTA alignment from pair-wise blastn results obeying bwa+gofasta altogether.

Results (biology)

We were lucky that omicron variant arose with two key mutations
Q498R N501Y

The virus will not have a chance to undo these two covariant mutations at once in a single mutation event.

Preprint on bioRxiv: https://doi.org/10.1101/2025.04.23.650148

Data: https://zenodo.org/records/15102607

Code: https://github.com/host-patho-evo/mutation_scatter_plot

Example figures: https://host-patho-evo.github.io/mutation_scatter_plot/

Acknowledgement

Charles University team

Jiri Zahradnik, Bailasan Haidar, Andrea Galisova, Aditi Konar, Adam Hruška, Ruojin Tian, Prokopios Andrikopoulos

Weizmann Institute of Science team

Gideon Schreiber, Jiri Zahradnik, Eyal Zoler

National Institute of Virology and Bacteriology

Funded by the European Union NextGenerationEU

CZECH RECOVERY PLAN

MINISTRY OF EDUCATION, YOUTH AND SPORTS

GAČR CZECH SCIENCE FOUNDATION

CIISB Czech Infrastructure for Integrative Structural Biology

ISRAEL SCIENCE FOUNDATION

AZV ČR CZECH HEALTH RESEARCH COUNCIL

ISIDORe