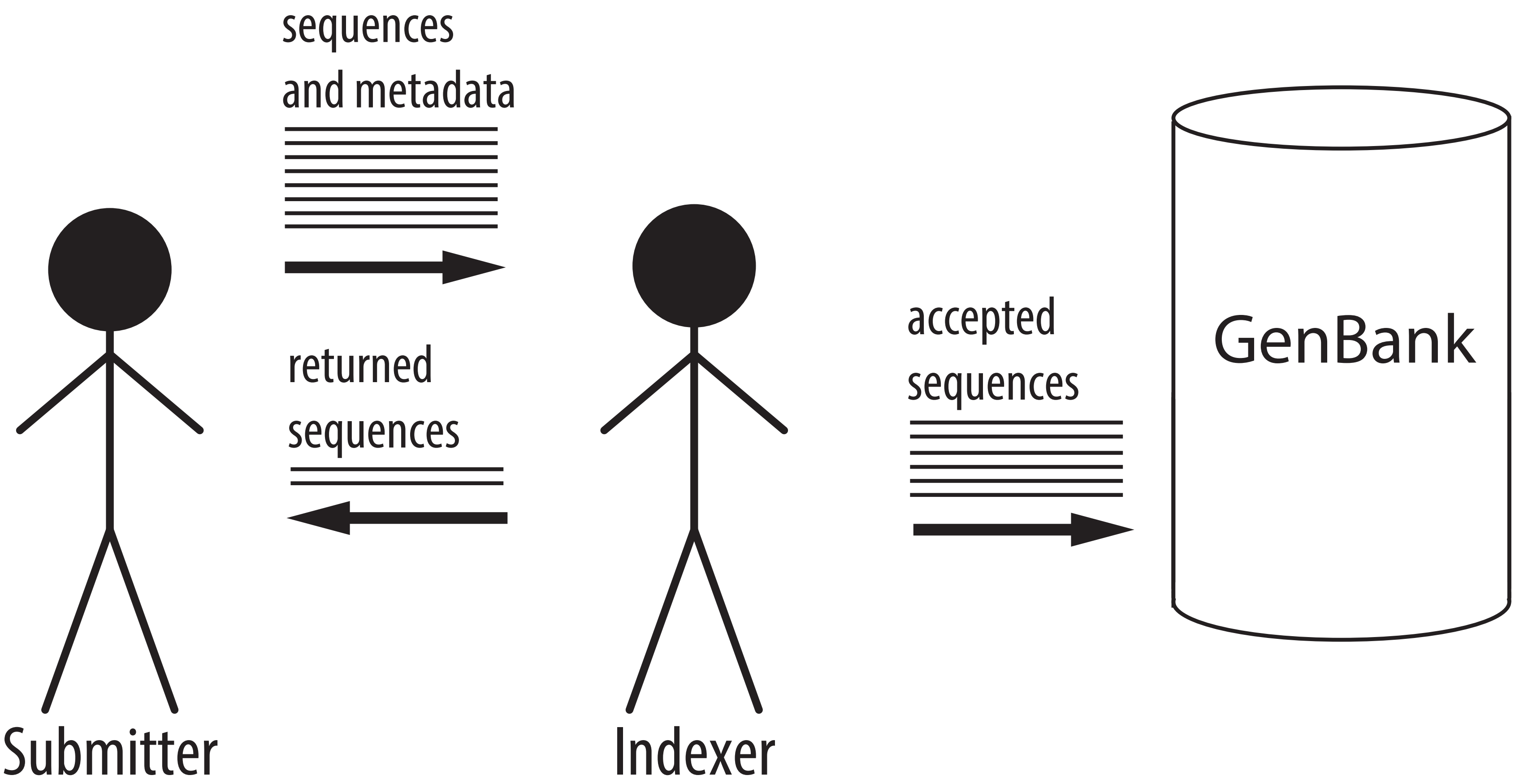
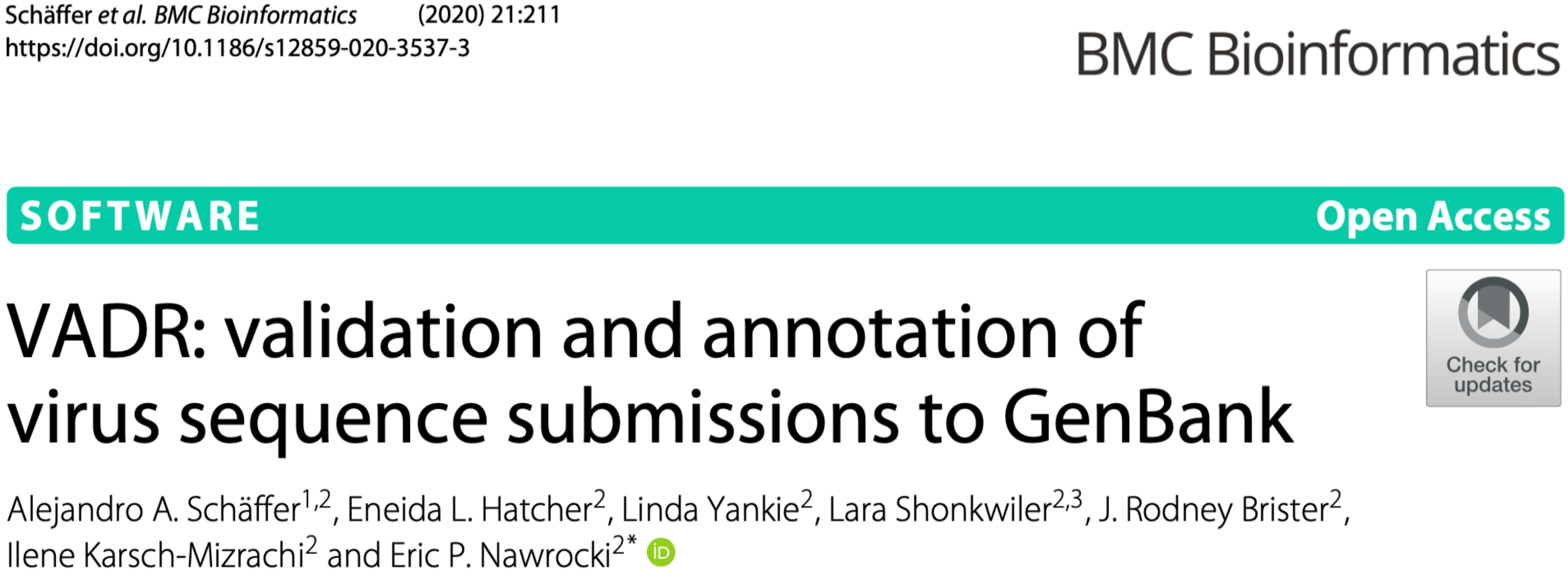
Validation and annotation of SARS-CoV-2 sequences for GenBank using VADR

Eric Nawrocki
Staff Scientist

Computational Biology Branch
National Center for Biotechnology Information
National Library of Medicine

GenBank indexers handle incoming sequence submissions





Schäffer *et al. BMC Bioinformatics* (2020) 21:211
https://doi.org/10.1186/s12859-020-3537-3

BMC Bioinformatics

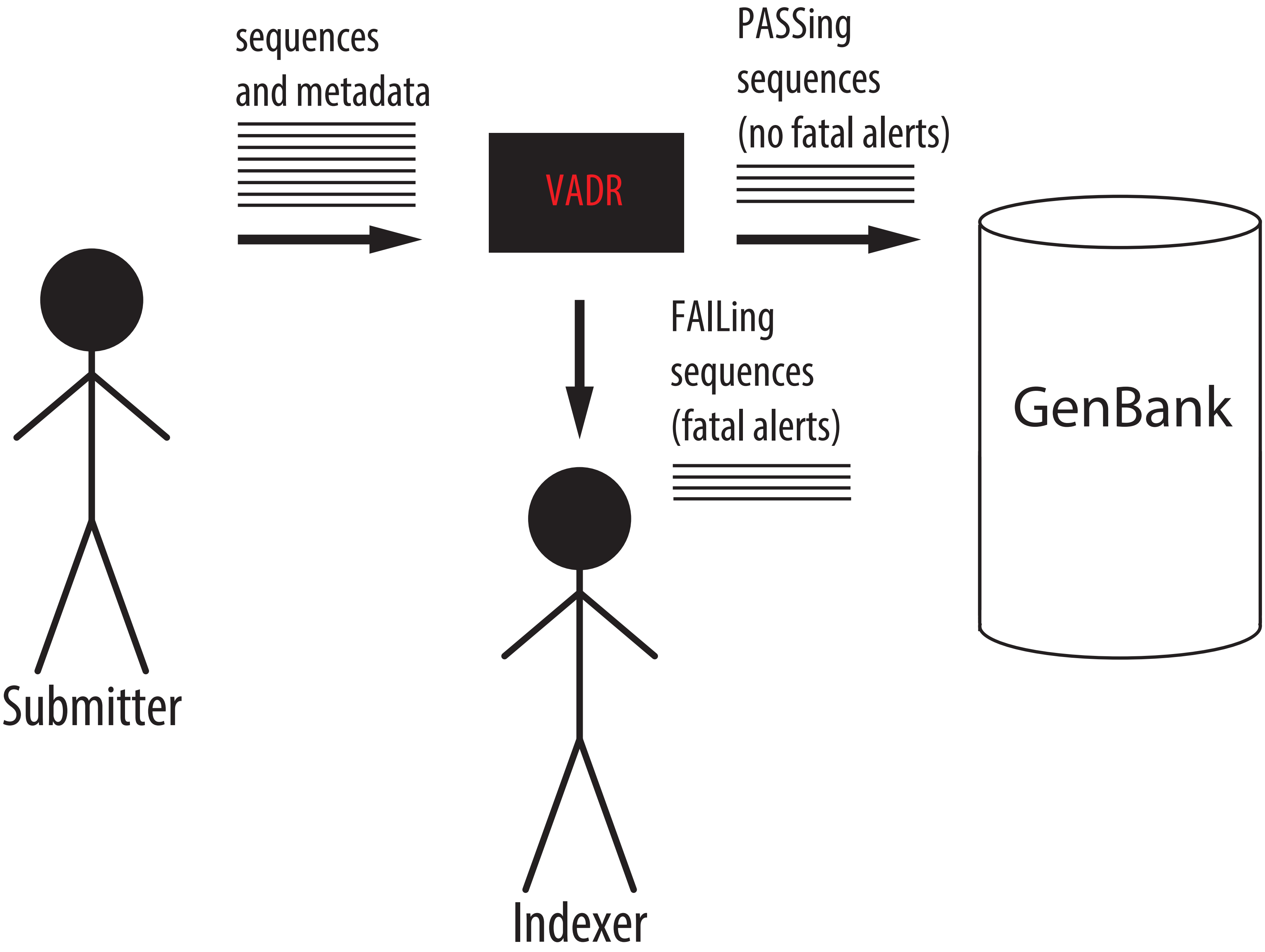
SOFTWARE **Open Access**

VADR: validation and annotation of virus sequence submissions to GenBank

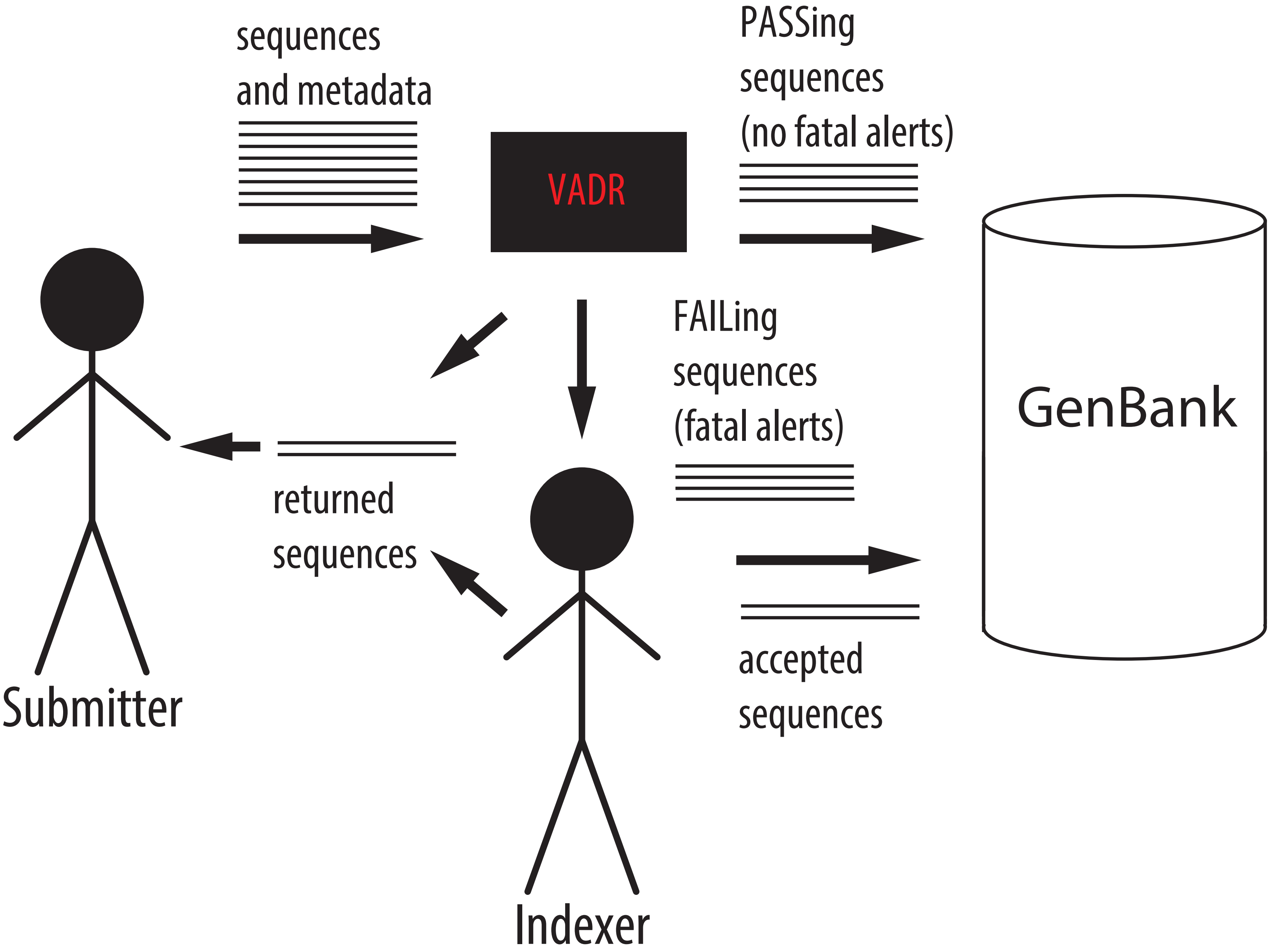
Alejandro A. Schäffer[1,2], Eneida L. Hatcher[2], Linda Yankie[2], Lara Shonkwiler[2,3], J. Rodney Brister[2], Ilene Karsch-Mizrachi[2] and Eric P. Nawrocki[2*]

- general tool for reference-based annotation of viral sequences
- used for dengue virus and norovirus submissions since 2018
- used for SARS-CoV-2 submissions since March 2020

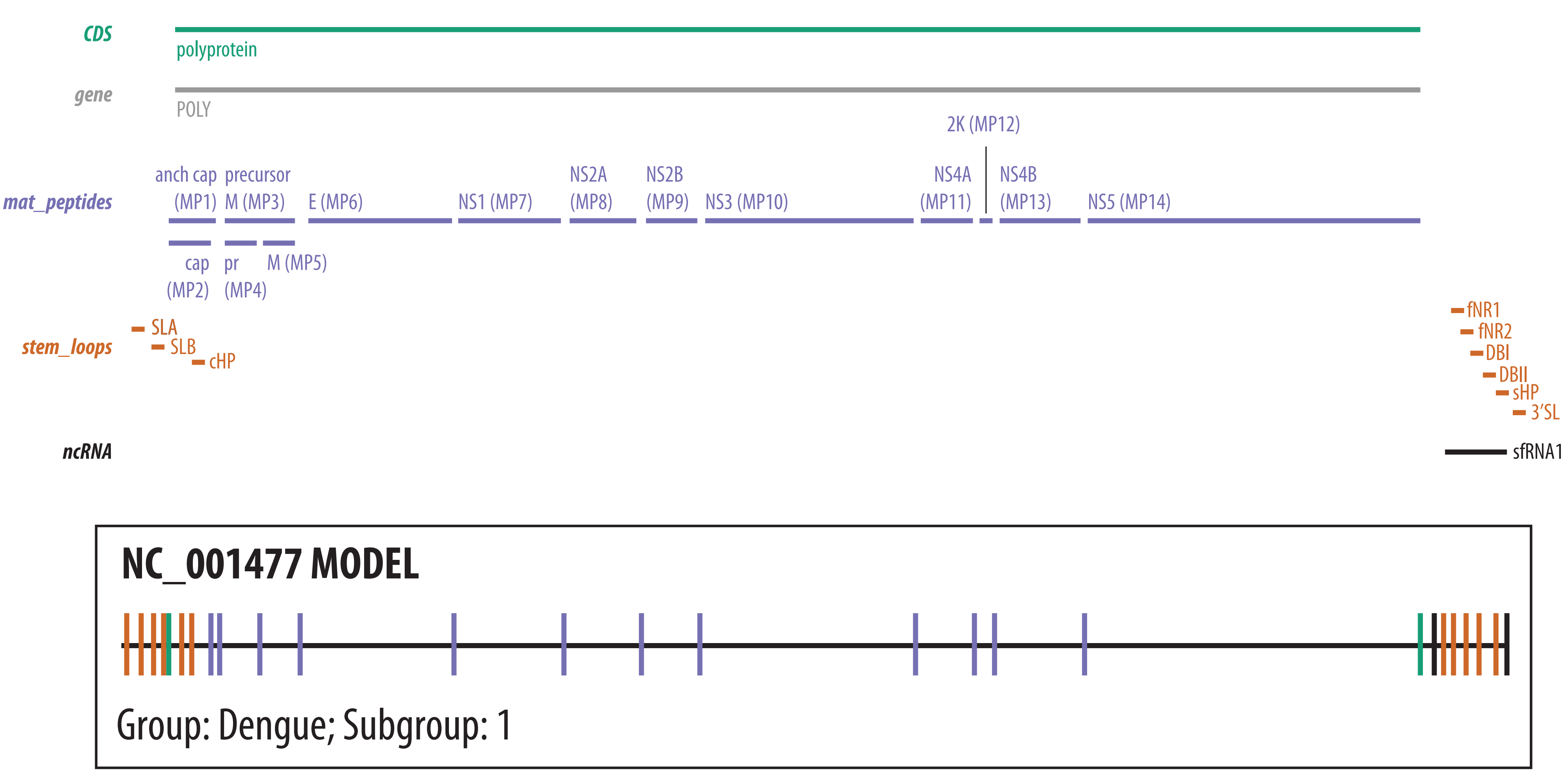
VADR assists GenBank indexers:
Each sequence PASSes or FAILs



Indexers decide fate of most FAILing sequences but some are sent directly back to submitter with error reports



VADR builds a reference model of a RefSeq and its features

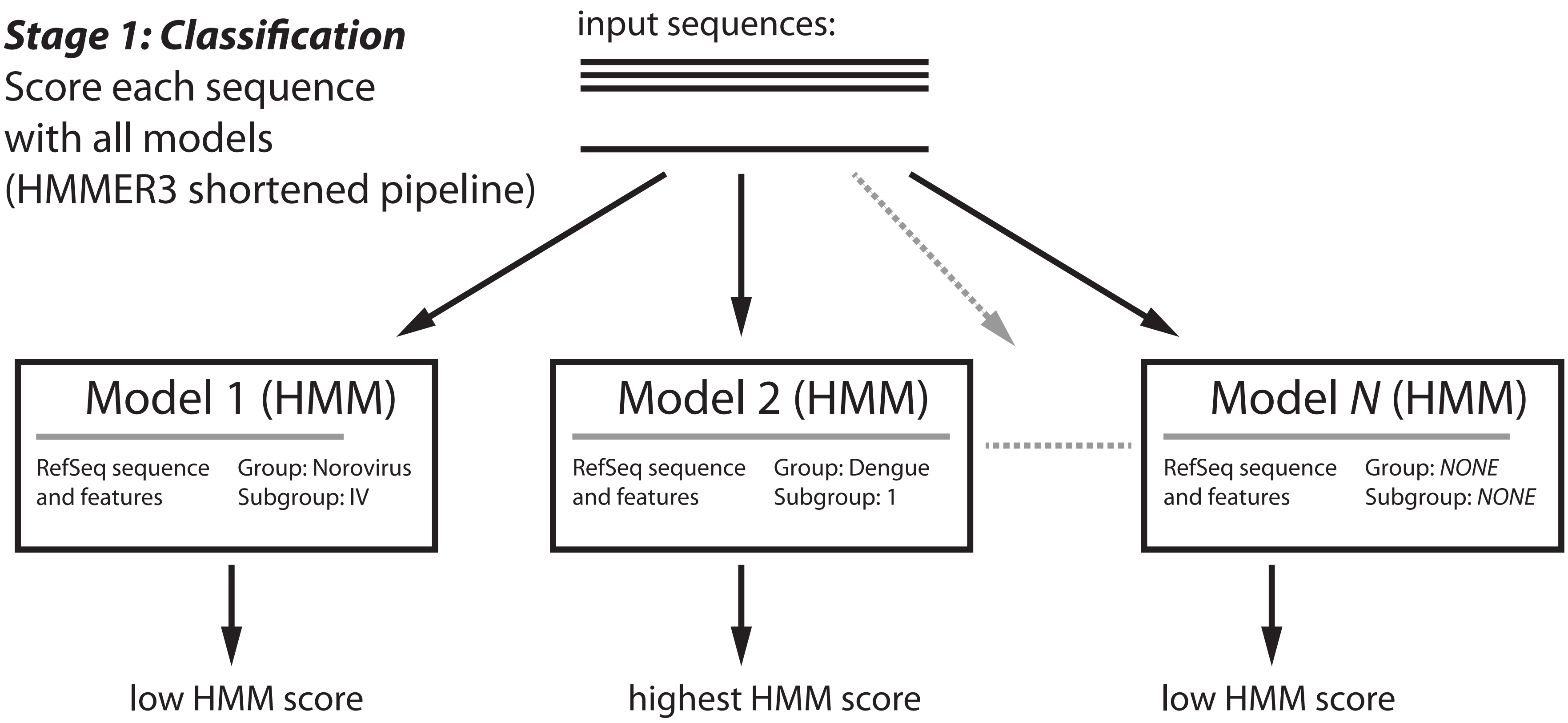


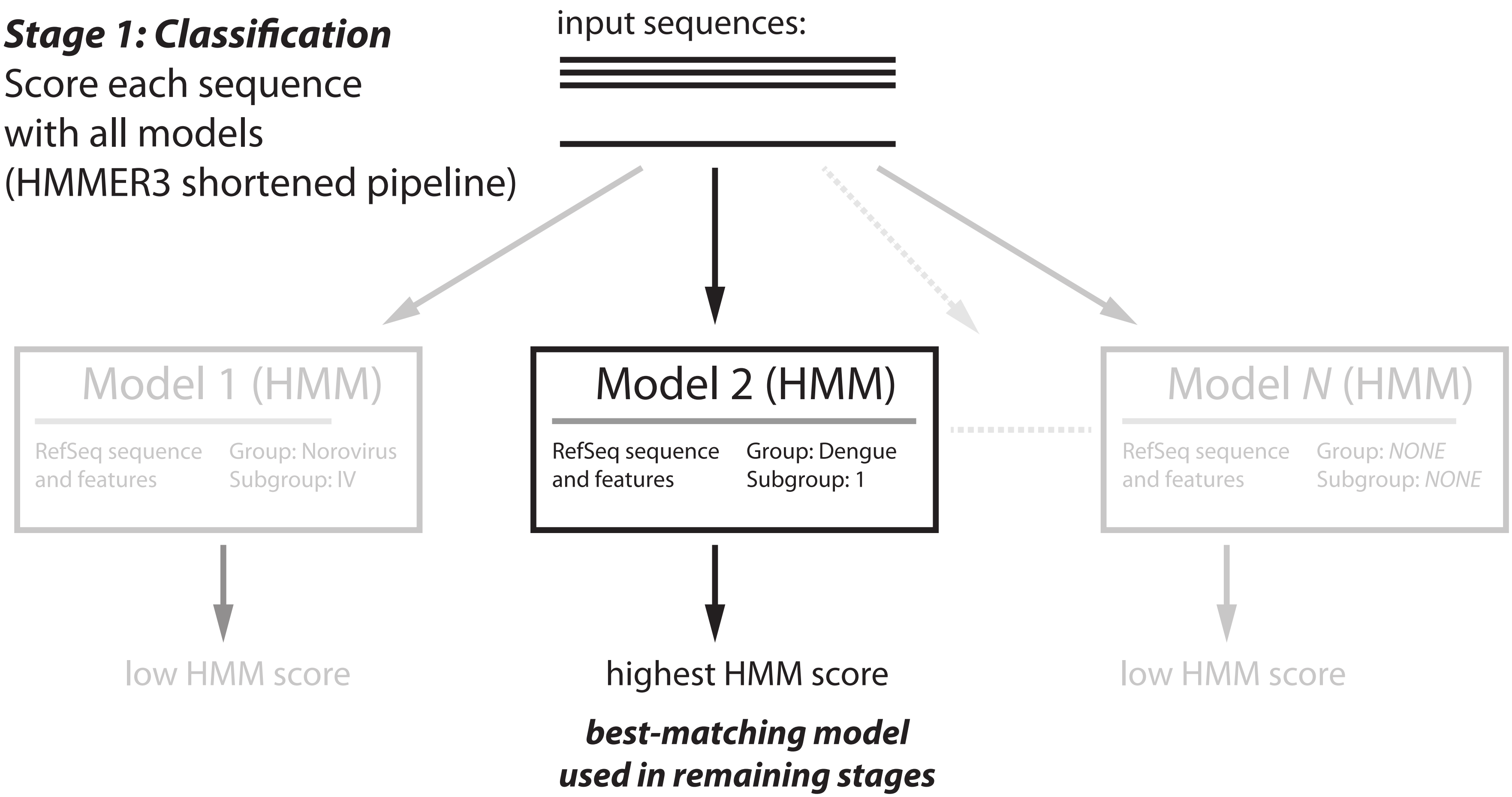
VADR validates and annotates each input sequence using its best-matching model

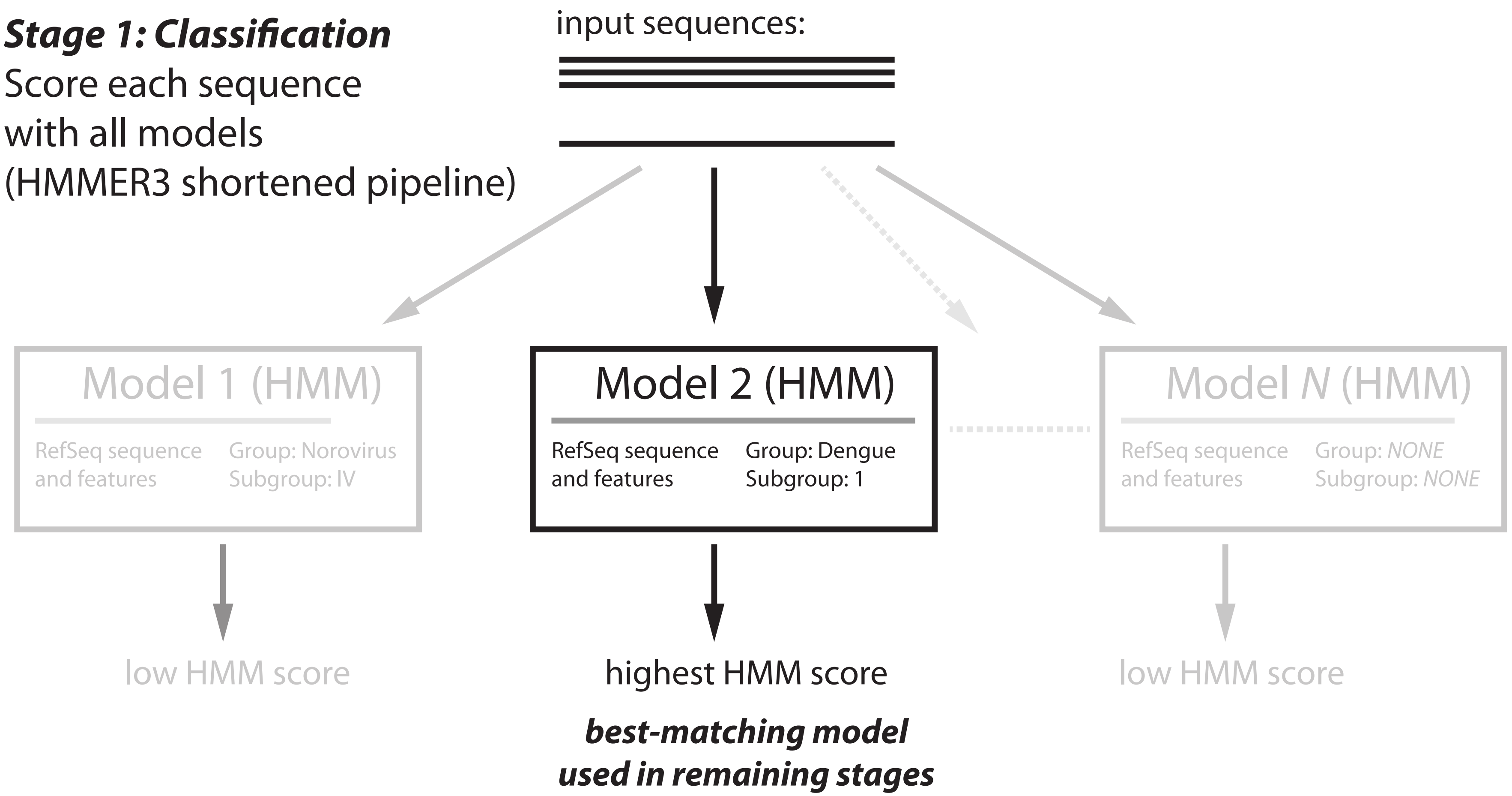
- Each sequence S proceeds through 4 stages:

  1. **Classification**
  2. **Coverage determination**
  3. **Alignment**
  4. **Protein validation**

Different types of alerts are identified and reported at each stage


Stage 1: Classification
Score each sequence
with all models
(HMMER3 shortened pipeline)
input sequences:
Model 1 (HMM)
RefSeq sequence
and features
Group: Norovirus
Subgroup: IV
Model 2 (HMM)
RefSeq sequence
and features
Group: Dengue
Subgroup: 1
Model N (HMM)
RefSeq sequence
and features
Group: NONE
Subgroup: NONE
low HMM score
highest HMM score
low HMM score

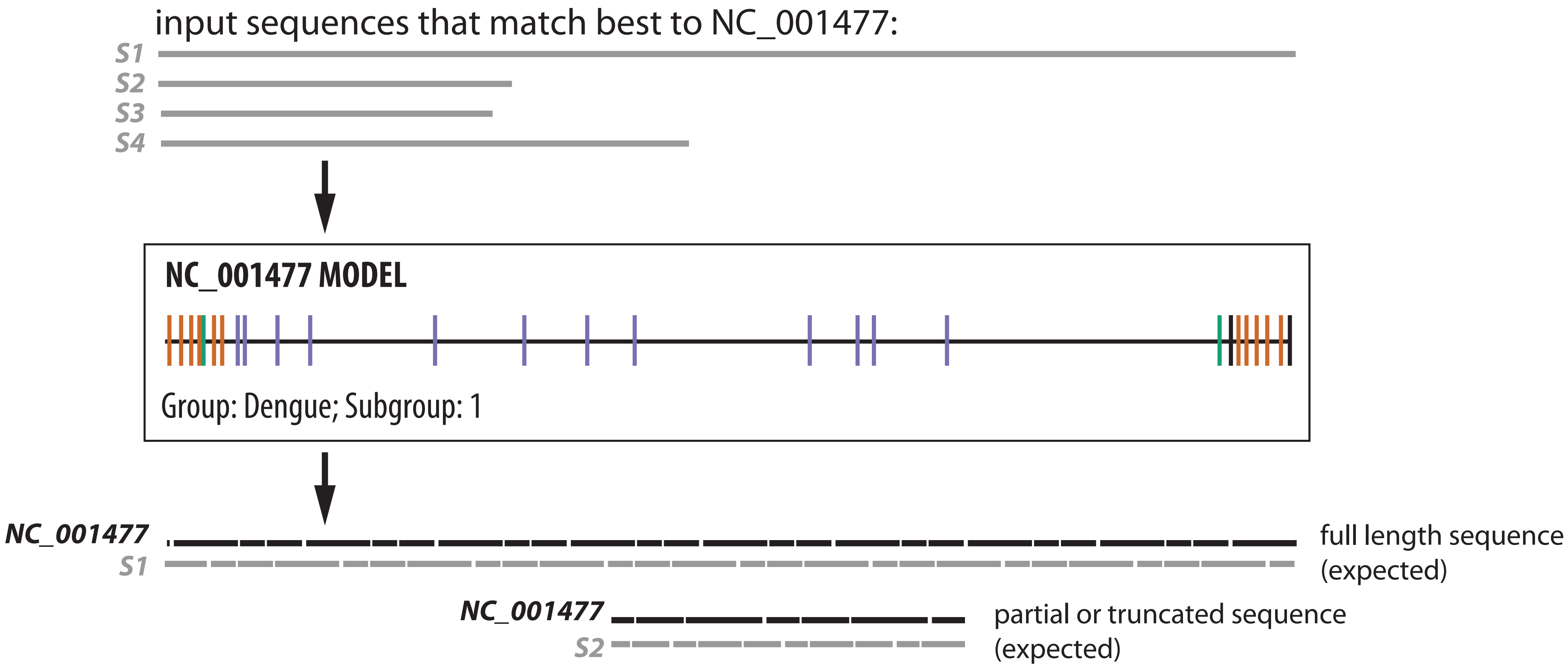

Stage 1: Classification
Score each sequence
with all models
(HMMER3 shortened pipeline)
input sequences:
Model 1 (HMM)
RefSeq sequence and features
Group: Norovirus
Subgroup: IV
Model 2 (HMM)
RefSeq sequence and features
Group: Dengue
Subgroup: 1
Model N (HMM)
RefSeq sequence and features
Group: NONE
Subgroup: NONE
low HMM score
highest HMM score
best-matching model used in remaining stages
low HMM score



code	S/F	error message	description
Fatal alerts detected in the classification stage			
noannotn*	S	NO_ANNOTATION	no significant similarity detected
revcompl*	S	REVCOMPLEM	sequence appears to be reverse complemented
incsbgrp	S	INCORRECT_SPECIFIED_SUBGROUP	score difference too large between best overall model and best specified subgroup model
incgroup	S	INCORRECT_SPECIFIED_GROUP	score difference too large between best overall model and best specified group model
Non-fatal alerts detected in the classification stage			
qstsbgrp	S	QUESTIONABLE_SPECIFIED_SUBGROUP	best overall model is not from specified subgroup
qstgroup	S	QUESTIONABLE_SPECIFIED_GROUP	best overall model is not from specified group
indfclas	S	INDEFINITE_CLASSIFICATION	low score difference between best overall model and second best model (not in best model's subgroup)
lowscore	S	LOW_SCORE	score to homology model below low threshold

Stage 2: Coverage determination

Search each sequence with best-matching model (HMMER3 full pipeline)



input sequences that match best to NC_001477:

S1

S2

S3

S4

NC_001477 MODEL

Group: Dengue; Subgroup: 1

NC_001477

S1

full length sequence (expected)

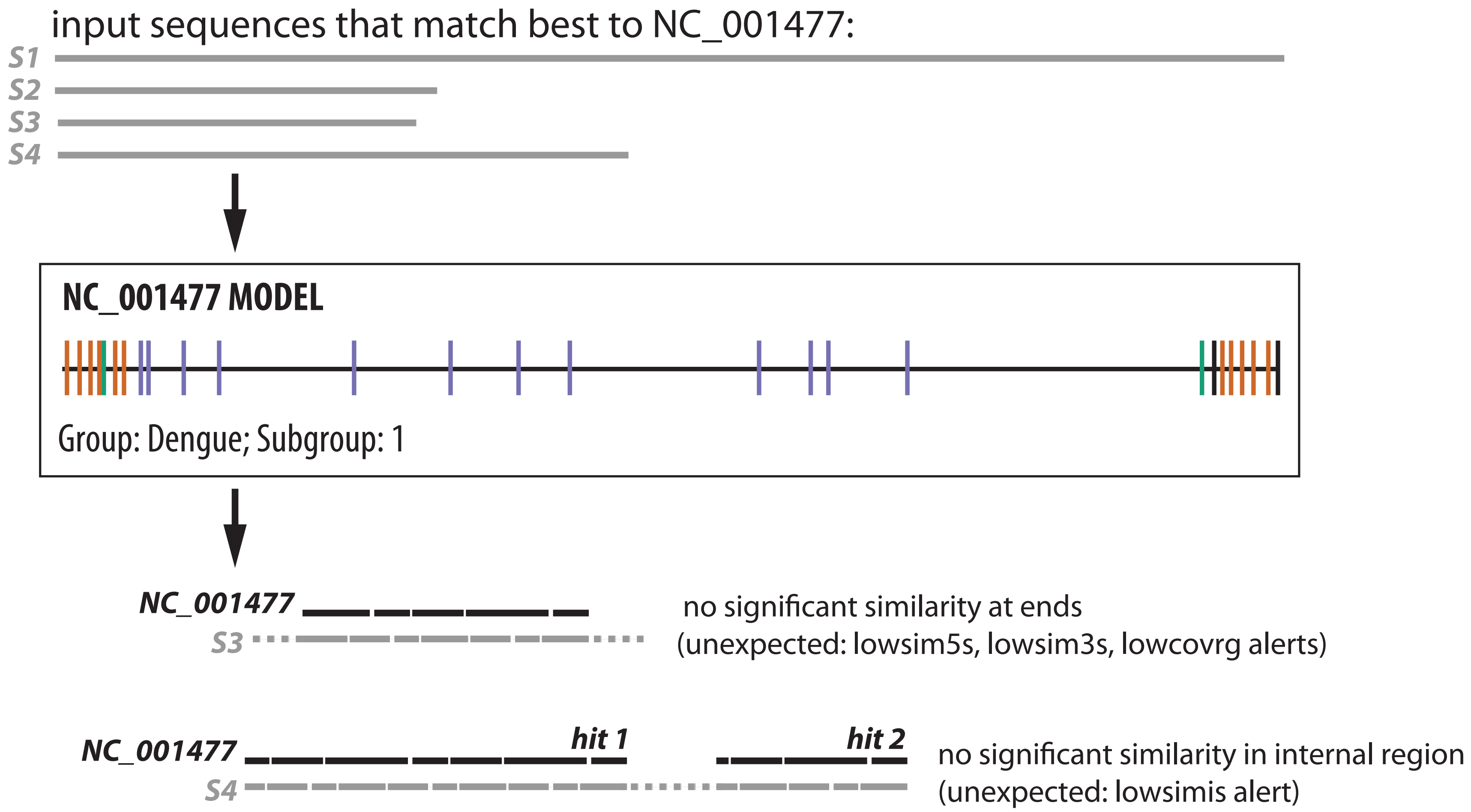
NC_001477

S2

partial or truncated sequence (expected)

Stage 2: Coverage determination

Search each sequence with best-matching model (HMMER3 full pipeline)


input sequences that match best to NC_001477:

no significant similarity at ends
(unexpected: lowsim5s, lowsim3s, lowcovrg alerts)

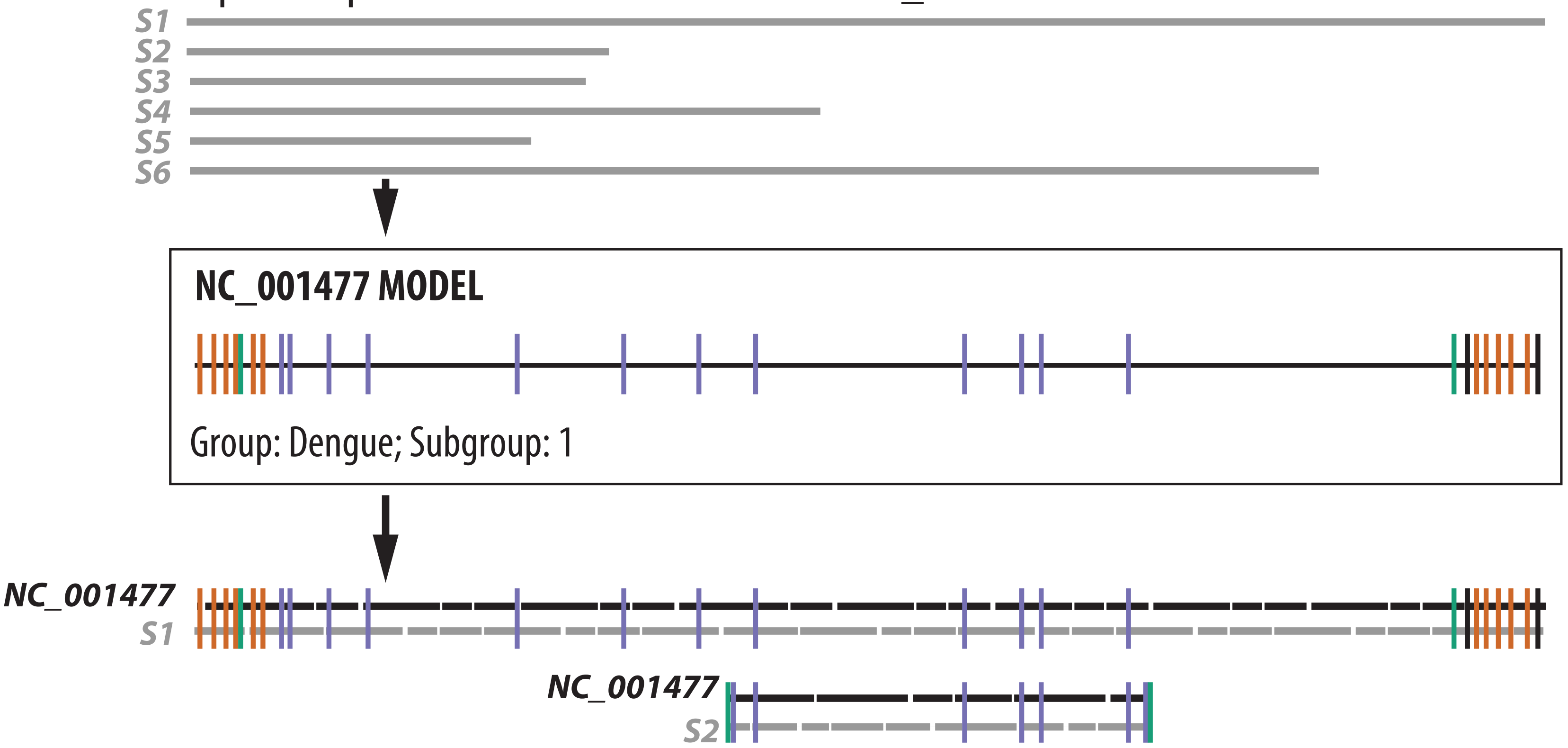
no significant similarity in internal region
(unexpected: lowsimis alert)

code	S/F	error message	description
Fatal alerts detected in the coverage stage			
lowcovrg	S	LOW_COVERAGE	low sequence fraction with significant similarity to homology model
dupregin	S	DUPLICATE_REGIONS	similarity to a model region occurs more than once
discontn	S	DISCONTINUOUS_SIMILARITY	not all hits are in the same order in the sequence and the homology model
indfstrn	S	INDEFINITE_STRAND	significant similarity detected on both strands
lowsim5s	S	LOW_SIMILARITY_START	significant similarity not detected at 5' end of the sequence
lowsim3s	S	LOW_SIMILARITY_END	significant similarity not detected at 3' end of the sequence
lowsimis	S	LOW_SIMILARITY	internal region without significant similarity
Non-fatal alerts detected in the coverage stage			
biasdseq	S	BIASED_SEQUENCE	high fraction of score attributed to biased sequence composition

Stage 3: Alignment and feature mapping

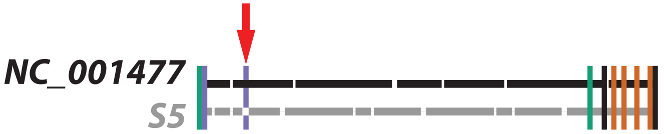
Align each sequence to its best-matching model (Infernal's cmalign)

input sequences that match best to NC_001477:

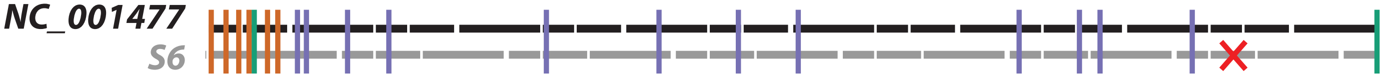


Stage 3: Alignment and feature mapping

Align each sequence to its best-matching model (Infernal's cmalign)



gap or low confidence at feature boundary
(indf5gap, indf3gap, indf5loc, indf3loc alert)

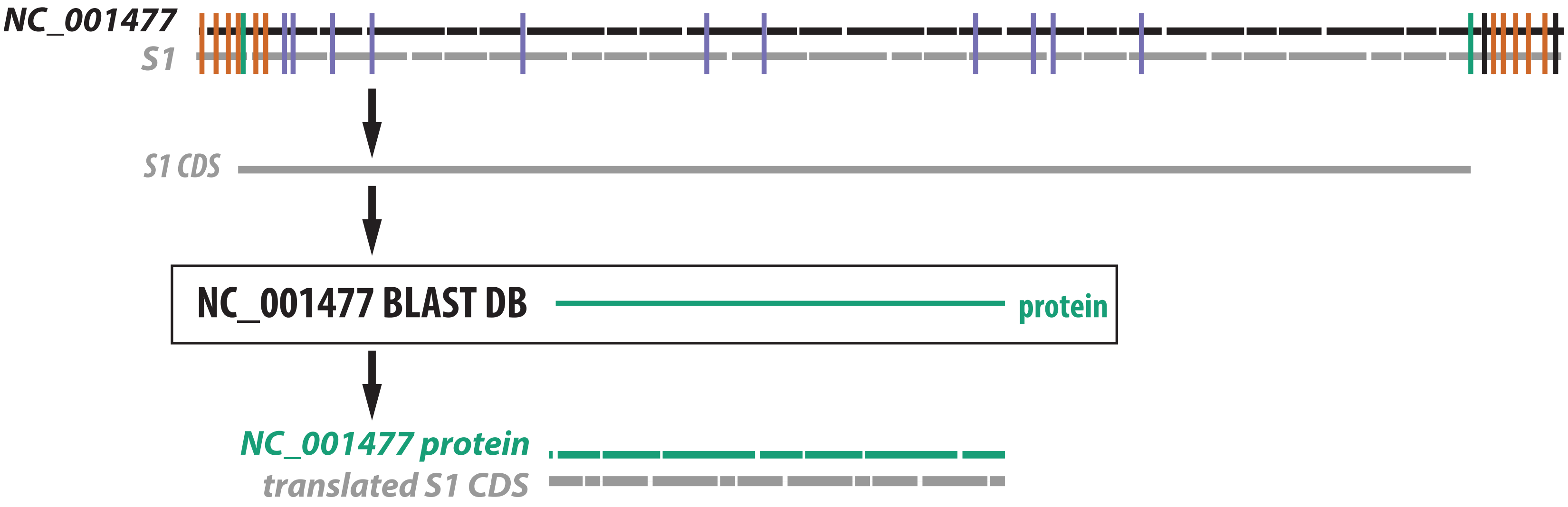


early stop codon in CDS
(cdsstopn alert)

code	**S/F**	**error message**	**description**
Fatal alerts detected in the annotation stage			
unexdivg*	S	UNEXPECTED_DIVERGENCE	sequence is too divergent to confidently assign nucleotide-based annotation
noftrann*	S	NO_FEATURES_ANNOTATED	sequence similarity to homology model does not overlap with any features
mutstart	F	MUTATION_AT_START	expected start codon could not be identified
mutendcd	F	MUTATION_AT_END	expected stop codon could not be identified, predicted CDS stop by homology is invalid
mutendns	F	MUTATION_AT_END	expected stop codon could not be identified, no in-frame stop codon exists 3' of predicted valid start codon
mutendex	F	MUTATION_AT_END	expected stop codon could not be identified, first in-frame stop codon exists 3' of predicted stop position
unexleng	F	UNEXPECTED_LENGTH	length of complete coding (CDS or mat_peptide) feature is not a multiple of 3
cdsstopn	F	CDS_HAS_STOP_CODON	in-frame stop codon exists 5' of stop position predicted by homology to reference
peptrans	F	PEPTIDE_TRANSLATION_PROBLEM	mat_peptide may not be translated because its parent CDS has a problem
pepadjcy	F	PEPTIDE_ADJACENCY_PROBLEM	predictions of two mat_peptides expected to be adjacent are not adjacent
indfantn	F	INDEFINITE_ANNOTATION	nucleotide-based search identifies CDS not identified in protein-based search
indf5gap	F	INDEFINITE_ANNOTATION_START	alignment to homology model is a gap at 5' boundary
indf5loc	F	INDEFINITE_ANNOTATION_START	alignment to homology model has low confidence at 5' boundary
indf3gap	F	INDEFINITE_ANNOTATION_END	alignment to homology model is a gap at 3' boundary
indf3loc	F	INDEFINITE_ANNOTATION_END	alignment to homology model has low confidence at 3' boundary
lowsim5f	F	LOW_FEATURE_SIMILARITY_START	region within annotated feature at 5' end of sequence lacks significant similarity
lowsim3f	F	LOW_FEATURE_SIMILARITY_END	region within annotated feature at 3' end of sequence lacks significant similarity
lowsimif	F	LOW_FEATURE_SIMILARITY	region within annotated feature lacks significant similarity

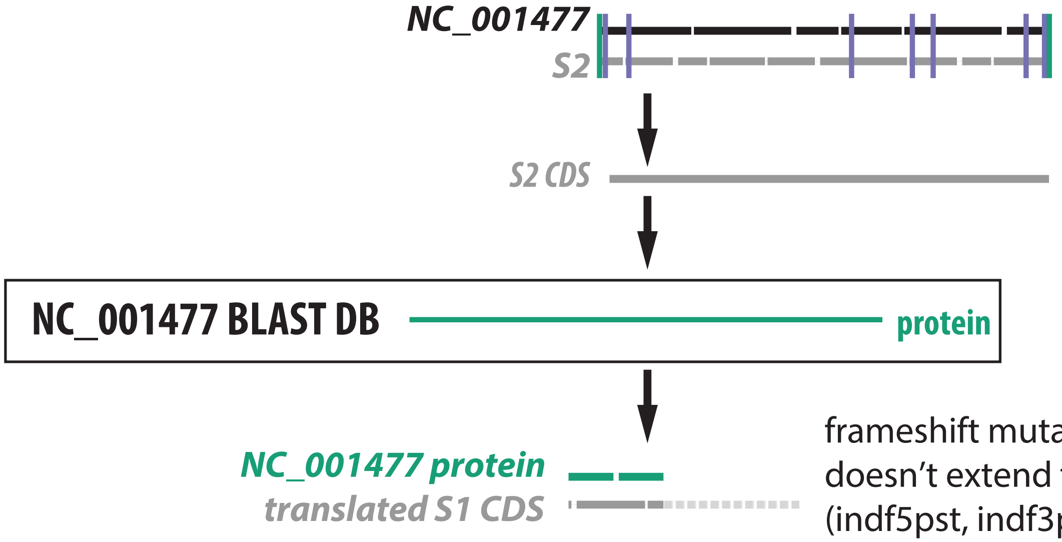
Stage 4: Protein validation (Alejandro Schäffer)

Compare each predicted CDS to model (RefSeq) proteins with BLASTX



Stage 4: Protein validation (Alejandro Schäffer)

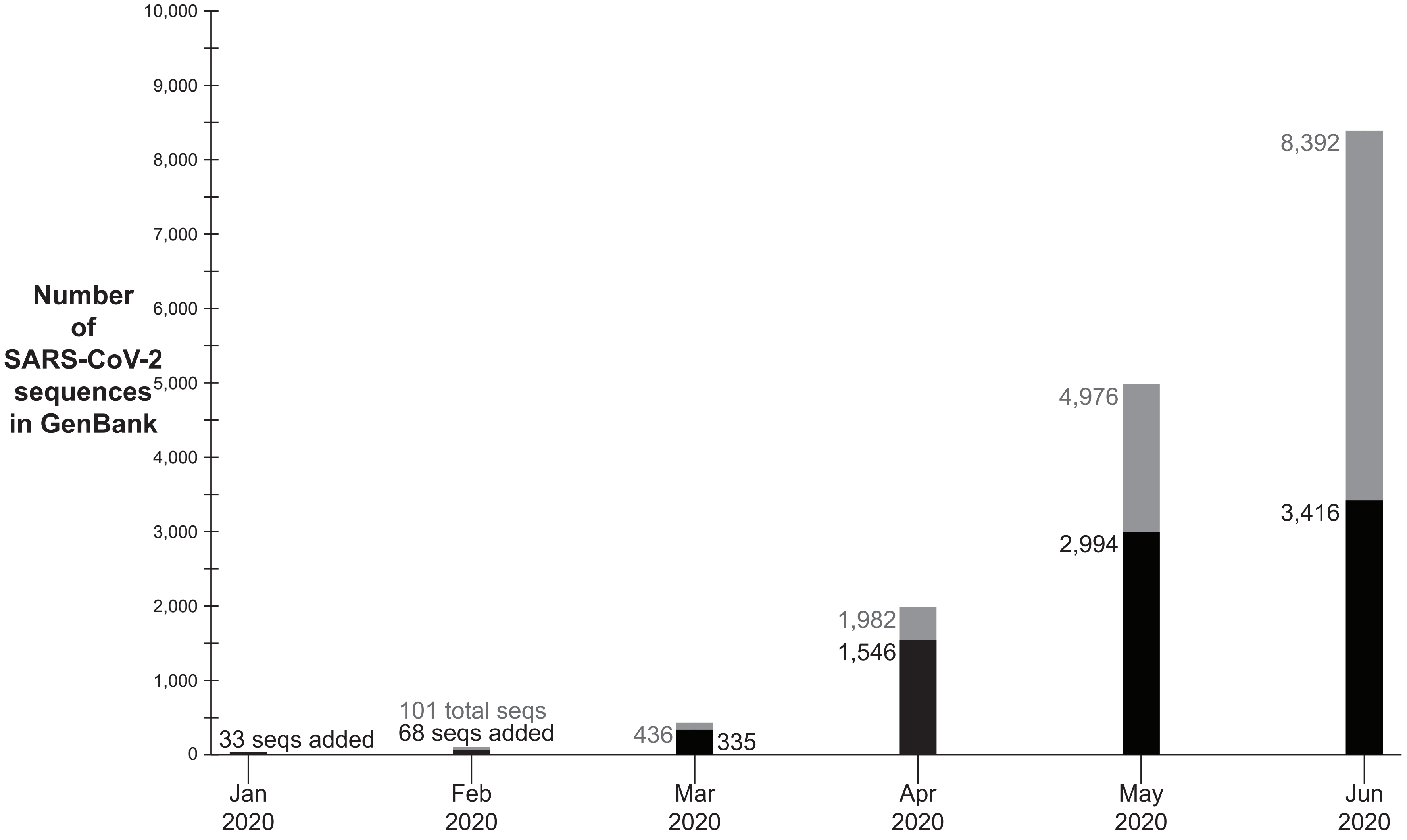
Compare each predicted CDS to model (RefSeq) proteins with BLASTX



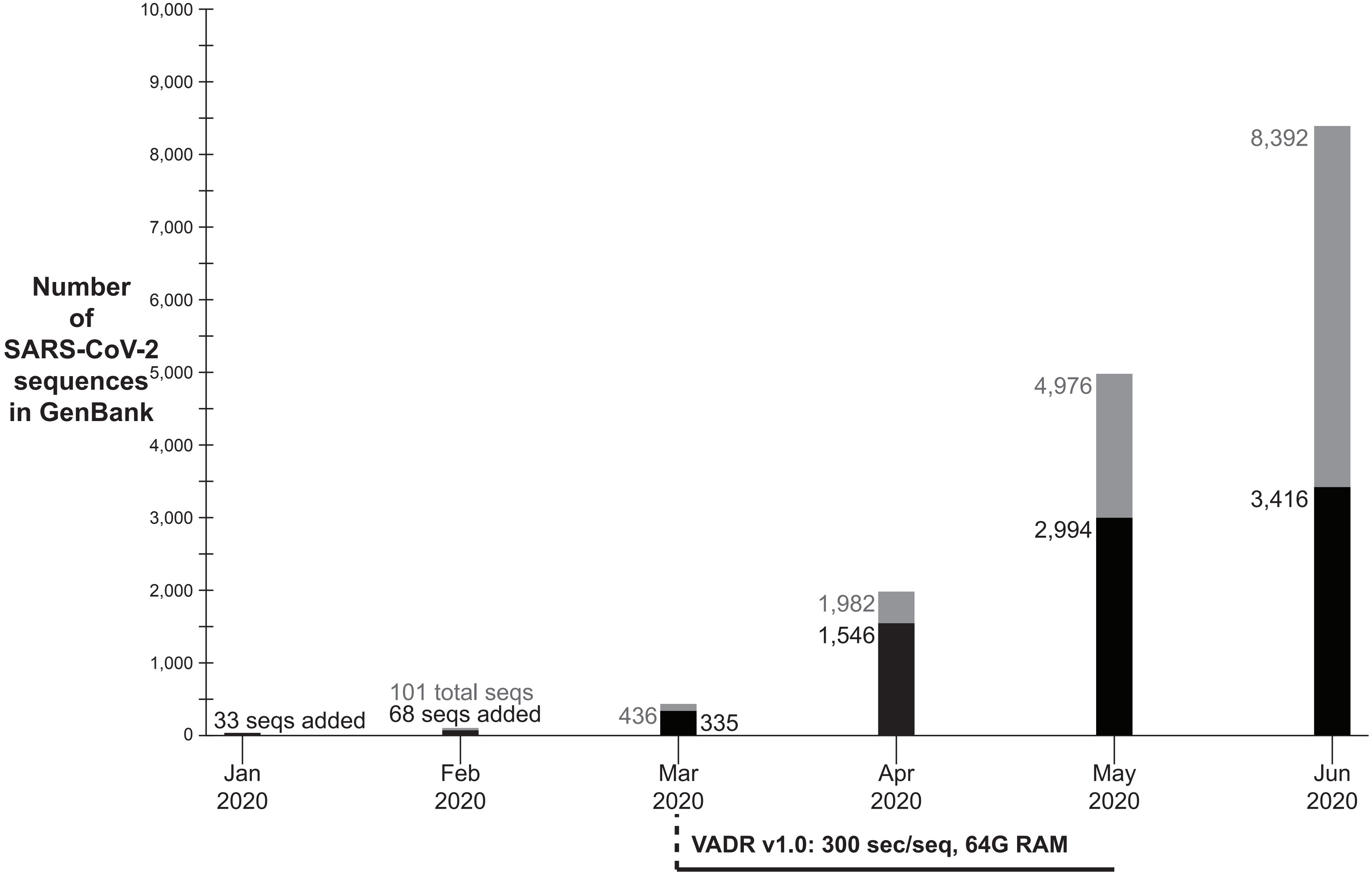
frameshift mutation; BLASTX alignment doesn't extend full length of CDS prediction (indf5pst, indf3pst alerts)

code	S/F	error message	description
Fatal alerts detected in the protein validation stage			
cdsstopp	F	CDS_HAS_STOP_CODON	stop codon in protein-based alignment
indfantp	F	INDEFINITE_ANNOTATION	protein-based search identifies CDS not identified in nucleotide-based search
indf5plg	F	INDEFINITE_ANNOTATION_START	protein-based alignment extends past nucleotide-based alignment at 5' end
indf5pst	F	INDEFINITE_ANNOTATION_START	protein-based alignment does not extend close enough to nucleotide -based alignment 5' endpoint
indf3plg	F	INDEFINITE_ANNOTATION_END	protein-based alignment extends past nucleotide-based alignment at 3' end
indf3pst	F	INDEFINITE_ANNOTATION_END	protein-based alignment does not extend close enough to nucleotide -based alignment 3' endpoint
indfstrp	F	INDEFINITE_STRAND	strand mismatch between protein-based and nucleotide-based predictions
insertnp	F	INSERTION_OF_NT	too large of an insertion in protein-based alignment
deletinp	F	DELETION_OF_NT	too large of a deletion in protein-based alignment

SARS-CoV-2 sequences in GenBank: Jan 2020 to June 2020



VADR 1.0: functional but slow

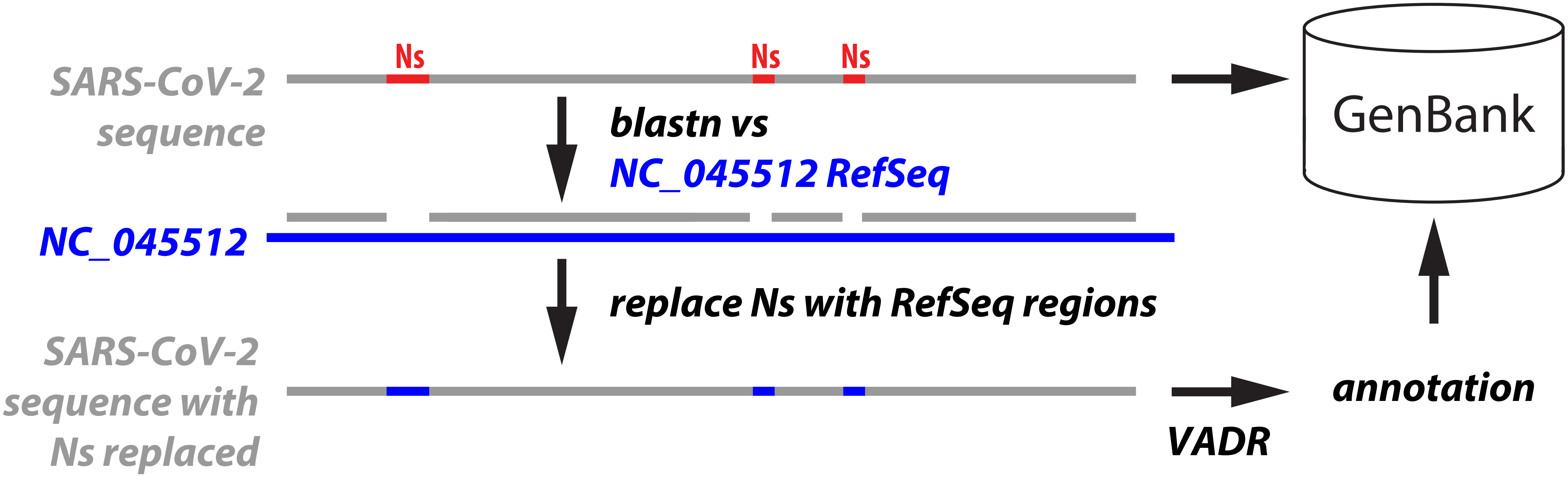


SARS-CoV-2 sequences have a lot of ambiguous nucleotides (Ns)

virus	% of nucleotides that are Ns	% of seqs w/stretch of Ns >= 50 nt
Dengue virus	0.0037%	0.0070%
Norovirus	0.296%	0.628%
SARS-CoV-2	1.12%	*26.4%*

SARS-CoV-2 sequences have a lot of ambiguous nucleotides (Ns)

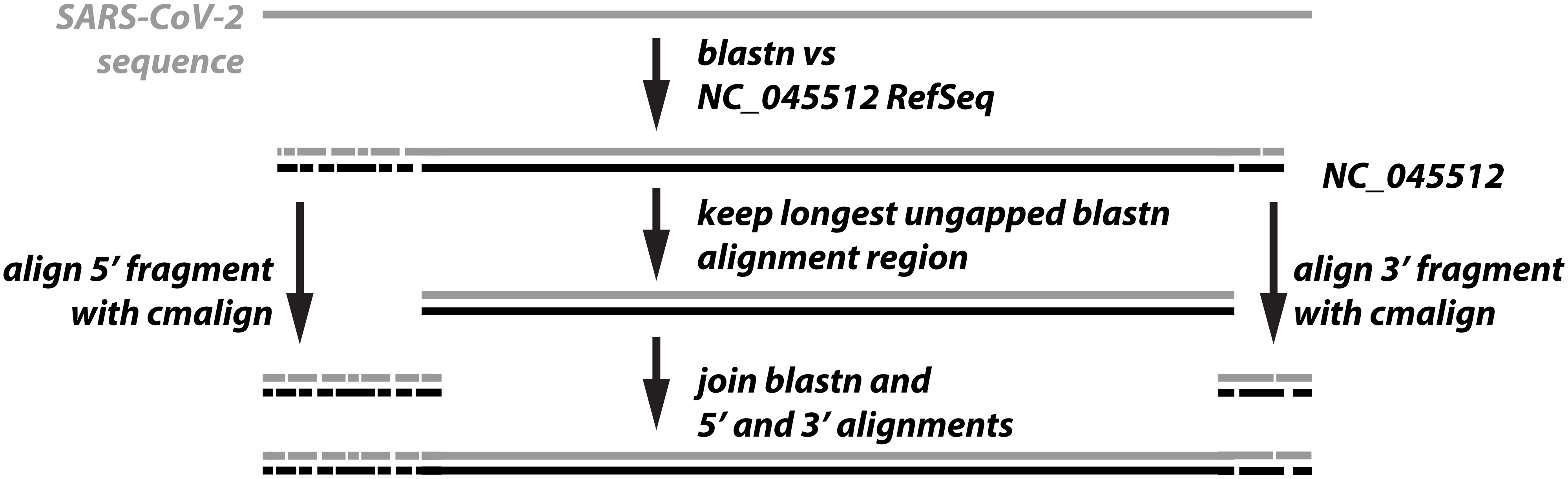
virus	% of nucleotides that are Ns	% of seqs w/stretch of Ns >= 50 nt
Dengue virus	0.0037%	0.0070%
Norovirus	0.296%	0.628%
SARS-CoV-2	1.12%	*26.4%*


SARS-CoV-2 sequence
Ns
Ns
Ns
blastn vs NC_045512 RefSeq
GenBank
NC_045512
replace Ns with RefSeq regions
SARS-CoV-2 sequence with Ns replaced
annotation
VADR

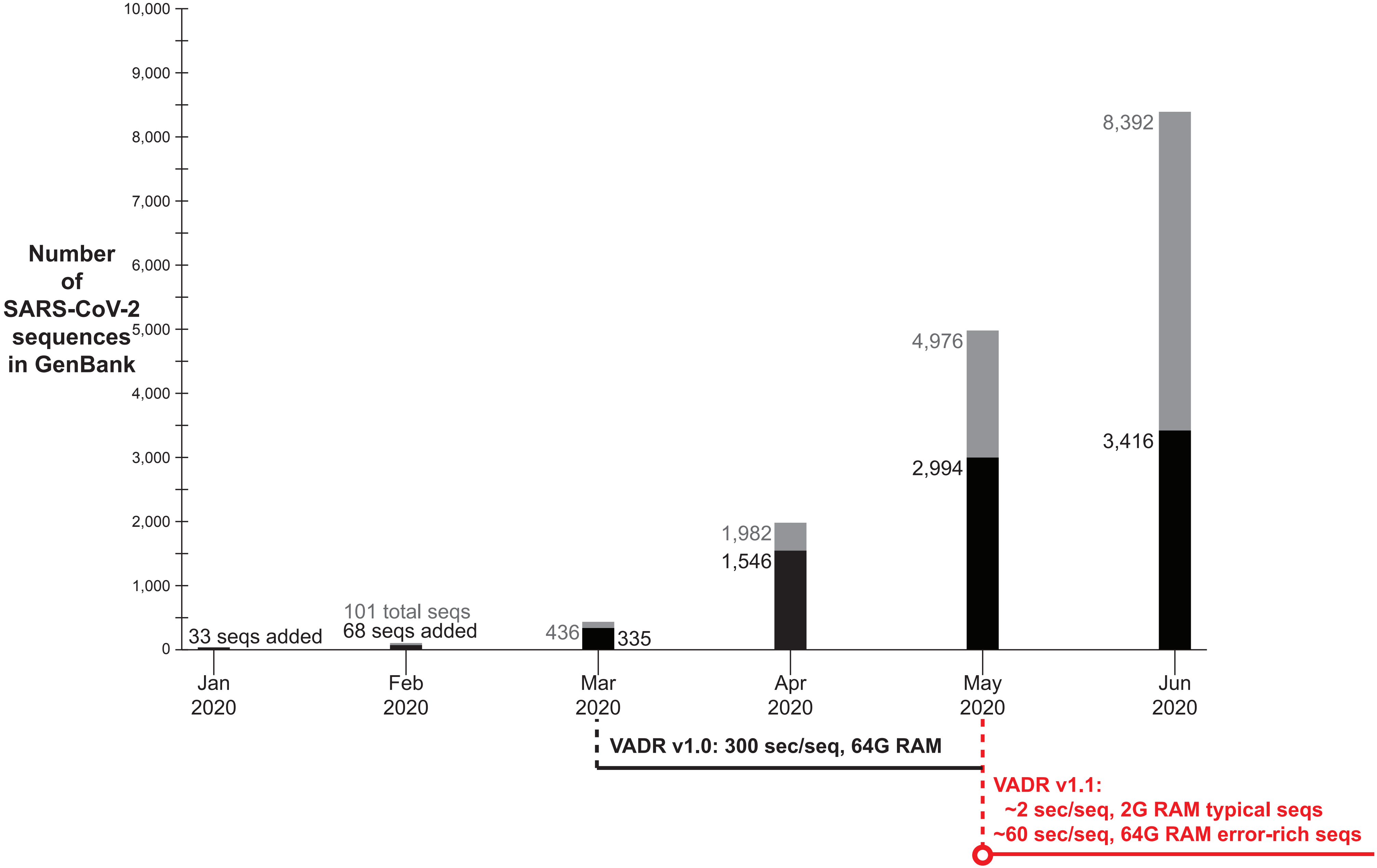
VADR 1.1 exploits high similarity (typically > 99.5%) of SARS-CoV-2 sequences to the RefSeq

- blastn replaces hmmer3 in classification and coverage determination stages

- max ungapped blastn alignment region *seeds* the cmalign alignment


SARS-CoV-2 sequence
blastn vs NC_045512 RefSeq
NC_045512
keep longest ungapped blastn alignment region
align 5' fragment with cmalign
align 3' fragment with cmalign
join blastn and 5' and 3' alignments

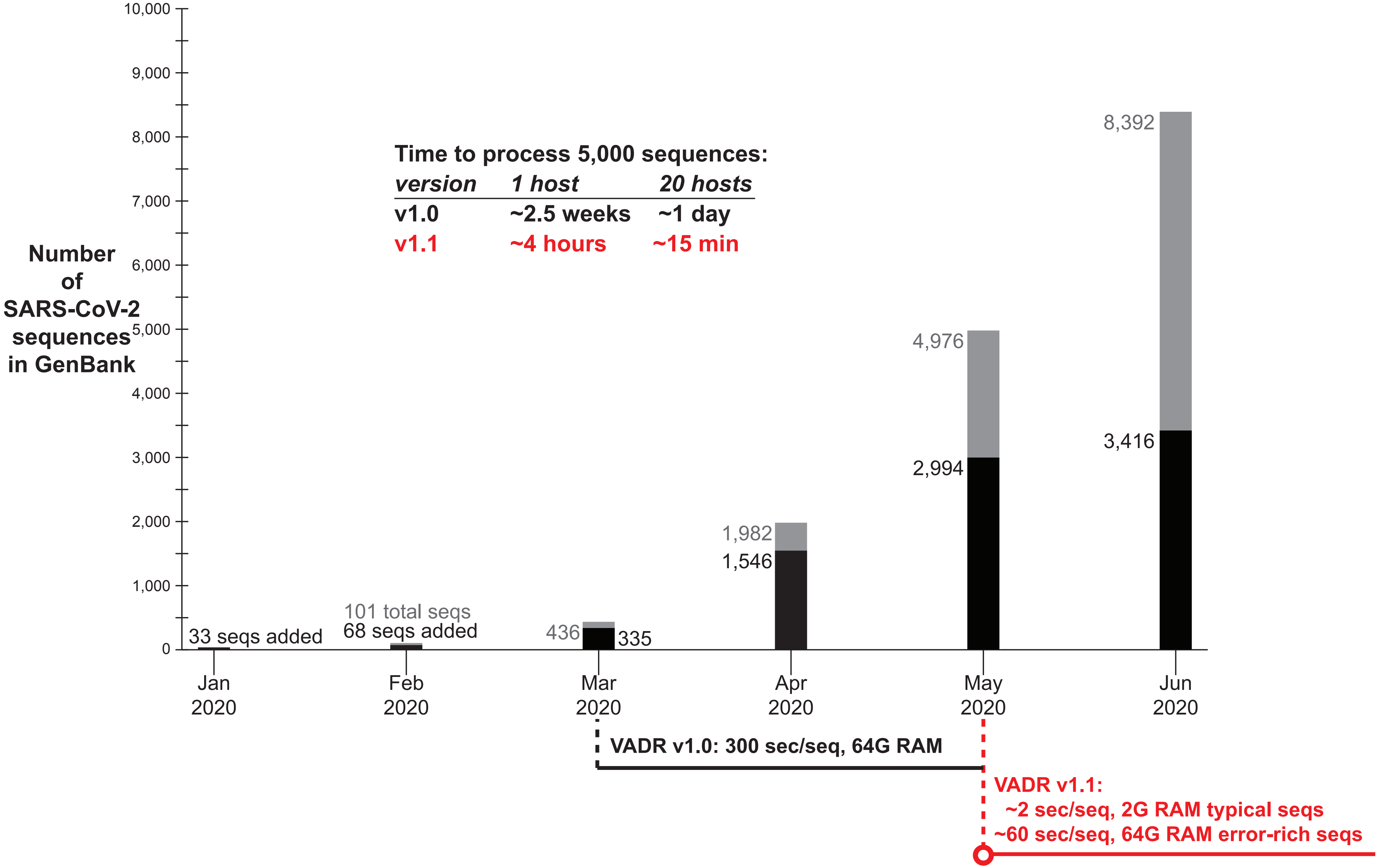
VADR 1.1: 150X speedup on typical sequences



VADR v1.0: 300 sec/seq, 64G RAM

VADR v1.1:
~2 sec/seq, 2G RAM typical seqs
~60 sec/seq, 64G RAM error-rich seqs

VADR 1.1: 150X speedup on typical sequences



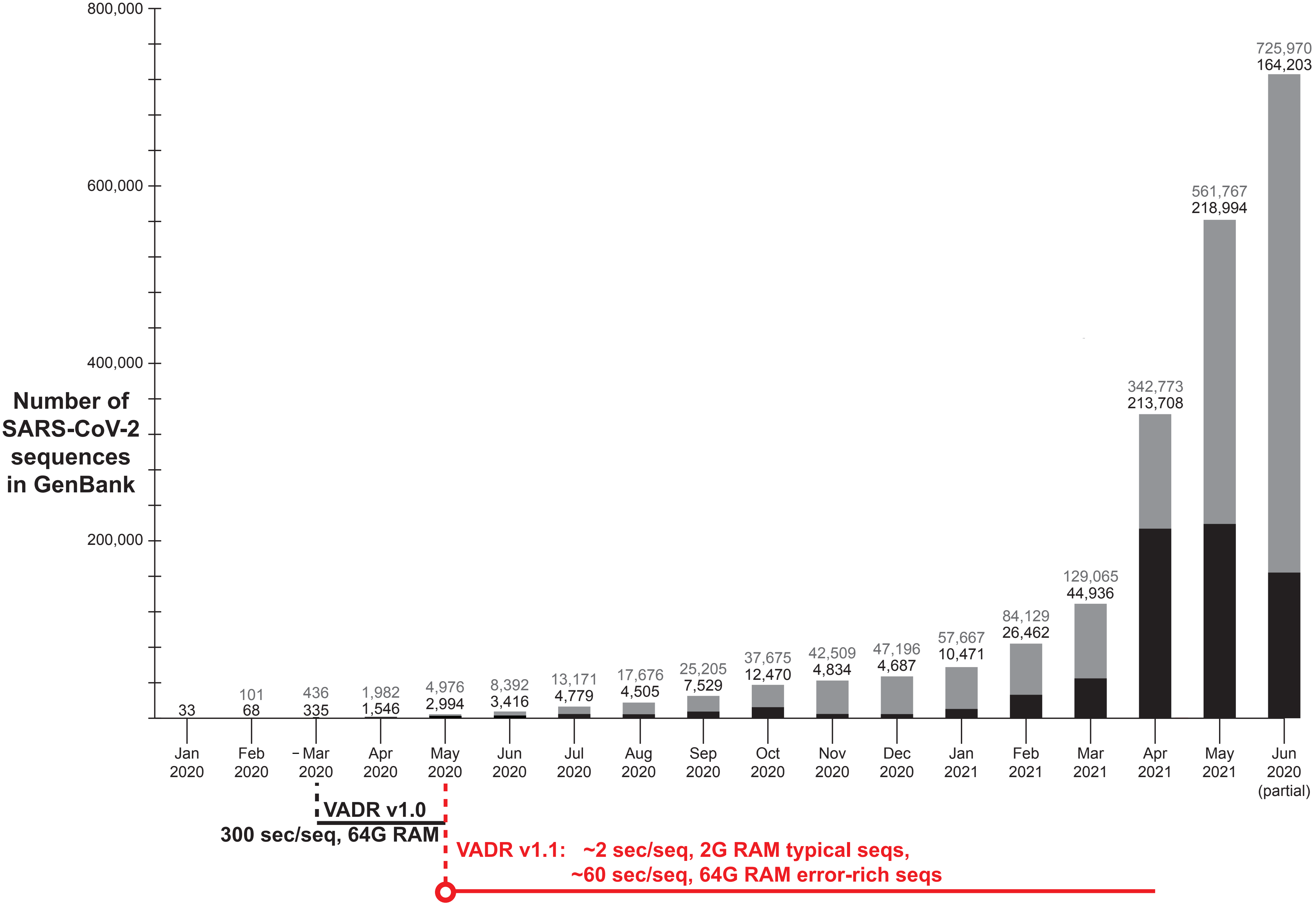
Time to process 5,000 sequences:

version	*1 host*	*20 hosts*
v1.0	~2.5 weeks	~1 day
v1.1	~4 hours	~15 min

VADR v1.0: 300 sec/seq, 64G RAM

VADR v1.1:
~2 sec/seq, 2G RAM typical seqs
~60 sec/seq, 64G RAM error-rich seqs

Sequence volume increased dramatically in 2021

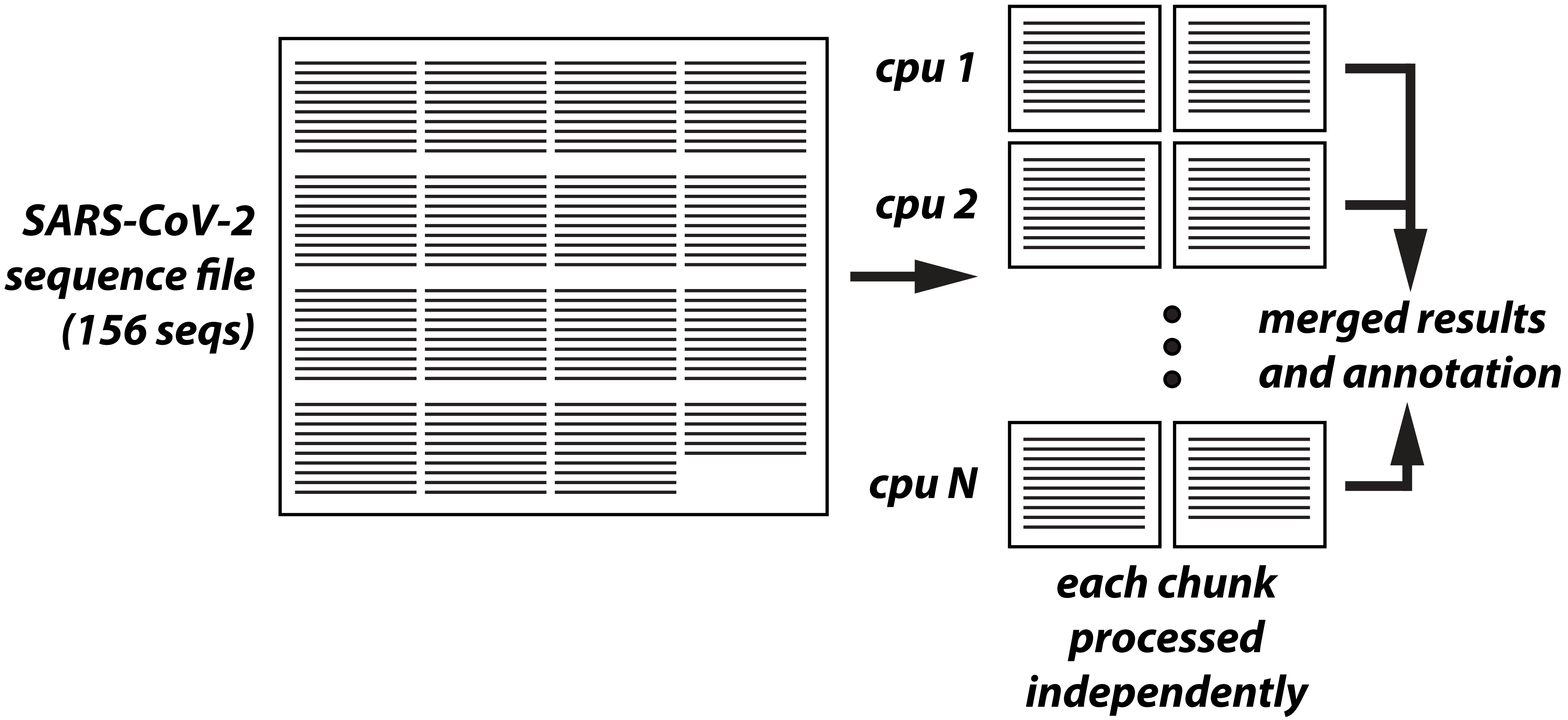


VADR v1.0
300 sec/seq, 64G RAM

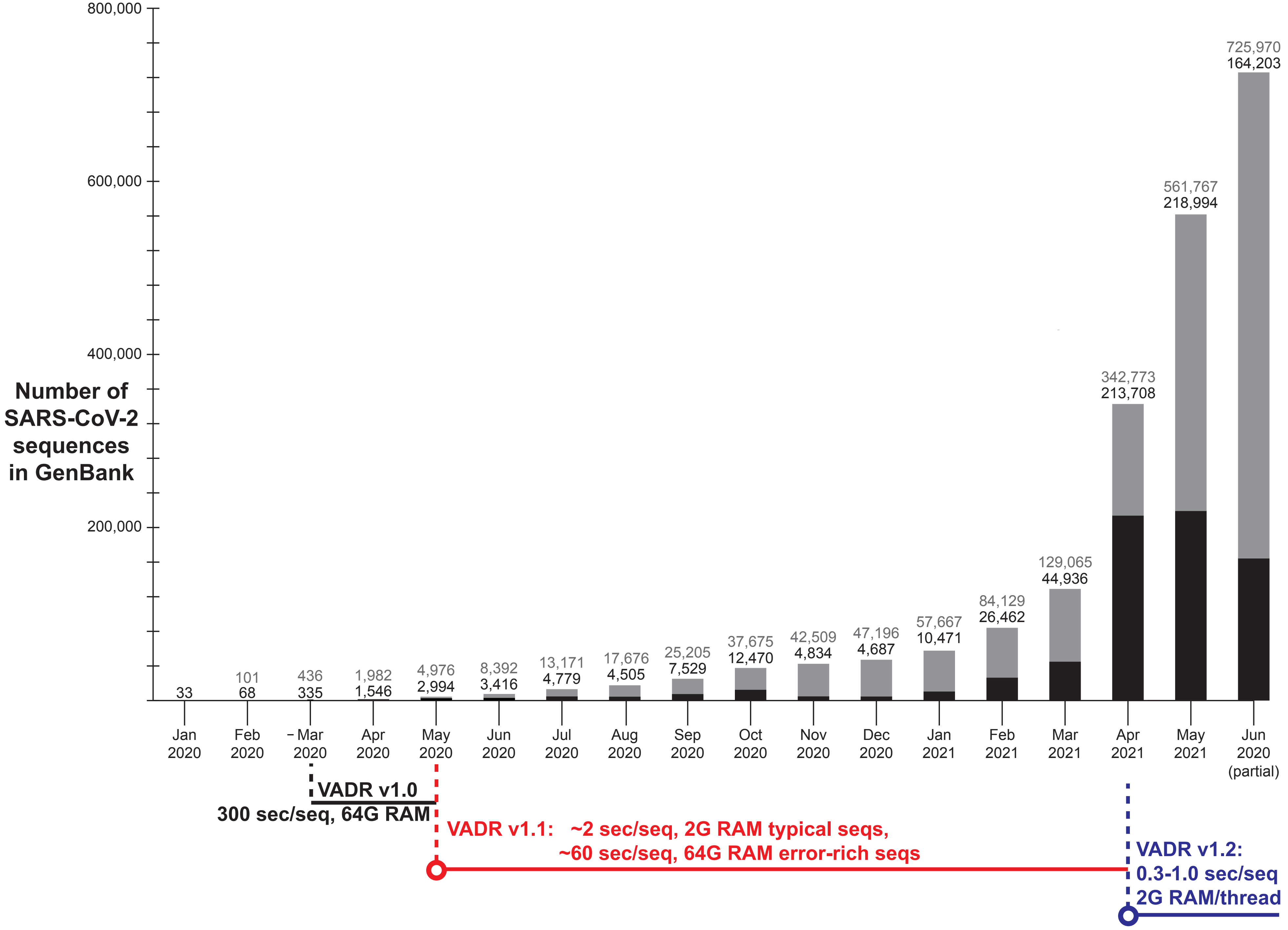
VADR v1.1: ~2 sec/seq, 2G RAM typical seqs,
~60 sec/seq, 64G RAM error-rich seqs

Speed and memory bottleneck in VADR 1.1 is cmalign

- VADR 1.2 replaces cmalign with glsearch ('glocal' alignment)
  - lower memory requirement (2G max) opens door for multi-threading



VADR v1.2 is about 10X faster than v1.1

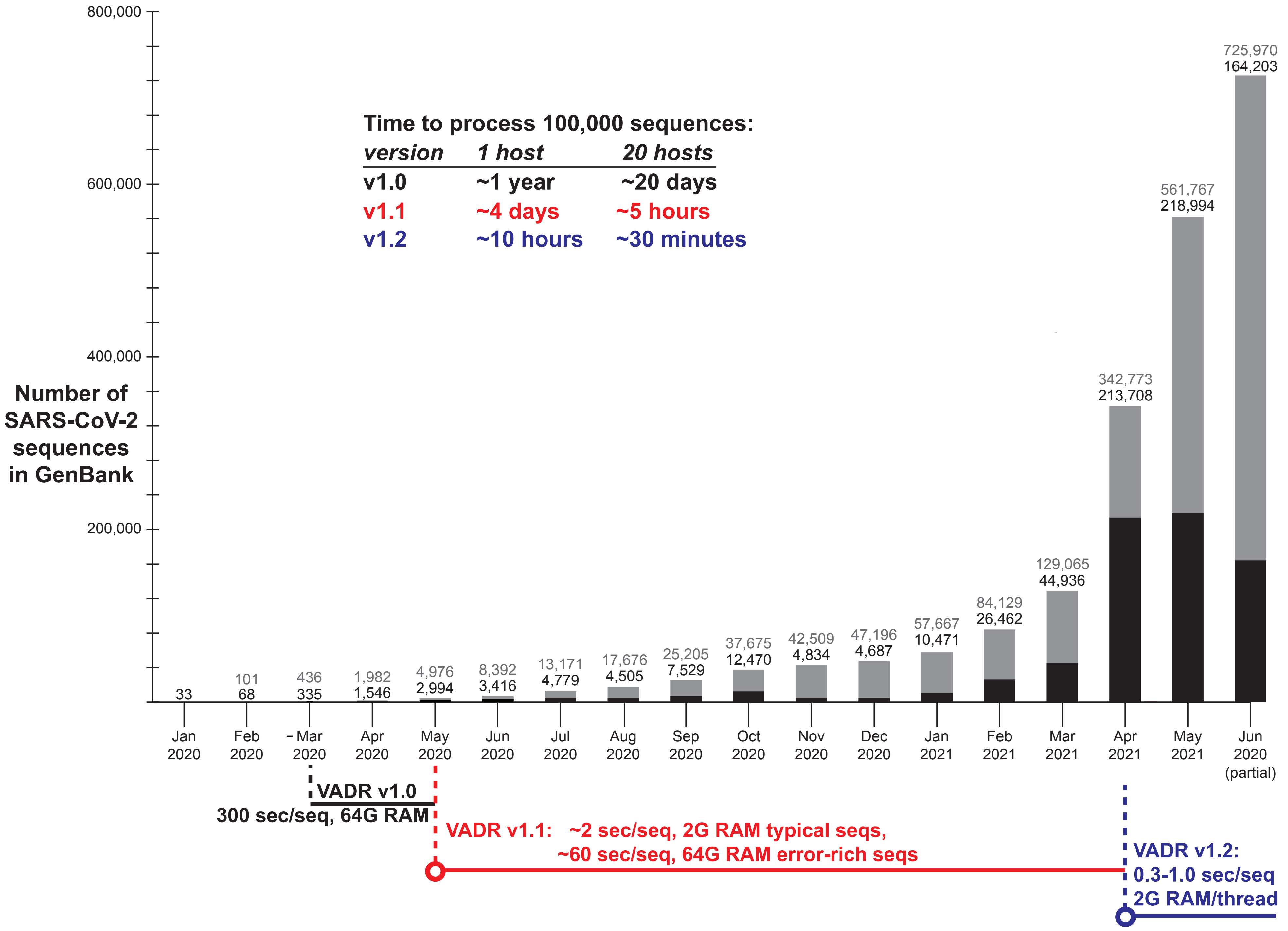


VADR v1.0
300 sec/seq, 64G RAM

VADR v1.1: ~2 sec/seq, 2G RAM typical seqs,
~60 sec/seq, 64G RAM error-rich seqs

VADR v1.2:
0.3-1.0 sec/seq
2G RAM/thread

GenBank is now better prepared for large sequence submissions



Time to process 100,000 sequences:

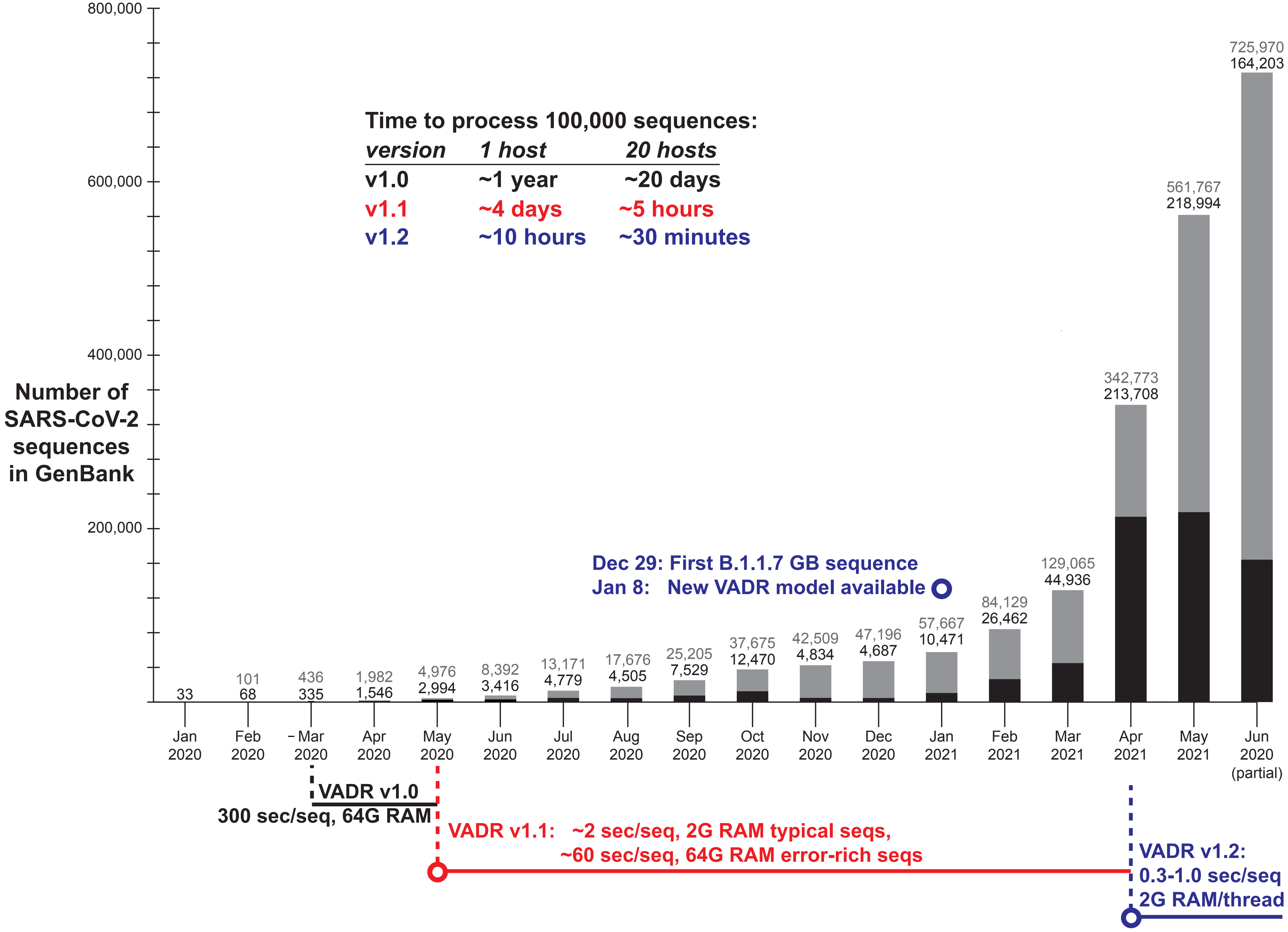
version	*1 host*	*20 hosts*
v1.0	~1 year	~20 days
v1.1	~4 days	~5 hours
v1.2	~10 hours	~30 minutes

VADR v1.0
300 sec/seq, 64G RAM

VADR v1.1: ~2 sec/seq, 2G RAM typical seqs,
~60 sec/seq, 64G RAM error-rich seqs

VADR v1.2:
0.3-1.0 sec/seq
2G RAM/thread

Besides getting faster, VADR has improved in other ways



Time to process 100,000 sequences:

version	*1 host*	*20 hosts*
v1.0	~1 year	~20 days
v1.1	~4 days	~5 hours
v1.2	~10 hours	~30 minutes

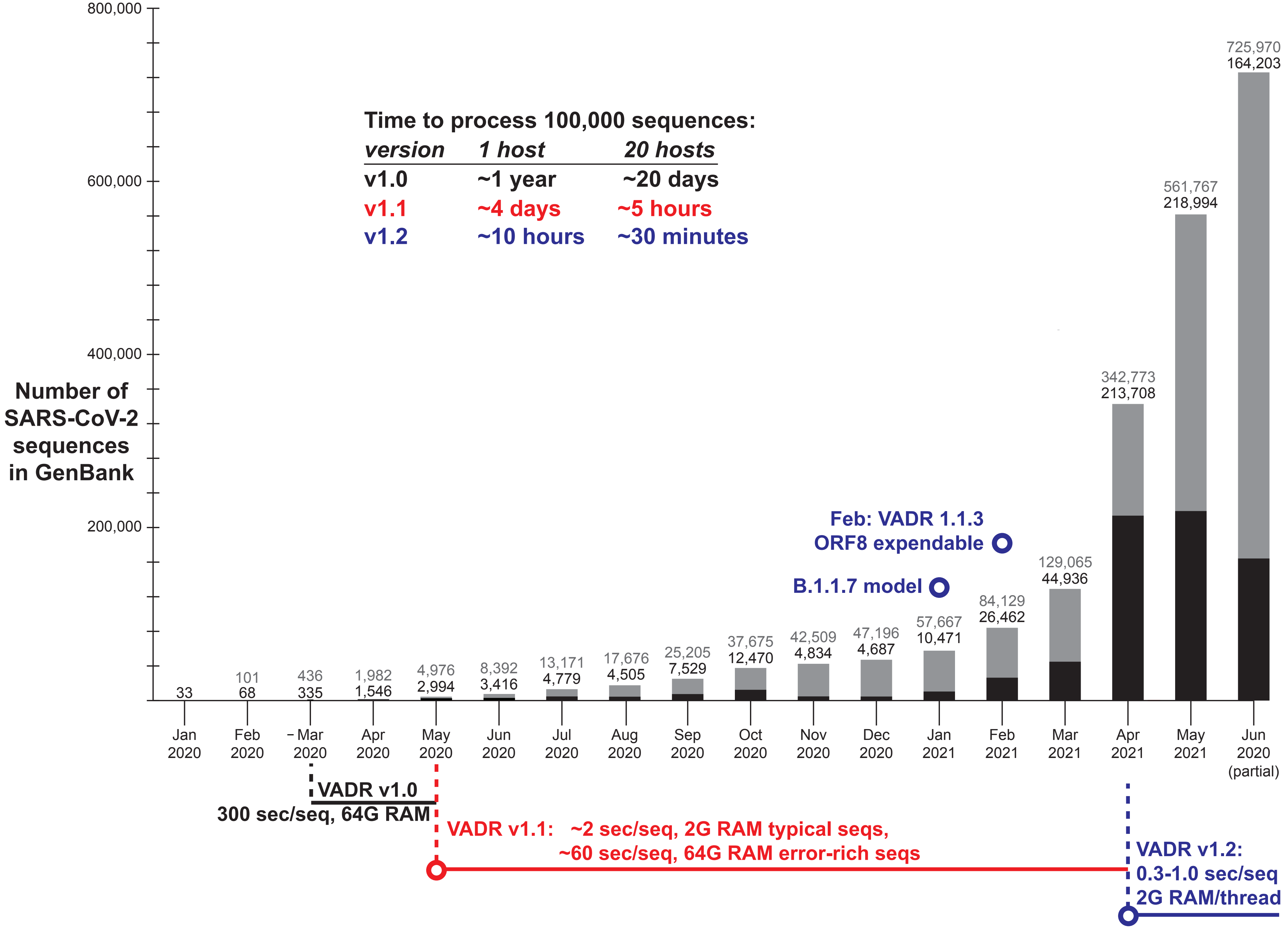
Dec 29: First B.1.1.7 GB sequence
Jan 8: New VADR model available

VADR v1.0
300 sec/seq, 64G RAM

VADR v1.1: ~2 sec/seq, 2G RAM typical seqs,
~60 sec/seq, 64G RAM error-rich seqs

VADR v1.2:
0.3-1.0 sec/seq
2G RAM/thread

Besides getting faster, VADR has improved in other ways



Time to process 100,000 sequences:

version	*1 host*	*20 hosts*
v1.0	~1 year	~20 days
v1.1	~4 days	~5 hours
v1.2	~10 hours	~30 minutes

Feb: VADR 1.1.3
ORF8 expendable

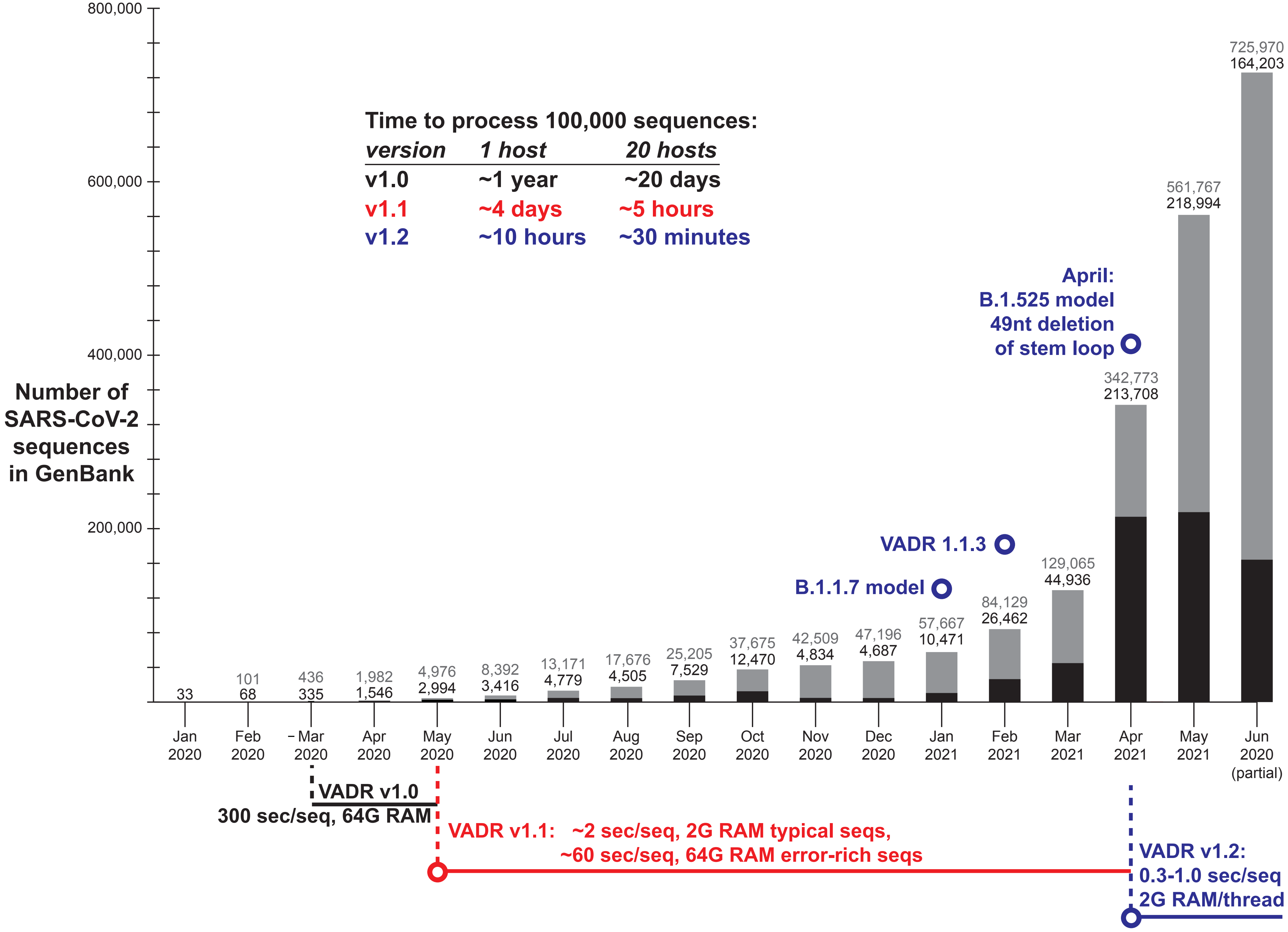
B.1.1.7 model

VADR v1.0
300 sec/seq, 64G RAM

VADR v1.1: ~2 sec/seq, 2G RAM typical seqs,
~60 sec/seq, 64G RAM error-rich seqs

VADR v1.2:
0.3-1.0 sec/seq
2G RAM/thread

Besides getting faster, VADR has improved in other ways



Time to process 100,000 sequences:

version	*1 host*	*20 hosts*
v1.0	~1 year	~20 days
v1.1	~4 days	~5 hours
v1.2	~10 hours	~30 minutes

April:
B.1.525 model
49nt deletion
of stem loop

VADR 1.1.3

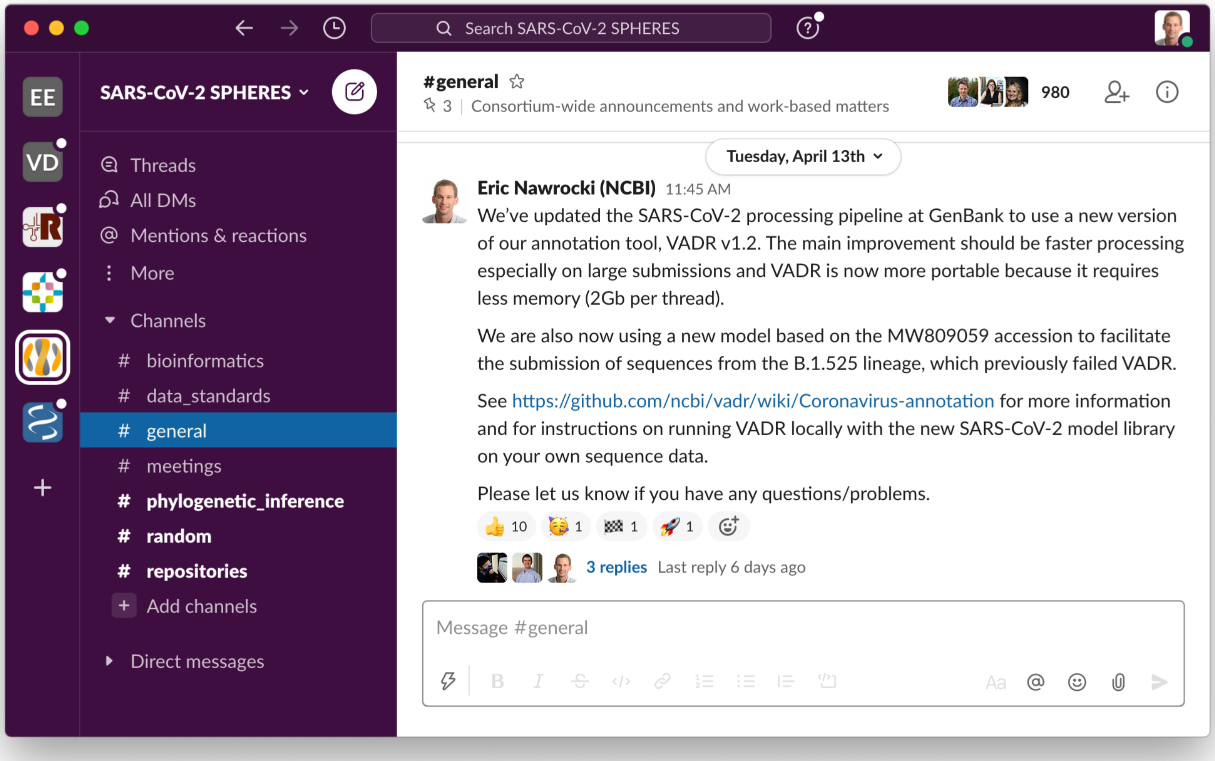
B.1.1.7 model

VADR v1.0
300 sec/seq, 64G RAM

VADR v1.1: ~2 sec/seq, 2G RAM typical seqs,
~60 sec/seq, 64G RAM error-rich seqs

VADR v1.2:
0.3-1.0 sec/seq
2G RAM/thread

We actively support (and are helped by) the SPHERES community



Eric Nawrocki (NCBI) 11:45 AM

We've updated the SARS-CoV-2 processing pipeline at GenBank to use a new version of our annotation tool, VADR v1.2. The main improvement should be faster processing especially on large submissions and VADR is now more portable because it requires less memory (2Gb per thread).

We are also now using a new model based on the MW809059 accession to facilitate the submission of sequences from the B.1.525 lineage, which previously failed VADR.

See https://github.com/ncbi/vadr/wiki/Coronavirus-annotation for more information and for instructions on running VADR locally with the new SARS-CoV-2 model library on your own sequence data.

Please let us know if you have any questions/problems.

- VADR is portable and is run locally by labs on their sequences prior to submission
- Docker container adds to portability (thanks to Anders Goncalves da Silva, Curtis Kapsak and StaPH-B!)
- SPHERES/CDC alert us of problems with VADR and model coverage

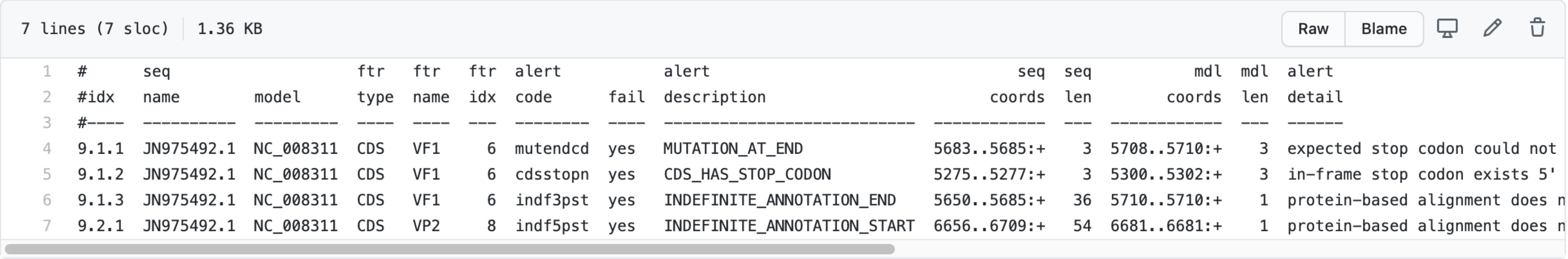
Future improvements: VADR 1.2.2 TODO list

- Reviewed sequences that fail VADR but should pass
  - allow problems in other non-essential genes (misc_featurization)
    - ORF3a
    - ORF6
    - ORF7a
    - ORF7b
    - ORF10
- Review VADR error messages, and add parseable position data (SPHERES)

Reference position data for alerts in VADR 1.2.2

- https://github.com/ncbi/vadr/blob/alert-info/documentation/formats.md#alt

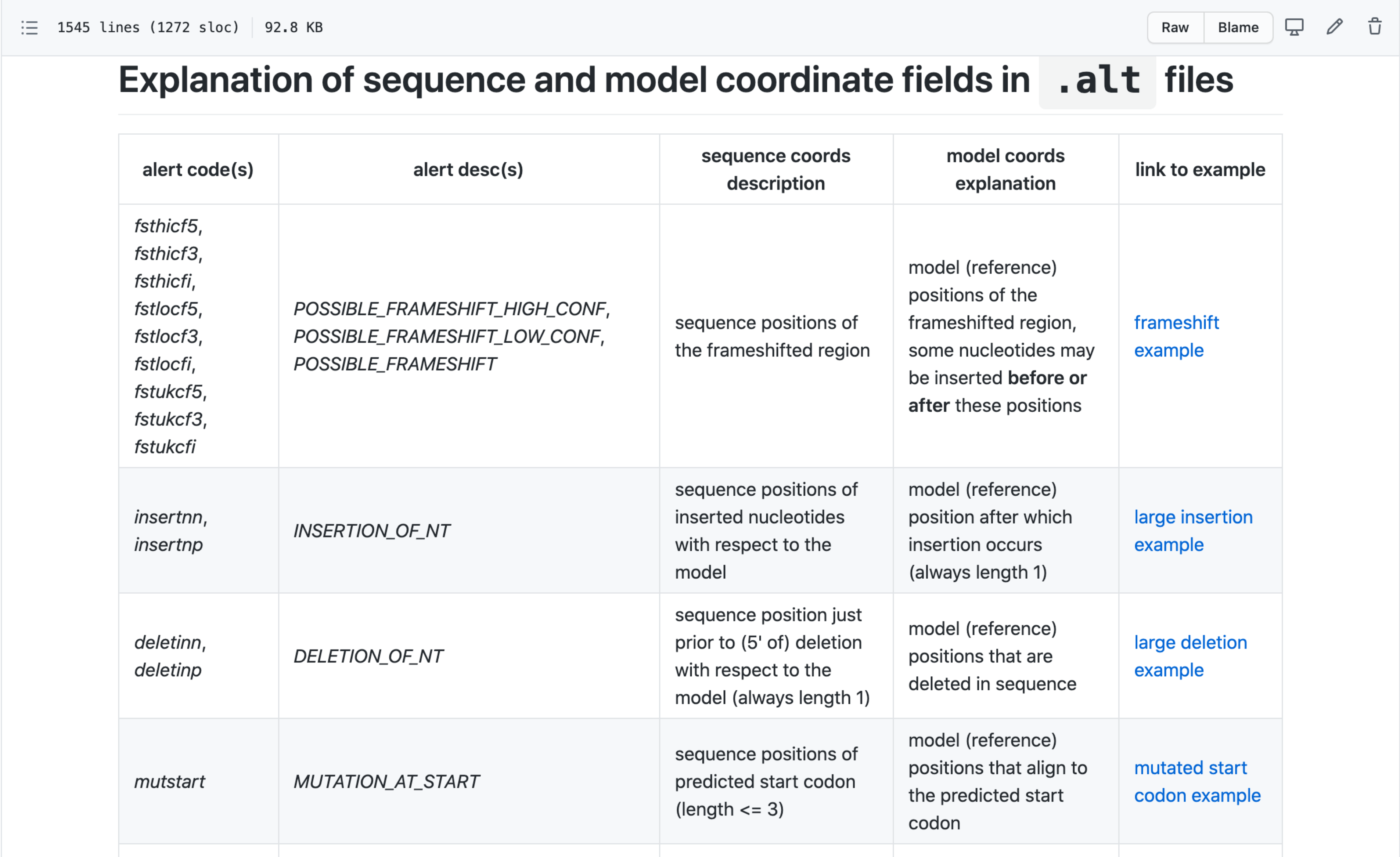
- https://github.com/ncbi/vadr/blob/alert-info/documentation/alerts.md



```
#      seq                    ftr   ftr   ftr  alert           alert                                 seq  seq           mdl  mdl  alert
#idx   name        model      type  name  idx  code      fail  description                        coords  len        coords  len  detail
#----  ----------  ---------  ----  ----  ---  --------  ----  ---------------------------  ------------  ---  ------------  ---  ------
9.1.1  JN975492.1  NC_008311  CDS   VF1     6  mutendcd  yes   MUTATION_AT_END              5683..5685:+    3  5708..5710:+    3  expected stop codon could not
9.1.2  JN975492.1  NC_008311  CDS   VF1     6  cdsstopn  yes   CDS_HAS_STOP_CODON           5275..5277:+    3  5300..5302:+    3  in-frame stop codon exists 5'
9.1.3  JN975492.1  NC_008311  CDS   VF1     6  indf3pst  yes   INDEFINITE_ANNOTATION_END    5650..5685:+   36  5710..5710:+    1  protein-based alignment does n
9.2.1  JN975492.1  NC_008311  CDS   VP2     8  indf5pst  yes   INDEFINITE_ANNOTATION_START  6656..6709:+   54  6681..6681:+    1  protein-based alignment does n
```

Reference position data for alerts in VADR 1.2.2

- https://github.com/ncbi/vadr/blob/alert-info/documentation/formats.md#alt
- https://github.com/ncbi/vadr/blob/alert-info/documentation/alerts.md



Explanation of sequence and model coordinate fields in `.alt` files

alert code(s)	alert desc(s)	sequence coords description	model coords explanation	link to example
fsthicf5, *fsthicf3*, *fsthicfi*, *fstlocf5*, *fstlocf3*, *fstlocfi*, *fstukcf5*, *fstukcf3*, *fstukcfi*	*POSSIBLE_FRAMESHIFT_HIGH_CONF*, *POSSIBLE_FRAMESHIFT_LOW_CONF*, *POSSIBLE_FRAMESHIFT*	sequence positions of the frameshifted region	model (reference) positions of the frameshifted region, some nucleotides may be inserted **before or after** these positions	frameshift example
insertnn, *insertnp*	*INSERTION_OF_NT*	sequence positions of inserted nucleotides with respect to the model	model (reference) position after which insertion occurs (always length 1)	large insertion example
deletinn, *deletinp*	*DELETION_OF_NT*	sequence position just prior to (5' of) deletion with respect to the model (always length 1)	model (reference) positions that are deleted in sequence	large deletion example
mutstart	*MUTATION_AT_START*	sequence positions of predicted start codon (length <= 3)	model (reference) positions that align to the predicted start codon	mutated start codon example

There are other viruses...

- VADR was designed to be general to other, short ($<$ 30Kb) non-segmented viruses
  - also used for Norovirus and Dengue virus
  - we'd like to expand to other flaviriruses and caliciviruses and beyond
  - small scale use for PRRSV and Herpes Simplex Virus 2 (HSV2, 150Kb)

There are other viruses...

- VADR was designed to be general to other, short ($<$ 30Kb) non-segmented viruses
  - also used for Norovirus and Dengue virus
  - we'd like to expand to other flaviriruses and caliciviruses and beyond
  - small scale use for PRRSV and Herpes Simplex Virus 2 (HSV2, 150Kb)
- VADR can also be used for other sequence elements:
  - COX1 sequences, a mitochondrial protein coding gene used for animal phylogenetics
  - may expand to other commonly submitted protein-coding genes

VADR documentation is on GitHub

- https://github.com/ncbi/vadr

- https://github.com/ncbi/vadr#readme

- https://github.com/ncbi/vadr/wiki/Coronavirus-annotation

- VADR paper:
  https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-0
  3537-3

Acknowledgements

NCBI - viral annotation
Alejandro Schäffer (now NCI)

Ilene Mizrachi
Colleen Bollin
Linda Yankie
Vincent Calhoun
Susan Schafer
Beverly Underwood
Vasuki Gobu
Sergiy Gotvyanskyy
Alex Kotliarov

Rodney Brister
Eneida Hatcher

Lara Shonkwiler
Sophia Hu

Wratko Hlavina
Eyal Mozes
Ron Patterson
Sumit Saluja

NCBI - leadership
David Landsman
Kim Pruitt
Steve Sherry
Jim Ostell
David Lipman

NLM - leadership
Patti Brennan
Jerry Sheehan
Valerie Florance

Software developers
Sean Eddy (HMMER/Infernal/Easel)
Travis Wheeler (HMMER)
Tom Madden and BLAST team
William Pearson (FASTA/glsearch)
Michael Farrar (HMMER/glsearch)

VADR docker image
Curtis Kapsak
Anders Goncalves da Silva

NIH NLM

NCBI National Center for Biotechnology Information