



## Supplementary Material for **Learning the language of viral evolution and escape**

Brian Hie, Ellen Zhong, Bonnie Berger\*, Bryan Bryson\*

\*Corresponding author. Email: bab@mit.edu (B.Be.); bryand@mit.edu (B.Br.)

Published 15 January 2021, *Science* **371**, 284 (2021)  
DOI: 10.1126/science.abd7331

**This PDF file includes:**

Materials and Methods  
Supplementary Text  
Figs. S1 to S3  
Tables S1 to S3  
References

**Other Supplementary Material for this manuscript includes the following:**  
(available at [science.sciencemag.org/content/371/6256/284/suppl/DC1](https://science.sciencemag.org/content/371/6256/284/suppl/DC1))

MDAR Reproducibility Checklist

## Materials and Methods

### Language model selection and training details

To select the model architecture described in **Note S2**, we performed a small-scale grid search using categorical cross entropy loss on the influenza dataset described below. We evaluated language model performance with a test set of held-out HA sequences where the first recorded date was before 1990 or after 2017, yielding a test set of 7,497 out of 44,999 sequences (about 17%). Hyperparameter search ranges were influenced by previous applications of recurrent architectures to protein sequence representation learning (9). We tested hidden unit dimensions of 128, 256, and 512. We tested architectures with one or two hidden layers. We tested three hidden-layer architectures: a densely connected neural network with access to both left and right sequence contexts, an LSTM with access to only the left context, and a BiLSTM with access to both left and right sequence contexts. We tested two Adam learning rates (0.01 and 0.001). All other architecture details (**Note S2**) were fixed to reasonable defaults. In total, we tested 36 conditions, and ultimately used a BiLSTM architecture with two hidden layers of 512 hidden units each, with an Adam learning rate of 0.001. We used the same architecture for all experiments. We train the language model to predict the observed amino acid residue at all positions in each sequence, using the remaining sequence as the input; one training epoch is completed when the model has considered all positions in all sequences in the training corpus. We trained each model until convergence of cross entropy loss across one training epoch.

### News headline data, model training, and CSCS

Preprocessed headlines (stripped of punctuation, space-delimited, and lower-cased) from the Australian Broadcasting Corporation (early-2013 through the end of 2019) were obtained from <https://www.kaggle.com/therohk/million-headlines>. We trained a word-level BiLSTM

language model. Semantic embeddings and grammaticality were quantified as described in the **Supplementary Text**. To obtain CSCS-proposed headline mutations, we considered all possible single-word mutations and acquired the top according to the CSCS objective. For comparison, we also acquired the single-word mutated headline with the closest embedding vector to the original headline as the “semantically closest” mutation.

#### Viral protein sequence datasets and model training

We trained three separate language models for influenza HA, HIV Env, and SARS-CoV-2 Spike using the model architecture described in **Note S2**. One training epoch consisted of predicting each token over all sequences in the training set.

Influenza HA amino acid sequences were downloaded from the “Protein Sequence Search” section of <https://www.fludb.org>. We only considered complete hemagglutinin sequences from virus type A. We trained an amino acid residue-level language model on a total of 44,851 unique influenza A hemagglutinin (HA) amino acid sequences observed in animal hosts from 1908 through 2019.

HIV Env protein sequences were downloaded from the “Sequence Search Interface” at the Los Alamos National Laboratory (LANL) HIV database (<https://www.hiv.lanl.gov>) (29). All complete HIV-1 Env sequences were downloaded from the database, excluding sequences that the database had labeled as “problematic.” We additionally only considered sequences that had length between 800 and 900 amino acid residues, inclusive. We trained an amino acid residue-level language model on a total of 57,730 unique Env sequences.

*Coronaviridae* spike glycoprotein sequences were obtained from the Gene/Protein Search portal of the ViPR database (<https://www.viprbrc.org/brc/home.spg?decorator=corona>) across the entire *Coronaviridae* family. We only included amino acid sequences with “spike” gene

products. SARS-CoV-2 Spike sequences were obtained from the Severe acute respiratory syndrome coronavirus 2 datahub at NCBI Virus (<https://www.ncbi.nlm.nih.gov/labs/virus/vssi/>). Betacoronavirus spike sequences from GISAID also used in Starr et al.'s analysis (20) were obtained from [https://github.com/jbloomlab/SARS-CoV-2-RBD\\_DMS/blob/master/data/alignments/Spike\\_GISAID/spike\\_GISAID\\_aligned.fasta](https://github.com/jbloomlab/SARS-CoV-2-RBD_DMS/blob/master/data/alignments/Spike_GISAID/spike_GISAID_aligned.fasta). Across all coronavirus datasets, we furthermore excluded sequences with a protein sequence length of less than 1,000 amino acid residues. We trained an amino acid residue-level language model on a total of 4,172 unique Spike (and homologous protein) sequences.

#### Semantic embedding landscape visualization, clustering, and quantification

We used the language models for HA, Env, and Spike to produce semantic embeddings for sequences within each language model's respective training corpus, where the semantic embedding procedure is described in the **Supplementary Text**. Using the Scanpy version 1.4.5 Python package (30), we first constructed the Euclidean  $k$ -nearest neighbors (KNN) graph where each node corresponds to an embedded viral sequence ( $k = 100$  for influenza and HIV and  $k = 20$  for coronavirus). Based on the KNN graph, we used the UMAP (13) Python implementation (<https://github.com/lmcinnes/umap>) as wrapped by Scanpy (<https://scanpy.readthedocs.io/>) with default parameters to construct the two-dimensional visualizations.

Also based on the same KNN graph, we performed unsupervised clustering with Louvain community detection (14) with a resolution parameter of 1, also using the implementation wrapped by Scanpy, to cluster sequences within each viral corpus. Louvain cluster purity was evaluated with respect to a metadata class (e.g., host species or subtype) by first calculating the percent composition of each metadata class label (e.g., "H1" through "H16" for HA subtype)

within a given cluster and using the maximum composition over all class labels as the purity percentage; we calculated this purity percentage for each Louvain cluster.

We compare Louvain clustering purities to those from phylogenetic-based clustering strategies. We obtained phylogenetic trees using five tools: guide trees from MAFFT version 7.453 (21), guide trees from Clustal Omega version 1.2.4 (31), maximum likelihood (ML) trees from MrBayes version 3.2.7a (32) with mixed amino acid rate matrices, ML trees from RAxML version 8.2.12 (33) with a PROTCATBLOSUM62 substitution model, and ML trees from FastTree version 2.1.11 SSE3 (34) with a JTT+CAT model; we also tried PhyML version 3.3.20200621 (35) and ClustalW2 version 2.1 (36), but these two failed to scale to the large number of sequences in the training corpuses and exceeded the 768 GB memory capacity of our benchmarking hardware. The ML tree methods were constructed using MAFFT-aligned sequences as input (note that this is separate from the MAFFT-generated guide tree).

We used each of the five methods to construct a phylogenetic tree of the respective viral sequence corpus. We then used TreeCluster (37) (<https://github.com/niemasd/TreeCluster>) to group sequences using either the “max clade” or the “single linkage” algorithm. We performed a range search of the TreeCluster threshold parameter to ensure that the number of returned clusters was equal to the number of clusters returned by the Louvain clustering of the same sequence corpus; all other parameters were set to the defaults. Using the cluster labels returned by TreeCluster, we computed cluster purities using the same procedure described for Louvain clustering above. Louvain comparison to MAFFT with max clade clustering is provided in **Figure 2** and comparisons to all other clustering strategies are provided in **figure S1D**.

#### Fitness validation

We obtained mutational fitness preference scores for HA H1 A/WSN/1933 (WSN33) mutants from Doud and Bloom (17), preference scores for antigenic site B mutants in HA H3 strains A/Hong Kong/1/1968 (HK68), A/Bangkok/1/1979 (Bk79), A/Beijing/353/1989 (Bei89), A/Moscow/10/1999 (Mos99), A/Brisbane/10/2007 (Bris07), and A/North Dakota/26/2016 (NDako16) from Wu et al. (18), preference scores for Env BF520 and BG505 mutants from Haddox et al. (19), and Kd binding affinities between yeast-displayed SARS-CoV-2 RBD mutants and ACE2 from Starr et al. (20). For the replication fitness DMS data, we used the preference scores averaged across technical replicates (as done within each study). For the ACE2 DMS data, if a sequence had more than one measured Kd, we took the median Kd as the representative fitness score for that sequence.

Using the corresponding pretrained viral fitness model, we computed the grammaticality score and semantic change (with respect to the original wildtype sequence) for each mutant sequence produced in each of the above studies; semantic change and grammaticality computation is described for single-residue mutants in **Note S2** and for combinatorial mutants in **Note S3**. We computed the Spearman correlation between fitness preference or binding Kd with either the grammaticality score or the semantic change.  $P$  values are computed by assuming  $t$ -distributed correlation coefficients with  $N - 2$  degrees of freedom, where  $N$  is the number of samples. We used the implementation provided in the scipy version 1.3.1 Python package (<https://www.scipy.org/>) to compute Spearman correlation coefficients and corresponding  $P$  values.

#### Escape prediction validation

We obtained experimentally validated causal escape mutations to HA H1 WSN33 from Doud et al. (1), HA H3 Perth09 from Lee et al. (2), Env BG505 from Dingens et al. (3), Spike

from Baum et al. (5). To obtain Spike RBD DMS escape mutants, as done in Greaney et al. (4), we used mutations that decrease sensitivity to neutralizing antibody as well as those that are neither deleterious to ACE2 binding nor deleterious to RBD expression, where the latter two conditions were measured via DMS by Starr et al. (20). We then made, *in silico*, all possible single-residue mutations to H1 WSN33, H3 Perth09, Env BG505, Spike for the Baum et al. dataset (5), and only the Spike RBD for the Greaney et al. dataset (4). For each of these mutations, we computed semantic change and grammaticality and combined these scores using the CSCS rank-based acquisition function as described in **Note S2**. For a given viral protein, the value of the CSCS acquisition function was used to rank all possible mutants.

To assess enrichment of acquired escape mutants, we constructed a curve that plotted the top  $n$  CSCS-acquired mutants on the x-axis and the corresponding number of these mutants that were also causal escape mutations on the y-axis; the area under this curve, normalized to the total possible area, resulted in our normalized AUC metric for evaluating escape enrichment. The AUC is normalized to be between 0 and 1, where a value of 0.5 indicates random guessing and higher values indicate greater enrichment.

We computed a permutation-based  $P$  value to assess the statistical significance of the enrichment of a given CSCS ranking. To do so, we constructed a null distribution by randomly sampling (without replacement) a subset of mutants as a “null escape” set, controlling for the number of mutants by ensuring that the null escape mutant set was the same size as the true escape mutant set, and recalculating the normalized AUC accordingly (essentially “permuting” the escape versus non-escape labels). We repeated this for 100,000 permutations. Bonferroni-corrected  $P$  values were considered statistically significant if they were below 0.05.

#### Escape prediction benchmarking

We wanted to benchmark our ability to predict escape, which is based on combining grammaticality and semantic change, based on previous methods that assess either viral fitness (which we conceptually link to grammaticality) or learn functional representations (which we conceptually link to semantics) alone.

For our first fitness model, we use MAFFT to obtain an MSA separately for each viral sequence corpus (the same corpuses used to train our language models). We then compute the mutational frequency independently at each residue position, with respect to the wildtype sequence. To ensure high quality sequence alignments, we also restrict this computation to aligned sequences with a limited number of gap characters relative to the wildtype (0 gap characters for the influenza and coronavirus proteins and 15 gap characters for HIV Env). The mutational frequency for each amino acid at each residue was used as the measure of viral fitness for escape acquisition (acquiring mutants with higher observed frequency).

For our second fitness model, we use the EVcouplings framework (8, 38) (<https://github.com/debbiemarkslab/EVcouplings>), which leverages HMMER software for sequence alignment (39) (<http://hmmer.org/>), to estimate the predicted fitness using both the independent and epistatic models. We train the EVcouplings models using the same sequence corpuses that we used for training our language models. We acquired mutants based on higher predicted fitness scores obtained from the independent or the epistatic model. The epistatic model incorporates pairwise residue information by learning a probabilistic model in which each residue position corresponds to a random variable over an amino acid alphabet and pairwise information potentials can encode epistatic relationships.

We use a number of pretrained protein sequence embedding models to assess the ability for generic protein embedding models to capture antigenic information. We use the pretrained



soft symmetric alignment model with multitask structural training information from Bepler and Berger (9) (<https://github.com/tbepler/protein-sequence-embedding-iclr2019>). We also use the TAPE pretrained transformer model (10) and the UniRep pretrained model (11), both from the TAPE repository (<https://github.com/songlab-cal/tape>). Using each pretrained model, we computed an embedding for the wildtype sequence and for each single-residue mutant. We used the  $\ell_1$  distance between the wildtype and mutant embeddings as the semantic change score, as in **Note S2**. We acquired mutants favoring higher semantic change.

#### Combinatorial mutant re-infection analysis

We obtained the Spike sequences from the reported first and second rounds of SARS-CoV-2 infection of a single patient from To et al. (23). We computed the re-infection sequence's grammaticality as the average log language model probability across the individual mutant positions and the semantic change (relative to the first infection sequence) as the  $\ell_1$  distance between the original and mutant language model embeddings. The re-infection Spike sequence has four mutated positions relative to the first infection sequence. We note that the re-infection sequence was not present in the training corpus. We compared the predicted semantic change and grammaticality of the re-infection sequence to those of 891 unique SARS-CoV-2 Spike sequences from our training corpus, where semantic change was similarly defined with respect to the first infection sequence from To et al. (23). Additionally, we compared the re-infection sequence to a null distribution of 100 million sequences with four mutations compared to the first infection sequence. The mutations were chosen uniformly at random across each position and across the amino acid alphabet. A sequence from the null distribution was considered to have higher escape potential than the re-infection sequence if it had both higher fitness and higher semantic change.

As positive controls, we performed the same analysis on sequences in which SARS-CoV-2 Spike RBD was artificially replaced *in silico* with the RBD-ACE2 contacts of bat coronavirus RaTG13 (eight mutated positions relative to wildtype) or of SARS-CoV-1 (twelve mutated positions relative to wildtype), creating antigenically dissimilar sequences while preserving ACE2 binding, albeit with lower affinity (20). We note these *in silico* “recombinant” sequences are also not present in the training corpus. We again compared the semantic change and grammaticality of these recombinant sequences to the 891 surveilled Spike sequences in our training corpus (**fig. S2B**), as described in the previous paragraph; here, semantic change was defined relative to the wildtype Spike sequence.

#### Protein structure preprocessing and visualization

We calculated the escape potential at each position within a given viral sequence by summing the value of the CSCS rank-based acquisition function (i.e.,  $a'(\tilde{x}_i; \mathbf{x})$  in **Note S2**) across all amino acids. We then mapped these scores from our protein sequences of interest (used in the escape prediction validation experiments) to three-dimensional structural loci. As in Doud et al. (1), we mapped the positions from WSN33 to the structure of HA H1 A/Puerto Rico/8/1934 (PDB: 1RVX) (40). As in Lee et al. (2), we mapped the positions from Perth09 to the structure of HA H3 A/Victoria/361/2011 (PDB: 4O5N) (41). As in Dingens et al. (3), we used the structure of BG505 SOSIP (PDB: 5FYI) (42). We used the structure of closed-state Spike (PDB: 6VXX) (43). Escape potential across each of the structures was colored using a custom generated PyMOL script deposited to Zenodo (27) as `brianhie-viral-mutation-ff6765f/bin/color_protein.py`. Using this script, the protein was visualized with PyMOL version 2.3.3 (<https://pymol.org/2/>).

#### Protein structure regional enrichment and depletion quantification

We quantified the enrichment or depletion of escape prediction scores within a given region of a protein sequence. We define a region as a (potentially non-contiguous) set of positions; regions of each viral protein that we considered are provided in **table S3**. Head and stalk regions for HA were determined based on the coordinates used by Kirkpatrick et al. (24). Region positions for Env were determined using the annotation provided by UniProt (ID: QN0S5) and hypervariable loops were determined as defined by the HIV LANL database ([https://www.hiv.lanl.gov/content/sequence/VAR\\_REG\\_CHAR/variable\\_region\\_characterization\\_explanation.html](https://www.hiv.lanl.gov/content/sequence/VAR_REG_CHAR/variable_region_characterization_explanation.html)). Region positions for SARS-CoV-2 were determined using the annotation provided by UniProt (ID: P0DTC2).

To assess statistical significance, we construct a null distribution using a permutation-based procedure. We “permute” the labels corresponding to the region of interest by randomly selecting (without replacement) a set of positions that has an equal size as the region of interest; we then compute the average escape potential over this randomly selected set of positions, repeating for 100,000 permutations. We compute a  $P$  value by determining the average escape potential for the true region of interest and comparing it to the null distribution, where we test for both enrichment or depletion of escape potential. Enrichment or depletion is considered statistically significant if its Bonferroni-corrected  $P$  value is less than 0.05.

#### Computational resources and hardware

Models were trained and evaluated with tensorflow 2.2.0 and Python 3.7 on Ubuntu 18.04, with access to a Nvidia Tesla V100 PCIe GPU (32 GB RAM) and an Intel Xeon Gold 6130 CPU (2.10GHz, 768 GB of RAM). Using CUDA-based GPU acceleration, training on the influenza HA corpus required approximately 72 hours (all times are wall time) and evaluating all possible single-residue mutant sequences for a single strain required approximately 35 minutes.

Training on the HIV Env corpus required approximately 80 hours and evaluating all possible single-residue mutant sequences required approximately 90 minutes. Training on the coronavirus spike corpus required approximately 20 hours and evaluating all possible single-residue mutant sequences required approximately 10 hours.

## Supplementary Text

### Note S1: Problem Formulation

Intuitively, our goal is to identify mutations that induce high semantic change (e.g., a large impact on biological function) while being grammatically acceptable (e.g, biologically viable). More precisely, we are given a sequence of tokens defined as  $\mathbf{x} \stackrel{\text{def}}{=} (x_1, \dots, x_N)$  such that  $x_i \in \mathcal{X}, i \in [N]$ , where  $\mathcal{X}$  is a finite alphabet (e.g., characters or words for natural language, or amino acids for protein sequence). Let  $\tilde{x}_i$  denote a mutation at position  $i$  and the mutated sequence as  $\mathbf{x}[\tilde{x}_i] \stackrel{\text{def}}{=} (\dots, x_{i-1}, \tilde{x}_i, x_{i+1}, \dots)$ .

We first require a semantic embedding  $\mathbf{z} \stackrel{\text{def}}{=} f_s(\mathbf{x})$ , where  $f_s : \mathcal{X}^N \rightarrow \mathbb{R}^K$  embeds discrete-alphabet sequences into a  $K$ -dimensional continuous space, where, ideally, closeness in embedding space would correspond to semantic similarity (e.g., more similar in meaning). We denote semantic change as the distance in embedding space, i.e.,

$$\Delta \mathbf{z}[\tilde{x}_i] \stackrel{\text{def}}{=} \|\mathbf{z} - \mathbf{z}[\tilde{x}_i]\| = \|f_s(\mathbf{x}) - f_s(\mathbf{x}[\tilde{x}_i])\| \quad (1)$$

where  $\|\cdot\|$  denotes a vector norm. The grammaticality of a mutation is described by

$$p(\tilde{x}_i | \mathbf{x}) \quad (2)$$

which takes values close to zero if  $\mathbf{x}[\tilde{x}_i]$  is not grammatical and close to one if it is grammatical.

A mutation is considered grammatical if it conforms to the rules (e.g., morphology and syntax) within a given language; violation of these rules results in a loss of grammaticality.

Our objective combines semantic change and grammaticality. Taking inspiration from upper confidence bound acquisition functions in Bayesian optimization (44), we can combine terms (1) and (2) with a weight parameter  $\beta \in [0, \infty)$  above to compute

$$a(\tilde{x}_i; \mathbf{x}) \stackrel{\text{def}}{=} \Delta \mathbf{z}[\tilde{x}_i] + \beta p(\tilde{x}_i | \mathbf{x})$$

for each possible mutation  $\tilde{x}_i$ . Mutations  $\tilde{x}_i$  are prioritized based on  $a(\tilde{x}_i; \mathbf{x})$ ; we refer to this ranking of mutations based on semantic change and grammaticality as CSCS.

## Note S2: Algorithms

Algorithms for CSCS could potentially take many forms; for example, separate algorithms could be used to compute  $\Delta \mathbf{z}[\tilde{x}_i]$  and  $p(\tilde{x}_i | \mathbf{x})$  independently, or a two-step approach might be possible that computes one of the terms based on the value of the other.

Instead, we reasoned that a single approach could compute both terms simultaneously, based on learned language models that learn the probability distribution of a word given its context (6, 7, 45–47). The language model we use throughout our experiments considers the full sequence context of a word and learns a latent variable probability distribution  $\hat{p}$  and function  $\hat{f}_s$  over all  $i \in [N]$  where

$$\hat{p}(x_i | \mathbf{x}_{[N] \setminus \{i\}}, \hat{\mathbf{z}}_i) = \hat{p}(x_i | \hat{\mathbf{z}}_i) \quad \text{and} \quad \hat{\mathbf{z}}_i = \hat{f}_s(\mathbf{x}_{[N] \setminus \{i\}}),$$

i.e., latent variable  $\hat{\mathbf{z}}_i$  encodes the sequence context  $\mathbf{x}_{[N] \setminus \{i\}} \stackrel{\text{def}}{=} (\dots, x_{i-1}, x_{i+1}, \dots)$  such that  $x_i$  is conditionally independent of its context given the value of  $\hat{\mathbf{z}}_i$ .

We use different aspects of the language model to describe semantic change and grammaticality by setting terms (1) and (2) as

$$\Delta \mathbf{z}[\tilde{x}_i] \stackrel{\text{def}}{=} \|\hat{\mathbf{z}} - \hat{\mathbf{z}}[\tilde{x}_i]\|_1 \quad \text{and} \quad p(\tilde{x}_i | \mathbf{x}) \stackrel{\text{def}}{=} \hat{p}(\tilde{x}_i | \hat{\mathbf{z}}_i)$$

where  $\hat{\mathbf{z}} \stackrel{\text{def}}{=} \frac{1}{N} \sum_{i=1}^N \hat{\mathbf{z}}_i$  is the average embedding across all positions,  $\hat{\mathbf{z}}[\tilde{x}_i]$  is defined similarly but for the mutated sequence, and  $\|\cdot\|_1$  is the  $\ell_1$  norm, chosen because of more favorable properties compared to other standard distance metrics (48), though other metrics could be empirically quantified in future work.

Effectively, distances in embedding space approximate semantic change and the emitted probability approximates grammaticality. We call the emitted probability “grammaticality” because in natural language tasks, it tends to be high for grammatically correct sentences. In the case of viral sequences, the training distribution consists of viral proteins that have evolved for

high fitness/virality, so we hypothesize that high grammaticality corresponds to high viral fitness. We note that these modeling assumptions are not guaranteed to be perfectly specified, since, in the natural language setting for example, the language model output can also encode linguistic pragmatics in addition to grammaticality and antonyms may also be close in embedding space. However, we still find these modeling assumptions to have good empirical support.

Based on the success of recurrent architectures for protein-sequence representation learning (9–11), we use similar encoder models for viral protein sequences (**Fig. 1B**). Our model passes the full context sequence into BiLSTM hidden layers. We used the concatenated output of the final LSTM layers as the semantic embedding, i.e.,

$$\hat{\mathbf{z}}_i \stackrel{\text{def}}{=} \left[ \text{LSTM}_f \left( g_f(x_1, \dots, x_{i-1}) \right)^T \quad \dots \quad \text{LSTM}_r \left( g_r(x_{i+1}, \dots, x_N) \right)^T \right]^T$$

Where  $g_f$  is the output of the preceding forward-directed layer,  $\text{LSTM}_f$  is the final forward-directed LSTM layer, and  $g_r$  and  $\text{LSTM}_r$  are the corresponding reverse-directed components. The final output probability is a softmax-transformed linear transformation of  $\hat{\mathbf{z}}_i$ , i.e.,

$$\hat{p}(x_i | \hat{\mathbf{z}}_i) \stackrel{\text{def}}{=} \text{softmax}(\mathbf{W}\hat{\mathbf{z}}_i + \mathbf{b})$$

for some learned model parameters  $\mathbf{W}$  and  $\mathbf{b}$ . In our experiments, we used a 20-dimensional learned dense embedding for each element in the alphabet  $\mathcal{X}$ , two BiLSTM layers with 512 units, and categorical cross entropy loss optimized by Adam with a learning rate of 0.001,  $\beta_1 = 0.9$ , and  $\beta_2 = 0.999$ . Hyperparameters and architecture were selected based on a small-scale grid search as described in **Materials and Methods**.

Rather than acquiring mutations based on raw semantic change and grammaticality values, which may be on very different scales, we find that calibrating  $\beta$  is much easier in

practice when first rank-transforming the semantic change and grammaticality terms, i.e., acquiring based on

$$a'(\tilde{x}_i; \mathbf{x}) \stackrel{\text{def}}{=} \text{rank}(\Delta \mathbf{z}[\tilde{x}_i]) + \beta \text{rank}(p(\tilde{x}_i | \mathbf{x}))$$

All possible mutations  $\tilde{x}_i$  are then given priority based on the corresponding values of  $a'(\tilde{x}_i; \mathbf{x})$ , from highest to lowest. Our empirical results seem consistently well-calibrated around  $\beta = 1$  (equally weighting both terms), which we used in all of our experiments.

Note S3: Extension to combinatorial mutations

The above exposition is limited to the setting in which mutations are assumed to be single-token. We perform a simple extension to handle combinatorial mutations. We denote such a mutant sequence as  $\tilde{\mathbf{x}} = (\tilde{x}_1, \dots, \tilde{x}_N)$ , which has the same length as  $\mathbf{x}$ , where the set of mutations consists of the tokens in  $\tilde{\mathbf{x}}$  that disagree with those at the same position in  $\mathbf{x}$ , which we denote

$$\mathcal{M}(\mathbf{x}, \tilde{\mathbf{x}}) \stackrel{\text{def}}{=} \{\tilde{x}_i \mid \tilde{x}_i \neq x_i\}.$$

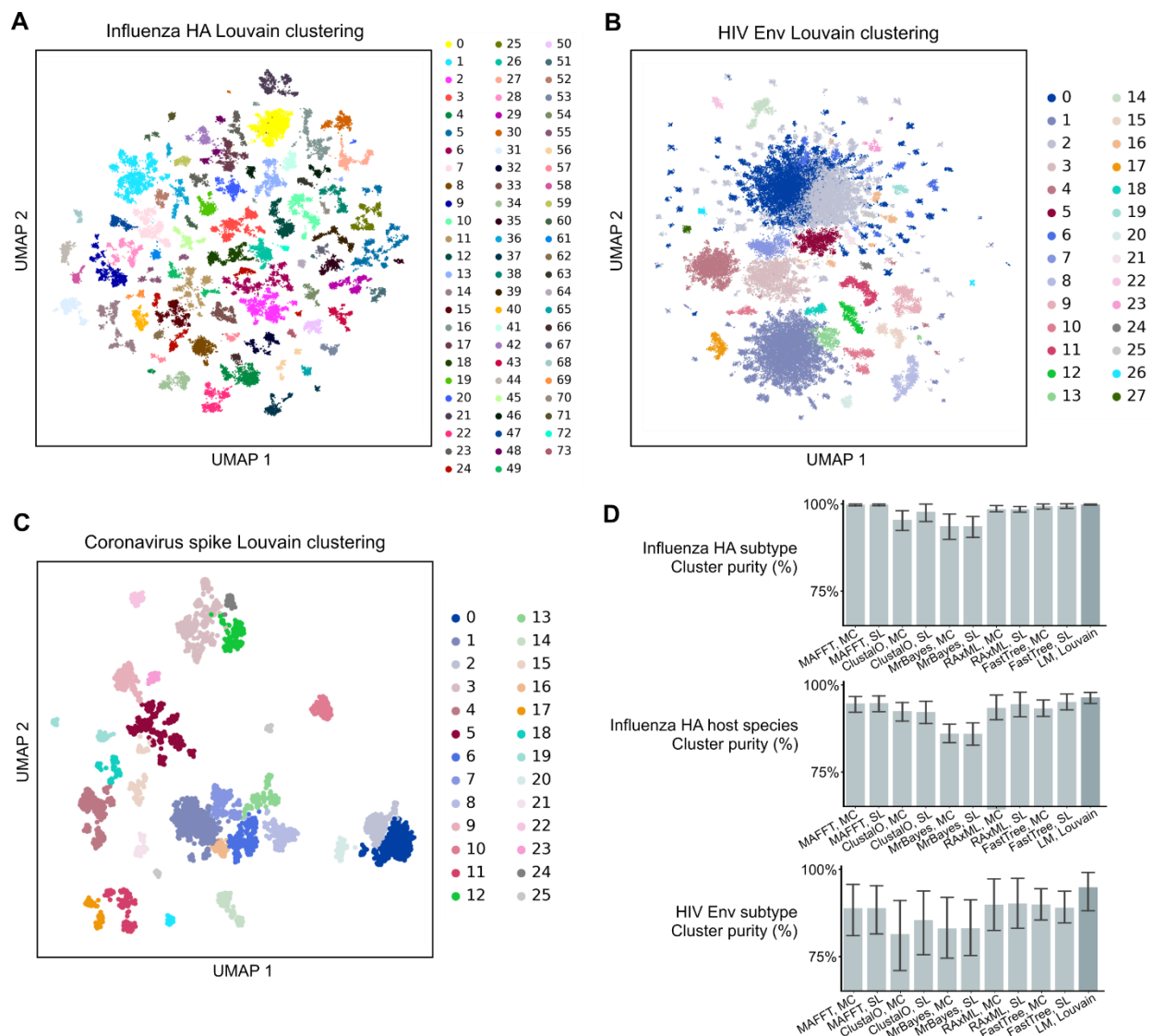
The semantic embedding can simply be computed as  $f_s(\tilde{\mathbf{x}})$  from which semantic change can be computed as above. For the grammaticality score, we make a simple modeling assumption and compute grammaticality as

$$\prod_{\tilde{x}_i \in \mathcal{M}(\mathbf{x}, \tilde{\mathbf{x}})} p(\tilde{x}_i | \mathbf{x}),$$

i.e., the product of the probabilities of the individual point-mutations (implemented in the log domain for better numerical precision). We note that this works well empirically in the combinatorial fitness datasets that we test, even when the number of mutations is not fixed as in the SARS-CoV-2 DMS Kd dataset. Other ways of estimating joint, combinatorial grammaticality terms while preserving efficient inference are also worth considering in future work.



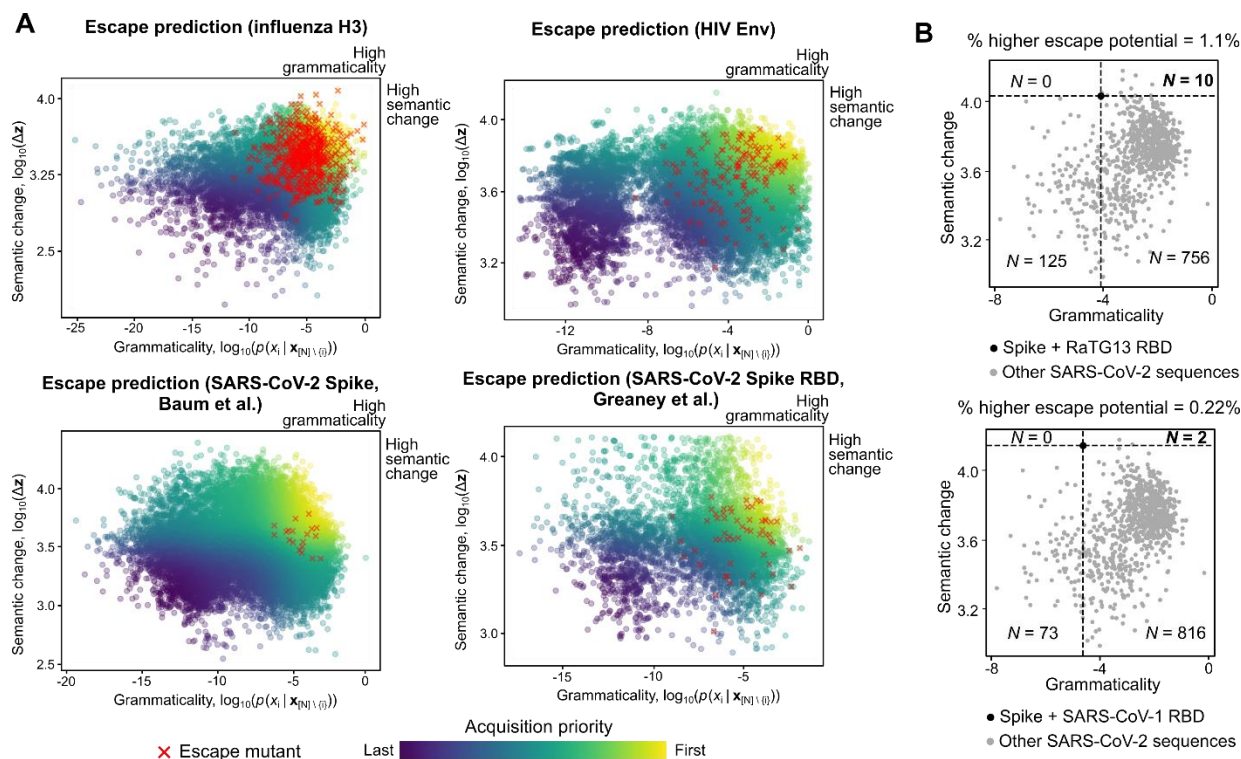
While we do not consider insertions or deletions in this study, we do note that, in viral sequences, insertions and deletions are rarer than substitutions by a factor of four or more (49) and the viral mutation datasets that we considered exclusively profiled substitution mutations alone. Extending our algorithms to compute semantic change of sequences with insertions or deletions would be essentially unchanged from above. The more difficult task is in reasoning about and modeling the grammaticality of an insertion or a deletion. While various grammaticality heuristics based on the language model output may be possible, this is also an interesting area for further methodological development.



**Fig. S1. Visualization of semantic landscape Louvain clustering**

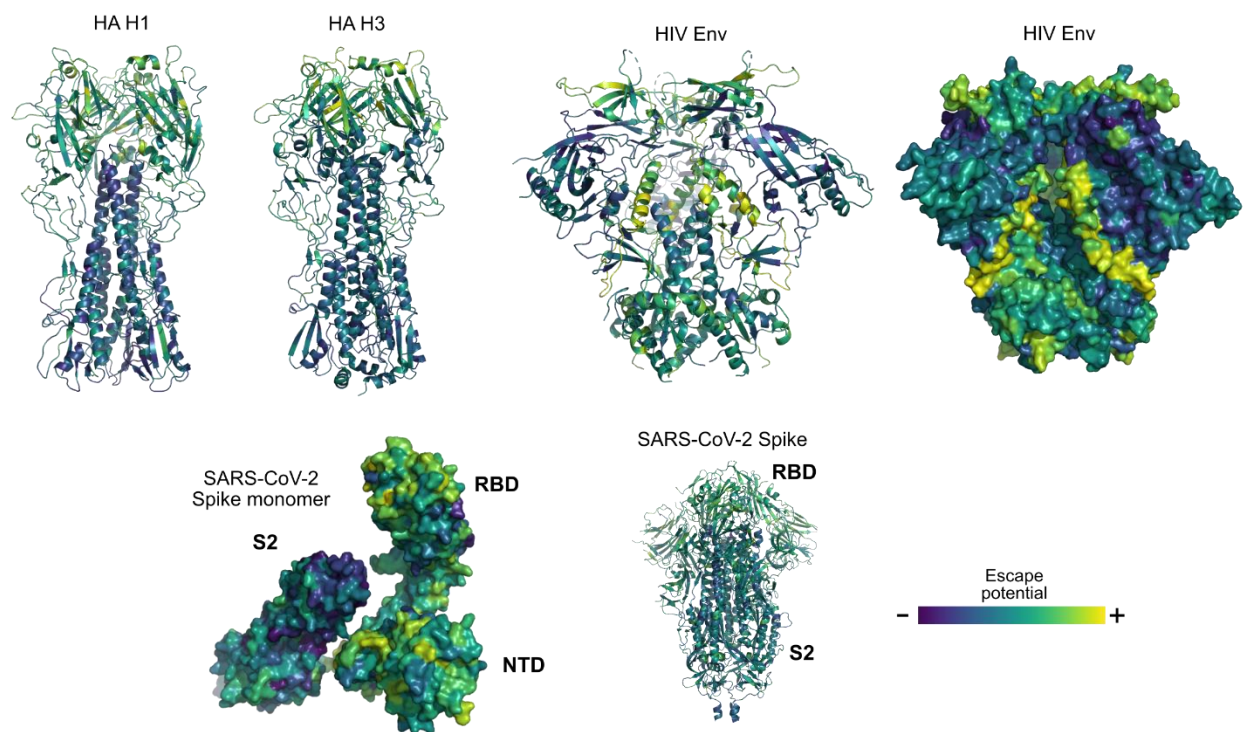
(A-C) Louvain cluster labels, used to evaluate cluster purity of HA subtype, HA host species, and HIV subtype, are visualized with the same UMAP coordinates as in **Figure 2**. Part of HA cluster 30 was highlighted in **Figure 2C**. Coronavirus Louvain clusters 0 and 2 were highlighted in **Figure 2G**. (D) Cluster purities of Louvain clustering on language model (LM) semantic embeddings were compared to those of clustering with either the max clade (MC) or single linkage (SL) algorithms applied to phylogenetic trees constructed by MAFFT, Clustal Omega (ClustalO), MrBayes, RAxML, or FastTree. Results for Louvain compared to MAFFT with MC

clustering are also shown in **Figure 2**. Bar height: mean; error bars: 95% confidence over  $N = 74$  clusters for HA and  $N = 28$  clusters for Env.



**Fig. S2. Semantic change and grammaticality of single and combinatorial mutations**

(A) Each point in the scatter plot corresponds to a single-residue mutation of the indicated viral protein or protein domain. Points are colored by CSCS acquisition priority (**Note S2**) and a red X is additionally drawn over the points that correspond to escape mutations. (B) Across 891 unique, surveilled SARS-CoV-2 Spike sequences, ten (1.1%) have higher semantic change and grammaticality compared to a Spike sequence modified to have RaTG13 RBD-ACE2 contact residues and two (0.22%) have higher semantic change and grammaticality compared to a Spike sequence with a SARS-CoV-1 RBD-ACE2 contact residues.



**Fig. S3. Additional protein structure visualizations**

Cartoon illustration of HA H1 and HA H3; view of HIV Env as cartoon and surface oriented to illustrate the semantically important inner domain; and views of SARS-CoV-2 Spike in monomeric (surface) and trimeric form (cartoon) illustrating S2 escape depletion.

<b>Model</b>	<b>WSN33</b>	<b>Bei89</b>	<b>Bk79</b>	<b>Bris07L194</b>	<b>HK68</b>	<b>Mos99</b>	<b>NDako16</b>	<b>BF520</b>	<b>BG505</b>	<b>Spike</b>
Semantic change (Spearman r)	-0.1175	-0.1653	-0.4040	-0.0051	-0.2549	-0.2097	-0.0711	-0.1024	-0.1101	-0.4421
Grammaticality (Spearman r)	0.2789	0.1876	0.4274	0.4021	0.5396	0.2854	0.3508	0.2063	0.2684	0.4852
Semantic change ( <i>P</i> value)	2.94E-34	6.69E-05	5.02E-24	0.9034	5.38E-10	3.80E-07	0.08842	1.20E-30	1.16E-35	<1E-308
Grammaticality ( <i>P</i> value)	1.08E-190	5.81E-06	5.55E-27	8.47E-24	7.97E-45	2.96E-12	4.02E-18	6.54E-121	4.85E-209	<1E-308

**Table S1. Fitness correlation and *P* values**

Values indicate Spearman correlation and corresponding two-sided *P* values between fitness and either semantic change or grammaticality. A *P* value of <1E-308 indicates a value that was below the floating-point precision of our computer.

Model	HA H1	HA H3	Env BG505	Spike Baum et al. (5)	Spike RBD Greaney et al. (4)
MAFFT	0.697*	0.598*	0.523	0.618	0.526
EVcouplings (ind.)	0.706*	0.691*	0.536	0.689	0.527
EVcouplings (epi.)	0.726*	0.687*	0.552	0.713	0.610*
Grammaticality (our model)	0.820*	0.684*	0.667*	0.820*	0.704*
Bepler	0.660*	0.644*	0.561	0.534	0.664*
TAPE transformer	0.584*	0.526	0.574*	0.667	0.556
UniRep	0.482	0.452	0.534	0.745*	0.606*
Semantic change (our model)	0.664*	0.709*	0.622*	0.660	0.584*
CSCS (our model)	<b>0.834*</b>	<b>0.771*</b>	<b>0.692*</b>	<b>0.854*</b>	<b>0.709*</b>

**Table S2. Escape prediction normalized AUC values**

Normalized AUC values for escape prediction as plotted in **Figure 3B**, as well as separate AUCs for grammaticality and semantic change alone (this information is combined, as described in the **Supplementary Text**, for our model's full CSCS acquisition). Rows involving our model are highlighted in blue. An asterisk (\*) indicates a significant AUC based on a Bonferroni-corrected one-sided permutation-based *P*-value of less than 0.05.

<b>Virus</b>	<b>Region name</b>	<b>Region positions</b>
H1 WSN33	Head	59 – 291
	Stalk	18 – 58, 292 – 528
H3 Perth09	Head	68 – 293
	Stalk	17 – 67, 294 – 530
Env BG505	V1 loop	130 – 148
	V2 loop	149 – 195
	V3 loop	295 – 329
	V4 loop	383 – 415
	V5 loop	458 – 468
	CD4 binding loop	362 – 372
	Fusion peptide	509 – 529
	Immunosuppression	571 – 589
	MPER	659 – 680
	gp120 A	33 – 139
	gp120 B	144 – 508
	gp41	527 – 716
	Glycosylation	87, 132, 136, 147, 151, 196, 233, 261, 275, 294, 300, 337, 353, 361, 384, 390, 445, 608, 615, 634
Spike	S1 NTD	13 – 303
	RBD	319 – 541
	Fusion peptide	788 – 806
	HR1	920 – 970
	HR2	1163 – 1202
	S2	686 – 1273
	Glycosylation	17, 61, 74, 122, 149, 165, 234, 282, 331, 343, 603, 616, 657, 709, 717, 801, 1074, 1098, 1134, 1158, 1173, 1194

**Table S3. Escape potential regions of interest**

Residue positions corresponding to the regions of interest tested for enrichment or depletion of escape potential in **Figure 4**. All ranges are inclusive.



## References and Notes

1. M. B. Doud, J. M. Lee, J. D. Bloom, How single mutations affect viral escape from broad and narrow antibodies to H1 influenza hemagglutinin. *Nat. Commun.* **9**, 1386 (2018). [doi:10.1038/s41467-018-03665-3](https://doi.org/10.1038/s41467-018-03665-3) [Medline](#)
2. J. M. Lee, R. Eguia, S. J. Zost, S. Choudhary, P. C. Wilson, T. Bedford, T. Stevens-Ayers, M. Boeckh, A. C. Hurt, S. S. Lakdawala, S. E. Hensley, J. D. Bloom, Mapping person-to-person variation in viral mutations that escape polyclonal serum targeting influenza hemagglutinin. *eLife* **8**, e49324 (2019). [doi:10.7554/eLife.49324](https://doi.org/10.7554/eLife.49324) [Medline](#)
3. A. S. Dingens, D. Arenz, H. Weight, J. Overbaugh, J. D. Bloom, An antigenic atlas of HIV-1 escape from broadly neutralizing antibodies distinguishes functional and structural epitopes. *Immunity* **50**, 520–532.e3 (2019). [doi:10.1016/j.immuni.2018.12.017](https://doi.org/10.1016/j.immuni.2018.12.017) [Medline](#)
4. A. J. Greaney, T. N. Starr, P. Gilchuk, S. J. Zost, E. Binshtein, A. N. Loes, S. K. Hilton, J. Huddleston, R. Eguia, K. H. D. Crawford, A. S. Dingens, R. S. Nargi, R. E. Sutton, N. Suryadevara, P. W. Rothlauf, Z. Liu, S. P. J. Whelan, R. H. Carnahan, J. E. Crowe Jr., J. D. Bloom, Complete mapping of mutations to the SARS-CoV-2 spike receptor-binding domain that escape antibody recognition. *bioRxiv* 2020.09.10.292078 (2020). [doi:10.1101/2020.09.10.292078](https://doi.org/10.1101/2020.09.10.292078) [Medline](#)
5. A. Baum, B. O. Fulton, E. Wloga, R. Copin, K. E. Pascal, V. Russo, S. Giordano, K. Lanza, N. Negron, M. Ni, Y. Wei, G. S. Atwal, A. J. Murphy, N. Stahl, G. D. Yancopoulos, C. A. Kyratsous, Antibody cocktail to SARS-CoV-2 spike protein prevents rapid mutational escape seen with individual antibodies. *Science* **369**, 1014–1018 (2020). [doi:10.1126/science.abd0831](https://doi.org/10.1126/science.abd0831) [Medline](#)
6. M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations. *Proc. NAACL-HLT*, 2227–2237 (2018).
7. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners. *OpenAI Blog* **1**, 9 (2019).
8. T. A. Hopf, A. G. Green, B. Schubert, S. Mersmann, C. P. I. Schärfe, J. B. Ingraham, A. Toth-Petroczy, K. Brock, A. J. Riesselman, P. Palmedo, C. Kang, R. Sheridan, E. J. Draizen, C. Dallago, C. Sander, D. S. Marks, The EVcouplings Python framework for coevolutionary sequence analysis. *Bioinformatics* **35**, 1582–1584 (2019). [doi:10.1093/bioinformatics/bty862](https://doi.org/10.1093/bioinformatics/bty862) [Medline](#)
9. T. Bepler, B. Berger, Learning protein sequence embeddings using information from structure. [arXiv:1902.08661](https://arxiv.org/abs/1902.08661) [cs.LG] (2019).
10. R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, P. Chen, J. Canny, P. Abbeel, Y. Song, Evaluating protein transfer learning with TAPE. *Proc. Adv. Neural Inf. Process. Syst.*, 9686–9698 (2019).
11. E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, G. M. Church, Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods* **16**, 1315–1322 (2019). [doi:10.1038/s41592-019-0598-1](https://doi.org/10.1038/s41592-019-0598-1) [Medline](#)
12. Materials and methods are available as supplementary materials.

13. L. McInnes, J. Healy, UMAP: Uniform Manifold Approximation and Projection for dimension reduction. [arXiv:1802.03426](https://arxiv.org/abs/1802.03426) [stat.ML] (2018).
14. V. D. Blondel, J. L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, P10008 (2008). [doi:10.1088/1742-5468/2008/10/P10008](https://doi.org/10.1088/1742-5468/2008/10/P10008)
15. R. Xu, D. C. Ekiert, J. C. Krause, R. Hai, J. E. Crowe Jr., I. A. Wilson, Structural basis of preexisting immunity to the 2009 H1N1 pandemic influenza virus. *Science* **328**, 357–360 (2010). [doi:10.1126/science.1186430](https://doi.org/10.1126/science.1186430) [Medline](#)
16. K. G. Andersen, A. Rambaut, W. I. Lipkin, E. C. Holmes, R. F. Garry, The proximal origin of SARS-CoV-2. *Nat. Med.* **26**, 450–452 (2020). [doi:10.1038/s41591-020-0820-9](https://doi.org/10.1038/s41591-020-0820-9) [Medline](#)
17. M. B. Doud, J. D. Bloom, Accurate measurement of the effects of all amino-acid mutations on influenza hemagglutinin. *Viruses* **8**, 155 (2016). [doi:10.3390/v8060155](https://doi.org/10.3390/v8060155) [Medline](#)
18. N. C. Wu, J. Otwinowski, A. J. Thompson, C. M. Nycholat, A. Nourmohammad, I. A. Wilson, Major antigenic site B of human influenza H3N2 viruses has an evolving local fitness landscape. *Nat. Commun.* **11**, 1233 (2020). [doi:10.1038/s41467-020-15102-5](https://doi.org/10.1038/s41467-020-15102-5) [Medline](#)
19. H. K. Haddox, A. S. Diggins, S. K. Hilton, J. Overbaugh, J. D. Bloom, Mapping mutational effects along the evolutionary landscape of HIV envelope. *eLife* **7**, e34420 (2018). [doi:10.7554/eLife.34420](https://doi.org/10.7554/eLife.34420) [Medline](#)
20. T. N. Starr, A. J. Greaney, S. K. Hilton, D. Ellis, K. H. D. Crawford, A. S. Diggins, M. J. Navarro, J. E. Bowen, M. A. Tortorici, A. C. Walls, N. P. King, D. Veasler, J. D. Bloom, Deep mutational scanning of SARS-CoV-2 receptor binding domain reveals constraints on folding and ACE2 binding. *Cell* **182**, 1295–1310.e20 (2020). [doi:10.1016/j.cell.2020.08.012](https://doi.org/10.1016/j.cell.2020.08.012) [Medline](#)
21. K. Katoh, D. M. Standley, MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. *Mol. Biol. Evol.* **30**, 772–780 (2013). [doi:10.1093/molbev/mst010](https://doi.org/10.1093/molbev/mst010) [Medline](#)
22. Y. Xiao, I. M. Rouzine, S. Bianco, A. Acevedo, E. F. Goldstein, M. Farkov, L. Brodsky, R. Andino, RNA recombination enhances adaptability and is required for virus spread and virulence. *Cell Host Microbe* **19**, 493–503 (2016). [doi:10.1016/j.chom.2016.03.009](https://doi.org/10.1016/j.chom.2016.03.009) [Medline](#)
23. K. K.-W. To, I. F.-N. Hung, J. D. Ip, A. W.-H. Chu, W.-M. Chan, A. R. Tam, C. H.-Y. Fong, S. Yuan, H.-W. Tsoi, A. C.-K. Ng, L. L.-Y. Lee, P. Wan, E. Tso, W.-K. To, D. Tsang, K.-H. Chan, J.-D. Huang, K.-H. Kok, V. C.-C. Cheng, K.-Y. Yuen, COVID-19 re-infection by a phylogenetically distinct SARS-coronavirus-2 strain confirmed by whole genome sequencing. *Clin. Infect. Dis.* ciaa1275 (2020). [Medline](#)
24. E. Kirkpatrick, X. Qiu, P. C. Wilson, J. Bahl, F. Krammer, The influenza virus hemagglutinin head evolves faster than the stalk domain. *Sci. Rep.* **8**, 10432 (2018). [doi:10.1038/s41598-018-28706-1](https://doi.org/10.1038/s41598-018-28706-1) [Medline](#)

25. S. Ravichandran, E. M. Coyle, L. Klenow, J. Tang, G. Grubbs, S. Liu, T. Wang, H. Golding, S. Khurana, Antibody signature induced by SARS-CoV-2 spike protein immunogens in rabbits. *Sci. Transl. Med.* **12**, eabc3539 (2020). [doi:10.1126/scitranslmed.abc3539](https://doi.org/10.1126/scitranslmed.abc3539) [Medline](#)
26. Z. S. Harris, Distributional structure. *Word* **10**, 146–162 (1954). [doi:10.1080/00437956.1954.11659520](https://doi.org/10.1080/00437956.1954.11659520)
27. B. Hie, brianhie/viral-mutation: viral-mutation release 0.3. Zenodo (2020); [doi:10.5281/zenodo.4034681](https://doi.org/10.5281/zenodo.4034681).
28. B. Hie, Data for “Learning the language of viral evolution and escape”. Zenodo (2020); [doi:10.5281/zenodo.4029296](https://doi.org/10.5281/zenodo.4029296).
29. B. Foley, C. Apetrei, I. Mizrachi, A. Rambaut, B. Korber, T. Leitner, B. Hahn, J. Mullins, S. Wolinsky, HIV Sequence Compendium 2018, technical report LA-UR 18-2 (2018).
30. F. A. Wolf, P. Angerer, F. J. Theis, SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018). [doi:10.1186/s13059-017-1382-0](https://doi.org/10.1186/s13059-017-1382-0) [Medline](#)
31. F. Sievers, D. G. Higgins, Clustal Omega, accurate alignment of very large numbers of sequences. *Methods Mol. Biol.* **1079**, 105–116 (2014). [doi:10.1007/978-1-62703-646-7\\_6](https://doi.org/10.1007/978-1-62703-646-7_6) [Medline](#)
32. F. Ronquist, M. Teslenko, P. van der Mark, D. L. Ayres, A. Darling, S. Höhna, B. Larget, L. Liu, M. A. Suchard, J. P. Huelsenbeck, MrBayes 3.2: Efficient Bayesian phylogenetic inference and model choice across a large model space. *Syst. Biol.* **61**, 539–542 (2012). [doi:10.1093/sysbio/sys029](https://doi.org/10.1093/sysbio/sys029) [Medline](#)
33. A. Stamatakis, RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* **30**, 1312–1313 (2014). [doi:10.1093/bioinformatics/btu033](https://doi.org/10.1093/bioinformatics/btu033) [Medline](#)
34. M. N. Price, P. S. Dehal, A. P. Arkin, FastTree 2—Approximately maximum-likelihood trees for large alignments. *PLOS ONE* **5**, e9490 (2010). [doi:10.1371/journal.pone.0009490](https://doi.org/10.1371/journal.pone.0009490) [Medline](#)
35. S. Guindon, J. F. Dufayard, V. Lefort, M. Anisimova, W. Hordijk, O. Gascuel, New algorithms and methods to estimate maximum-likelihood phylogenies: Assessing the performance of PhyML 3.0. *Syst. Biol.* **59**, 307–321 (2010). [doi:10.1093/sysbio/syq010](https://doi.org/10.1093/sysbio/syq010) [Medline](#)
36. M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. McGettigan, H. McWilliam, F. Valentin, I. M. Wallace, A. Wilm, R. Lopez, J. D. Thompson, T. J. Gibson, D. G. Higgins, Clustal W and Clustal X version 2.0. *Bioinformatics* **23**, 2947–2948 (2007). [doi:10.1093/bioinformatics/btm404](https://doi.org/10.1093/bioinformatics/btm404) [Medline](#)
37. M. Balaban, N. Moshiri, U. Mai, X. Jia, S. Mirarab, TreeCluster: Clustering biological sequences using phylogenetic trees. *PLOS ONE* **14**, e0221068 (2019). [doi:10.1371/journal.pone.0221068](https://doi.org/10.1371/journal.pone.0221068) [Medline](#)
38. T. A. Hopf, J. B. Ingraham, F. J. Poelwijk, C. P. I. Schärfe, M. Springer, C. Sander, D. S. Marks, Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.* **35**, 128–135 (2017). [doi:10.1038/nbt.3769](https://doi.org/10.1038/nbt.3769) [Medline](#)

39. S. R. Eddy, A probabilistic model of local sequence alignment that simplifies statistical significance estimation. *PLOS Comput. Biol.* **4**, e1000069 (2008). [doi:10.1371/journal.pcbi.1000069](https://doi.org/10.1371/journal.pcbi.1000069) [Medline](#)
40. S. J. Gamblin, L. F. Haire, R. J. Russell, D. J. Stevens, B. Xiao, Y. Ha, N. Vasisht, D. A. Steinhauer, R. S. Daniels, A. Elliot, D. C. Wiley, J. J. Skehel, The structure and receptor binding properties of the 1918 influenza hemagglutinin. *Science* **303**, 1838–1842 (2004). [doi:10.1126/science.1093155](https://doi.org/10.1126/science.1093155) [Medline](#)
41. P. S. Lee, N. Ohshima, R. L. Stanfield, W. Yu, Y. Iba, Y. Okuno, Y. Kurosawa, I. A. Wilson, Receptor mimicry by antibody F045-092 facilitates universal binding to the H3 subtype of influenza virus. *Nat. Commun.* **5**, 3614 (2014). [doi:10.1038/ncomms4614](https://doi.org/10.1038/ncomms4614) [Medline](#)
42. G. B. E. Stewart-Jones, C. Soto, T. Lemmin, G.-Y. Chuang, A. Druz, R. Kong, P. V. Thomas, K. Wagh, T. Zhou, A.-J. Behrens, T. Bylund, C. W. Choi, J. R. Davison, I. S. Georgiev, M. G. Joyce, Y. D. Kwon, M. Pancera, J. Taft, Y. Yang, B. Zhang, S. S. Shivatare, V. S. Shivatare, C.-C. D. Lee, C.-Y. Wu, C. A. Bewley, D. R. Burton, W. C. Koff, M. Connors, M. Crispin, U. Baxa, B. T. Korber, C.-H. Wong, J. R. Mascola, P. D. Kwong, Trimeric HIV-1-Env structures define glycan shields from clades A, B, and G. *Cell* **165**, 813–826 (2016). [doi:10.1016/j.cell.2016.04.010](https://doi.org/10.1016/j.cell.2016.04.010) [Medline](#)
43. A. C. Walls, Y. J. Park, M. A. Tortorici, A. Wall, A. T. McGuire, D. Veelsler, Structure, function, and antigenicity of the SARS-CoV-2 Spike glycoprotein. *Cell* **181**, 281–292.e6 (2020). [doi:10.1016/j.cell.2020.02.058](https://doi.org/10.1016/j.cell.2020.02.058) [Medline](#)
44. P. Auer, Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.* **2020**, 397–422 (2003).
45. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) [cs.CL] (2019).
46. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality. *Proc. Adv. Neural Inf. Process. Syst.*, 3111–3119 (2013).
47. A. M. Dai, Q. V. Le, Semi-supervised sequence learning. *Proc. Adv. Neural Inf. Process. Syst.*, 3079–3087 (2015).
48. C. C. Aggarwal, A. Hinneburg, D. A. Keim, in *Proceedings of the International Conference on Database Theory* (2001), vol. 1973, pp. 420–434.
49. R. Sanjuán, M. R. Nebot, N. Chirico, L. M. Mansky, R. Belshaw, Viral mutation rates. *J. Virol.* **84**, 9733–9748 (2010). [doi:10.1128/JVI.00694-10](https://doi.org/10.1128/JVI.00694-10) [Medline](#)