communications biology

Article

https://doi.org/10.1038/s42003-024-07262-7

Paying attention to the SARS-CoV-2 dialect : a deep neural network approach to predicting novel protein mutations

Check for updates

Magdalyn E. Elkin ✉ & Xingquan Zhu ✉

Predicting novel mutations has long-lasting impacts on life science research. Traditionally, this problem is addressed through wet-lab experiments, which are often expensive and time consuming. The recent advancement in neural language models has provided stunning results in modeling and deciphering sequences. In this paper, we propose a Deep Novel Mutation Search (DNMS) method, using deep neural networks, to model protein sequence for mutation prediction. We use SARS-CoV-2 spike protein as the target and use a protein language model to predict novel mutations. Different from existing research which is often limited to mutating the reference sequence for prediction, we propose a parent-child mutation prediction paradigm where a parent sequence is modeled for mutation prediction. Because mutations introduce changing context to the underlying sequence, DNMS models three aspects of the protein sequences: semantic changes, grammatical changes, and attention changes, each modeling protein sequence aspects from shifting of semantics, grammar coherence, and amino-acid interactions in latent space. A ranking approach is proposed to combine all three aspects to capture mutations demonstrating evolving traits, in accordance with real-world SARS-CoV-2 spike protein sequence evolution. DNMS can be adopted for an early warning variant detection system, creating public health awareness of future SARS-CoV-2 mutations.

Since its emergence in December 2019, severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) has become a major global public health concern[1–3]. As SARS-CoV-2 moves from pandemic to endemic, questions arise to what the future evolutionary changes will be. SARS-CoV-2 is a positive-sense single-stranded ribonucleic acid (+ssRNA) virus, which has a higher rate of mutation compared to double-stranded RNA viruses and DNA viruses[4]. Nevertheless, SARS-CoV-2 encodes a proof-reading mechanism that results in a lower mutation rate relative to other ssRNA viruses, such as influenza (-ssRNA virus), HIV and Hepatitis C (both +ssRNA viruses)[5–8]. The mutation rate of SARS-CoV-2 has been recorded as 1.87×10^{-6} nucleotide substitutions per site per day[5]. Adaptive evolution in protein coding sequences is often expressed by the dN/dS ratio, which measures rate of non-synonymous mutations, (dN), to synonymous mutations (dS). Synonymous mutations do not alter the amino acid sequence and are largely presumed to be neutral mutations, while non-synonymous mutations (which alter amino acid sequences) may experience selection[9]. For SARS-CoV-2, this ratio has been estimated as 0.56[3]; which shows approximately half of non-synonymous mutations are lost as natural selection eliminates deleterious mutations[3,5].

Sources of mutations can be due to random RNA replication errors and host-mediated mutations[7,10]. Apolipoprotein B mRNA editing catalytic polypeptide-like enzymes (APOBECs) are cytidine deaminases involved in innate immune responses responses against viruses and introduce a characteristically high C-U nucleotide substitutions in ssRNA viruses[4], including SARS-CoV-2[7,10]. APOBECs play an important role in viral evolution, and are associated with immune escape and drug resistance[4]. The high rate of C-U mutations in SARS-CoV-2 introduce extra complexities with SARS-CoV-2 evolution, the mutations may be a large source of the non-synonymous mutations[7], and some of these mutations have been shown beneficial to viral fitness[10]. Repeated C-U mutations and reversions also may cause convergence[7]; where mutations arise independently in separate lineages, a strong indicator of positive selection and a path towards dominate mutations[11].

Mutations in the spike surface glycoprotein (i.e. spike protein), are of major importance because the spike protein mediates attachment of the virus to host receptors through the receptor-binding domain (RBD) and is the major target of neutralizing antibodies[12,13]. Many spike mutations that

Dept. Electrical Engineering and Computer Science, Florida Atlantic University, 777 Glades Road, Boca Raton, FL, 33431, USA.
✉e-mail: melkin2017@fau.edu; xzhu3@fau.edu

have become predominate have been shown to improve viral fitness. This can be due to methods such as increased viral survival, increased transmissibility, and immune detection evasion. For example, the mutation D614G in the spike protein has advantages for infectivity and transmissibility by increasing receptor binding avidity[13,14]. This mutation quickly became dominant worldwide[6,7]. Other mutations in the Spike protein, such as E484K and N439K have been recorded as escape mutations, as they evade immune detection responses by reducing antibody neutralization[8,13,14]. Omicron BA.2 sublineages have converged with multiple mutations in RBD residues (R346, K444, L452, N450, N460, F486, F490, Q493, and S494); creating a "variant soup" with combinations leading to increased fitness and waves of infectivity[15].

Singular advantageous mutations often accumulate in variant strands, leading to classifications of Variant of Concern (VOC) or Variant of Interest (VOI); VOCs and VOIs are variant strands of SARS-CoV-2 that contain mutations with notable changes in biological characteristics with potential impacts on transmissibility and immunity[13]. VOCs have a higher alert status and are classified with stronger evidence of negative clinical impact[16].

Future variants of SARS-CoV-2 should be expected to arise as the virus becomes endemic. Positive selection for SARS-CoV-2 will drive variants for increased transmissibility, longer duration of infection, and ability to evade immune responses. Thus may enable transmission to previously immunized populations leading to new waves of infection[5,7]. Thus, anticipating future variants is a great public health concern. Early detection systems can create a proactive response for immune therapies, vaccines, and provide an understanding of future mutation consequences impact on viral spread[7].

Predicting evolution is a difficult task, as evolutionary models face many complications including broad potential host range, animal transmissibility, and large degrees of randomness. However evolutionary models can aid in exploring possible evolutionary paths [5,17]. The sequence space for a given protein has an infeasible number of evolutionary paths[17]. Proteins are built using 20 standard amino acids from the genetic code. Thus a given protein with length N has a total number of 20^N possible combinations. Wet-lab experiments can be used to generate the protein and investigate the resulting phenotype, however these are time-consuming and costly and can't be feasibly done for all exhaustive combinations[18]. Methods such as Deep Mutational Scanning (DMS) are able to generate large-scale mutagenesis datasets to assess a broad range of amino acid mutations[19]. While DMS can analyze thousands of mutations in a single viral protein comprehensively, it also has disadvantages. The large scale of DMS may compromise data accuracy and DMS datasets have an inherent degree of noise[20].

Computational studies can be conducted as a complement for wet-lab experimentation. Computational studies can give insights on likely amino acid mutations which guide wet-lab experimentation. And in turn, phenotype information derived from DMS datasets can aid computational studies.

A number of computational and machine learning studies have been conducted on SARS-CoV-2. These include analyzing SARS-CoV-2 mutations based on biochemical properties[21] or transmissibility[22,23]; forecasting future emerging VOCs[16,24,25]; generating novel epitope protein sequences[18]; site mutation prediction[26]; conservation prediction[27]; and identifying escape mutations[28].

While the above listed studies all target SARS-CoV-2 protein mutation analysis, other computational studies have been conducted targeting nucleotide mutations. Previous studies have targeted predicting nucleotide mutation rates[29–31]; predicting mutable sites/positions[30–32]; and prediction of recurrent mutations driven by host factors[33].

Prediction of mutation rates and mutable nucleotide sites/positions can aid protein mutation analysis by identifying general trends in rates and mutation "hotspots". In addition, found dominant trends of codon changes can help identify the likely future amino acid changes. However, it has been noted that amino acid changes in the spike protein are more difficult to predict using nucleotide data (compared to mutations in M-Pro, a protein involved in replication). A listed potential source for this difficulty is that predicted mutations may be deleterious and not observed in the spike protein[30].

The above-mentioned studies are a small subset of the multitude of computational studies conducted targeting varying aspects of SARS-CoV-2 mutations. In this study, we aim to model SARS-CoV-2 spike protein mutations using protein language models and protein sequence data alone.

Natural Language Processing (NLP) methods have shown to tremendously benefit bioinformatics and protein research[34,35], such as the AlphaFold[36] which predicts protein structures using sequences. Protein sequences are analogous to sentences in human languages, where proteins are represented by a sequence (sentence) that can be broken down into the underlying amino acid tokens (words). Human language and protein sequences also both share information completeness. Using NLP methods with human language, we're able to derive information such as sentiment, topics, clauses, etc. Similarly, NLP methods with protein sequences are able to determine protein information such as structure and function, as this is directly encoded by the amino acid sequence[34]. In addition to protein language models (trained on amino acid language), Codon language models have shown benefits for protein engineering[37]. With Codon language models, the underlying nucleotide sequence is broken down to the 64-codon alphabet, which translates to the 20 amino acids, or a stop sequence. The codon alphabet is degenerate, with multiple codons encoding a single amino acid; however, it was shown that synonymous codon information can aid protein folding prediction, making codon language models beneficial for specific types of downstream tasks[37].

A common task in NLP is to predict a word given an input sentence context. Language models trained on a large number of texts learn the grammar rules of the language to guide word prediction for coherent and grammatical sentences. An equivalent task in protein research may be to predict an amino acid (word) given an input sequence (sentence) context. In order to this, a language model produces posterior probabilities for all amino acid tokens at a specified position in the protein sequence. The probability values represent how well a given amino acid (word) obeys the "grammar" of biological rules learned by the protein language model[28].

Language models are also frequently tasked with creating a word (or sentence) embedding of an input token or sentence. Embeddings are vectors that encode the language model's representation of the input sequence. Language models trained on a given language corpus attempt to represent semantically similar concepts with similar vector representations (embeddings)[34]. The difference between two separate semantic embeddings represents the degree of semantic similarity between the two sequences.

Protein language models have previously displayed useful applications in protein sequence design and generation[38,39]; mutation effect predictions[40–43]; forecasting emerging variants[16,24]; site mutation prediction[26]; and are able to identify likely escape mutations[28]. Our work expands on previous methods from Hie et al. to utilize language models for protein evolution. In the previous study, they created Constrained Semantic Change Search (CSCS) to identify likely escape mutations using the reference sequence. CSCS uses a mutation's grammaticality and semantic change to create a ranked value[28]. Grammaticality is the defined as the probability of an amino acid at a given position and determines if the amino acid obeys the biological rules (grammar) of the protein sequence. Semantic change is the difference in the embedding of the reference sequence to the mutated sequence.

In this study, we present `DNMS` (Deep Novel Mutation Search), a NLP approach to predicting amino acid mutations. *Deep* refers to usage of a transformer model (a deep neural network architecture). We utilize a Bidirectional Encoder Representation from Transformer (BERT) model that was trained on protein sequences[44]. The protein BERT (ProtBERT) model was pre-trained on 216 million proteins in the UniRef100 database[45]. The pre-trained ProtBERT model can be said to have learned the *language* of proteins from diverse species. We fine-tune ProtBERT to SARS-CoV-2 spike protein sequences, which is akin to refining the model on the dialect of SARS-CoV-2 spike protein. We create a SARS-CoV-2 sequence database from the NCBI database from December 2019 up to January 2023. Using a

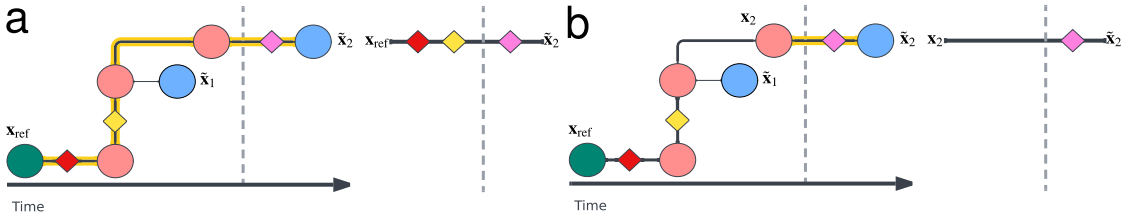

Fig. 1 | Visualization of an example phylogenetic tree to demonstrate differences between (a) mutating the reference sequence and (b) mutating a parent sequence. The blue circles represent sampled sequences, the green circle is the reference sequence (Wuhan-Hu-1), and the pink circles represent inferred internal nodes, which become parent sequences to leaf nodes (sampled sequences). The ancestral path of a given node can be traced back to the reference sequence. Sequences inherit mutations (marked with diamond shapes) from their ancestral path. Each diamond represents a unique mutation; the pink diamond represents a novel mutation, which has a first collection date after the cutoff point (gray dashed line).

cutoff date of January 1st, 2022, we create training and test sequences. The cutoff date represents a date in time where we can utilize information learned from all sequences collected to predict future mutations.

We define a *novel* mutation as one that has not previously been recorded in a sequence database. Each protein sequence in our database has a date of collection. We record the first date of a mutation from the minimum collection date of sequences containing the mutation. A *novel* mutation is one that first was recorded in a test sequence, i.e. the first recorded collection date is after the cutoff: January 1st, 2022.

Our goal is to predict novel amino acid substitution mutations; this goal differs from previous research as our ground truth mutations are previously unobserved mutations that have not occurred yet. Our prediction method involves a *Mutation Search* considering all possible amino acid substitution mutations for a given spike protein of interest. For each potential mutation, we use the fine-tuned ProtBERT model to calculate Grammaticality, Semantic Change and Attention Change. Grammaticality is the posterior probability of the mutated amino acid at the given position, it measures how well the mutation obeys the SARS-CoV-2 spike protein grammar, which the fine-tuned ProtBERT model has refined the dialect of. Semantic Change measures the difference in embedding space between the sequence of interest and the mutated sequence (sequence with an introduced mutation); this is a measure of similarity between two sequences. Attention change measures difference in attention weights between the sequence of interest and the mutated sequence; this is another measure of similarity from how the model is *paying attention* to the two sequences. Lastly, DNMS combines the three language model calculations with a ranking scheme to identify the most likely novel mutations.

Our target is any previously unrecorded amino acid substitution mutation in the SARS-CoV-2 spike protein sequence. And using DNMS we hypothesize that this can achieved using sequence data alone. Proteins follow biological rules encoded by the amino acid sequence. The pre-trained ProtBERT model has learned global biological language rules and is refined to SARS-CoV-2 spike dialect with fine-tuning. Mutations that obey these rules, will have corresponding higher grammaticality, the posterior probability of an amino acid at a given position given an input sequence. The protein language model infers statistical patterns of biological protein language rules resulting in identifying likely amino acids at given positions. This is similar to predicting an English word in a sentence given the surrounding sentence context. While fine-tuning helps refine specific patterns in SARS-CoV-2 spike protein, the downside is the model may be "over-fitted" to the training sequences and ultimately produce low probabilities for alternate amino acids, especially at positions with low genetic diversity in the training sequences. Thus we also use two measures of similarity, Semantic Change and Attention Change. These are measures of similarity between the sequence of interest (being mutated) and the mutated sequence (with an introduced amino acid substitution). Semantic Change measures the difference in embedding space between the two sequences; Attention Change is difference in attention weights (from attention heads in the transformer model) between the two sequences. As we are measuring single amino acid substitutions at a time, the mutated sequence will have a small change in the SARS-CoV-2 spike protein, and will result in a sequence that will be largely similar to the sequence of interest, as sequences with small changes in spike protein are typically clustered together in groups/clades. In DNMS, we rank all possible single-point amino acid substitutions to identify those with high grammaticality, low Semantic Change and low Attention Change. We hypothesize this can identify likely future mutations as those that obey rules of biological grammar, and are most similar to the sequence of interest.

While our target is based on genotype mutations based on sequence data alone, we have no inference on the resulting phenotype changes of these mutations. A single identified likely future mutation may have a neutral impact and not represent a relevant event in viral mutation. However, identifying likely future novel mutations may have applications in early variant warning detection systems. Computational studies that can identify likely future mutations can guide wet-lab experimentation towards the resulting mutated sequence to assess the resulting phenotypic characteristics. In our study, we identify single amino acid substitutions. While a single amino acid substitution may not lead to increased fitness alone; as seen with Omicron BA.2 sublineages, where the number of mutations in specific RBD residues correlates with increased fitness[15]. With all current circulating sequences, it would be infeasible to assess all single-point amino acid substitutions in wet-lab experimentation to assess future fitness consequences, where as computational studies for early warning detection systems can complement these with analyzed likelihood. In addition, a computational study that can identify the same likely future mutation over different lineages can aid in understanding evolutionary trends toward future emerging variants.

Previous studies on forecasting future emerging variants rely heavily on knowledge of current variants[16,24,25]. This limits the predictive value of such models, as it requires the variants to be known and recorded prior to prediction. Similarly, models based on singular adaptive traits, (transmissibility or escape potential) are often insufficient basis for predictions, reducing model complexity and predictive power[17]. Our approach differs from related work by Zhou et al.[26] in that we use a sequence timeline-based sampling method in order to derive a true novel mutation test set that represents previously unseen mutations. Predicting previously unobserved mutations is theoretically an intractable problem. But the benefit has great implications for early and quick anticipation of likely future mutations.

Evolution is a continuous process. Over time, mutations have occurred and become predominant. This changes the dialect of SARS-CoV-2 spike protein that the ProtBERT model was trained on. The reference sequence is used for comparison to measure genetic diversity and record future mutations; the reference sequence is also commonly used in computational studies to analyze mutations. However, the reference sequence doesn't contain the genetic diversity that our language model was trained on, thus doesn't have the proper context in order to accurately predict future amino acid changes. Using DNMS, we demonstrate that mutating a parent sequence in a phylogenetic tree has a greater advantage in determining likely future mutations. Fig. 1 visualizes the difference between (a) mutating the

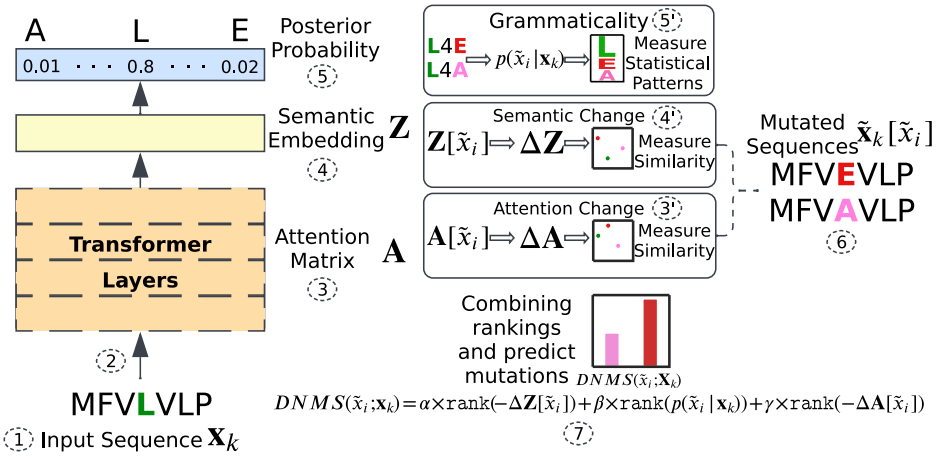

Fig. 2 | Summary of Deep Novel Mutation Search (DNMS). DNMS starts with ① an input sequence being fed into ProtBERT ②. From ProtBert DNMS extracts ③ the attention matrix **A**; ④ a protein semantic embedding **Z**; and ⑤ output posterior probability. DNMS calculates ③' Attention Change, ④' Semantic Change and ⑤' Grammaticality for every single point amino acid substitution for the input sequence. In this example, two mutations are visualized at position $i = 4$, where the input sequence has token L, x_i = L. The two mutations are L4A and L4E, denoted by $\tilde{x}_i$. ⑤' Grammaticality, denoted with $p(\tilde{x}_i|\mathbf{X}_k)$, for the two mutations are calculated from the posterior probability output from ProtBERT,⑤. Grammaticality is a measure of statistical patterns learned from the fine-tuned ProtBERT model. For each mutation, we pass into ProtBERT the mutated sequence, $\tilde{\mathbf{X}}_k[\tilde{x}_i]$ which represents the input sequence with the introduced mutation at position i. ③' We obtain the attention matrix for the mutated sequence, $\mathbf{A}[\tilde{x}_i]$, and calculate Attention Change (change from **A**), $\Delta\mathbf{A}$, which is a measure of similarity. ④' We obtain a protein semantic embedding for the mutated sequence, $\mathbf{Z}[\tilde{x}_i]$, and calculate Semantic Change (change from **Z**), $\Delta\mathbf{Z}$, which is an additional measure of similarity. ⑦ DNMS combines the rankings of Semantic Change, Grammaticality, and Attention Change; prioritizing high Grammaticality, and low Semantic Change and Attention Change. Future novel mutations are discovered using $\text{DNMS}(\tilde{x}_i; \mathbf{X}_k)$.

reference sequence and (b) mutating a parent sequence with an example phylogenetic tree. The green circle is the reference sequence. Pink circles represent inferred internal nodes, which become parent sequences to leaf nodes, represented by blue circles, which are sampled sequences in the database $\mathcal{S}$. Mutations are represented by diamonds. A gray dashed line shows a cutoff point, where everything prior to the cutoff point can be used for training and novel mutations that occur after the cutoff point we wish to predict for. Sequence $\tilde{\mathbf{x}}_1$ is pre-cutoff and part of the training data. Sequence $\tilde{\mathbf{x}}_2$ is post-cutoff and represents the first collection date of a sequence with the pink diamond mutation. Since this is after the cutoff, the pink diamond is a novel mutation that we aim to predict. In the case where red and yellow mutations have become predominant in the training dataset (and have become part of the dialect of the spike protein language), the model will give more weight to these mutations when mutating the reference sequence, as the predominant mutations aren't a part of the context of the reference sequence. Consequently, a model may predict previously recorded and predominant mutations when mutating the reference sequence. Additionally, the similarity between the reference sequence and the sequence $\tilde{\mathbf{x}}_2$ will be decreased, as the reference sequence doesn't have the inherited mutations that occurred prior to cutoff. Thus instead we mutate a parent sequence, $\mathbf{x}_2$, which is more representative of sequences circulating around the time of the cutoff that may mutate in the future.

In our approach, DNMS takes parent sequences from a phylogenetic tree that has child sequences with novel mutations (previously unrecorded in pre-cutoff) and mutates the parent sequences in silico in order to predict the future mutations. The contextual information from the parent sequence is used to calculate the grammaticality of potential mutations, and which mutated sequences are closest in semantic embedding and attention. These future likely mutations are determined by the ranking objective of DNMS. Thus we hope our methods can be adopted for public health awareness of future SARS-CoV-2 mutations, while providing a cohesive framework for collecting and analyzing SARS-CoV-2 sequences and a novel application of ProtBERT for future mutation prediction.

Our main contributions are as follows:

- We create a process for sequence data collection, analysis and prediction for novel mutation prediction that can be easily adopted for public health awareness.
- We demonstrate that fine-tuning a pre-trained language model to the dialect of SARS-CoV-2 protein sequences can predict likely novel mutations.
- We demonstrate that context matters in the prediction; mutating a parent sequence to predict future novel mutations has better performance compared to mutating the reference sequence.
- We demonstrate that attention weights have predictive power for mutations. Additionally, adding addition weights, semantic change and grammaticality proprieties of potential novel mutations has the best predictive performance.
- We investigate the correlation between protein language model calculations relationship with wet-lab experiments.

Results

Method summary

In this paper, we introduce a new method, Deep Novel Mutation Search (DNMS) to find future novel amino acid substitution mutations in SARS-CoV-2 spike protein. "Deep" refers to the usage of a multi-layer language model (ProtBERT) to analyze potential mutations (search) for the most likely future previously unseen (novel) by calculating grammaticality, semantic change and attention change given a sequence of interest. Fig. 2 shows a summary of DNMS.

For a given SARS-CoV-2 spike protein sequence, we mutate in silico all possible single-point amino acid substitutions. For each mutated sequence, we use the fine-tuned ProtBERT model to calculate the probability of the mutated amino acid at the given position. The difference in embedding space from the given sequence and the mutated sequence is also calculated. These are referred to as grammaticality and semantic change, respectively, as reported by Hie et al.[28]. Our method advances previous research by analyzing attention in a protein transformer model with regards to a given sequence and a mutated sequence. Previously, attention in protein transformer models have been shown to be associated with structural and functional properties of proteins[46]. However, to the best of our knowledge, attention analysis hasn't been applied for mutation prediction.

As our objective is to find the most likely single amino acid substitutions that will occur in the future, we hypothesize that the mutated sequence will have similarity to the sequence of interest, as sequences with small

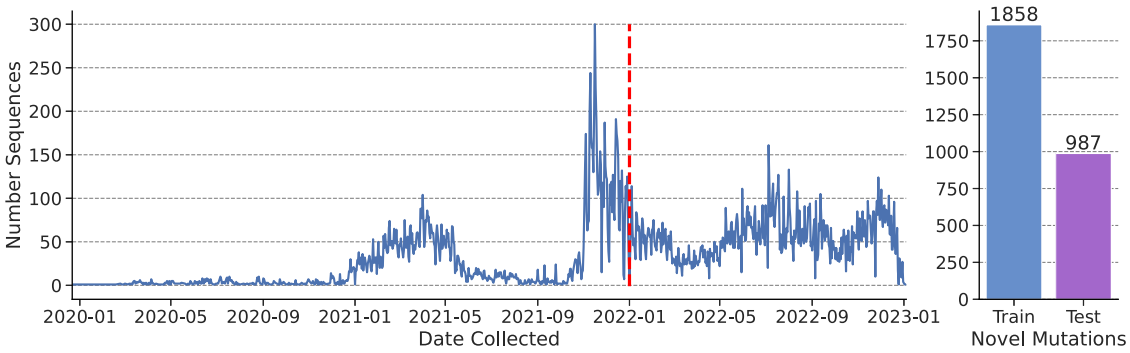

Fig. 3 | Sequences collected over time and number of mutations in training and test sets. The left side graph shows the number of unique sequences collected per date. The red line indicates the cutoff date (January 1st, 2022). A novel mutation is based on the first collection date of a sequence that contained said mutation. Right-hand graph shows the number of substitution mutations in the training set, where the first date was before the cutoff *vs* number of substitution mutations in the test set, where first date is after cutoff.

changes in the spike protein are typically clustered together within a group or clade. Additionally, the mutated amino acid should largely obey the grammar rules of the protein sequence, where such rules are learned by the ProtBERT model that was fine-tuned on the specific dialect of SARS-CoV-2 spike proteins. `DNMS` ranks all possible single-point amino acid substitutions to determine the ones with highest grammaticality, lowest semantic change and lowest attention change. These represent mutations that obey the grammar rules, and are closest in similarity to the sequence of interest and are determined to be the most likely future mutations.

Sequence dataset

The SARS-CoV-2 sequence database, $\mathcal{S}$, consists of sequences from December 2019 up to January 2023. After data filtering steps (see Supplementary Fig. 1) and retaining only unique spike sequences, the final sequence database has a total of $n = 35,943$ sequences. Using a cutoff date of January 1st, 2022, $n = 15,871$ sequences are pre-cutoff and $n = 20,072$ sequences are post-cutoff. We classify a mutation as being "novel" if it has not previously been seen in a sequence from database $\mathcal{S}$. The first collection date of a sequence given a particular mutation records when the mutation is novel. From the post-cutoff sequences, there are $n = 987$ novel mutations. Because the cutoff date is around the emergence of Omicron clades, the large majority (98%) of the test set mutations are Omicron specific. The number of sequences collected over time and the mutations in the training and test sets are shown in Fig. 3.

In order to derive parent-child relationships, the sequence database is used to build a phylogenetic tree using Nextstrain[47]. Supplementary Fig. 2 shows a subset of the phylogenetic tree; the full phylogenetic tree is shown in Supplementary Fig. 3.

Visualizations of sequence groups

In order to understand protein semantics and attention weight values w.r.t. sequence groups (clades) we display *t*-SNE clustering in Fig. 4. The top row, Fig. 4a, b, displays *t*-SNE clustering of full protein embeddings. The bottom row, Fig. 4c, d, displays *t*-SNE clustering of protein attention weight matrices. The left panel, Fig. 4a, c, shows all sequences, samples are labeled according to WHO or Nextstrain clade label. Nextstrain clades are labeled first based on the year (19 for 2019, 20 for 2020, etc.) and then subgroups defined with a single letter. For simplicity, 19A-B combines 19A and 19B; and 20A-F combines 20A, 20B, 20C, 20D, 20D, 20E and 20F. Due to the large number of sub-Omicron groups, the right panel, Fig. 4b, d, highlight Omicron subgroups, colored by Nextstrain clade label.

Generally, sequences within a clade and within sub-groups of Omicron are clustered together, with separations shown for both protein embedding values and attention weight matrices. This clustering of sequences within a clade together indicate that the protein language model is able to generate a semantic representation of a sequence that accurately reflects the genetic encoding or meaning of the sequence, as sequences in a clade are typically similar genetically. Additionally, the clustering with attention weights shows that the model is paying attention to sequences within a clade similarly. Together, the *t*-SNE clustering demonstrates the ability of protein semantics and attention weights for similarity measures.

As new mutations arise that are inherited from a parent to a child, it's likely that the child sequence will be in the same clade as the parent, especially when only considering single-point amino acid changes. Our hypothesis for determining future mutations is that the mutated sequence will have similar semantic representations and attention weights as the sequence of interest. The two sequences will have similar structure and genetic characteristics and will likely be in the same clade or grouping. Fig. 4 demonstrates that the ProtBERT model is able to represent this similarity by creating protein embeddings and have attention weights that place the genetically similar samples close in embedding space. Thus Fig. 4 shows validity to our DNMS ranking objective.

Variant fitness analysis

In order to analyze the relationship between the ProtBERT model calculations for mutations and mutational fitness, a deep mutational scanning (DMS) wet-lab experiment for mutations in the SARS-CoV-2 spike protein receptor-binding domain (RBD) is obtained from Cao et al.[48]. Using DMS data collected for the reference sequence, and two Omicron samples from sublineages BA.1 and BA.2, we compared grammaticality, semantic change, attention change and DNMS values for the mutations in the RBD with fitness values in the DMS dataset. The DMS dataset represents wet-lab experimental fitness values of ACE2 receptor binding affinity and RBD expression, which is a measurement of protein stability[49]. For both binding and expression, negative values suggest a deleterious mutation and positive values indicate enhanced binding/expression; higher values indicate higher viral fitness.

Figure 5 displays correlation between the language model calculations and the viral fitness for the reference sequence, a BA.1 sequence and a BA.2 sequence. For all three variants of SARS-CoV-2, grammaticality and DNMS show a positive correlation with viral fitness; semantic change and attention change both show negative correlation with viral fitness. All correlations were found to be statistically significant with a Bonferroni-corrected *p*-value of less than 0.05. Supplementary Figs. 4, 5, and 6 show correlation scatter plots for the three sequence experimental fitness and language model calculations.

Between the fitness scores for binding and expression, the language model correlations are shown to be stronger for expression. Indicating that the language model calculations, particularly grammaticality, are more indicative of protein stability effects of mutations than binding effects. Grammaticality represents the probability values of a single mutation which demonstrates the mutation obeying the grammar rules of a protein. Mutations that affect the stability of the protein may then be disobeying these grammar rules as learned by the ProtBERT model.

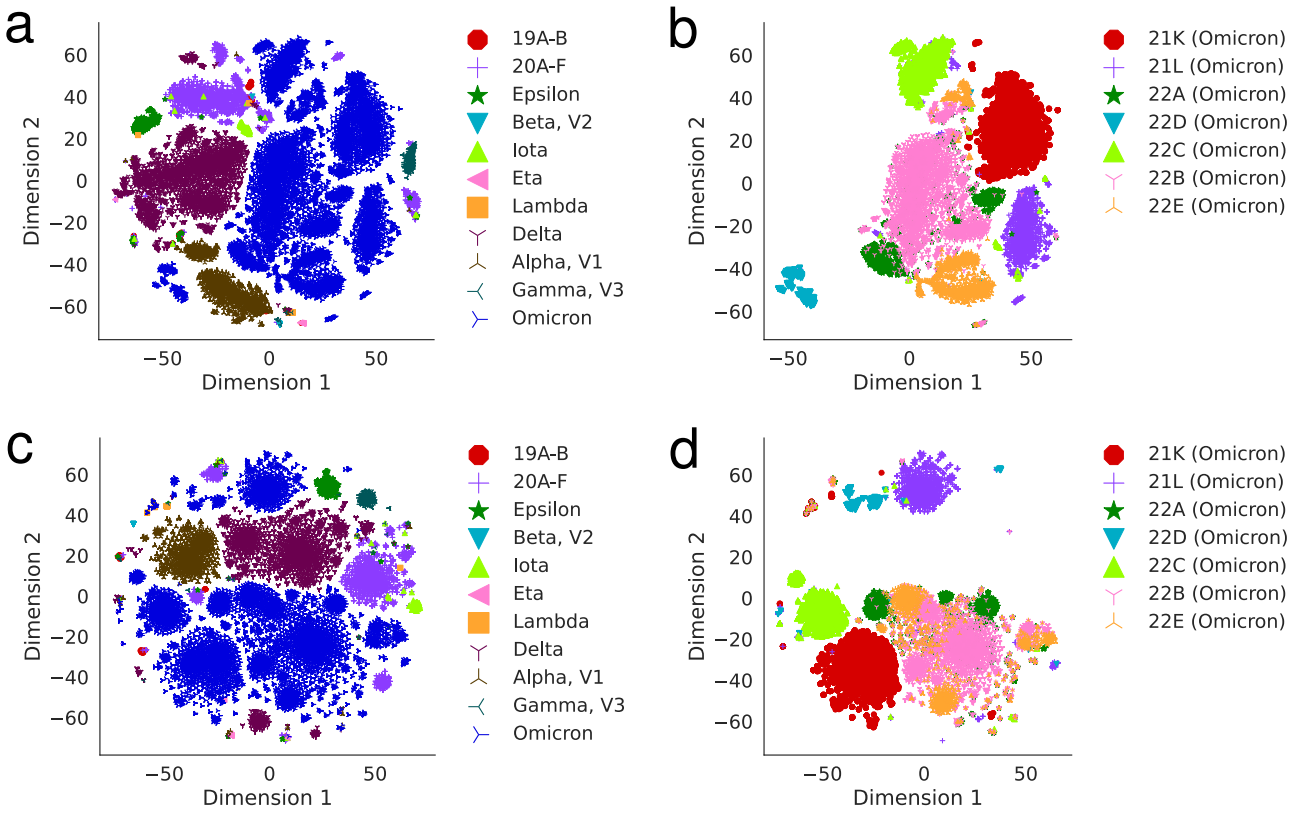

Fig. 4 | Visualization of sequence groups (clades). *t*-SNE clustering of Protein Embedding Values (**a**, **b**) and Attention Weight Matrices (**c**, **d**). All sequences are shown in (**a**, **c**), colored by WHO label or Nextstrain clade label; and Omicron subclades are highlighted in (**b**, **d**), colored by Nextstrain clade label.

Fig. 5 | DMS experimental results of ACE2 receptor binding and RBD expression for RBD mutations correlation with grammaticality, semantic change, attention change and DNMS calculated from ProtBERT model. Only mutations with experimental DMS values are considered. For the reference sequence, $n = 3802$ mutations have ACE2 receptor binding values, and $n = 3798$ mutations have RBD expression values for BA.1, $n = 3819$ mutations are included; for BA.2, $n = 3801$ mutations are included. All correlations were found to be statistically significant with a Bonferroni-corrected p-value of less than 0.05.

With respect to mutation binding effects, it's been shown that many mutations show neutral or enhanced ACE2 binding affinity, many of these may have corresponding constraints for protein stability[50]. This suggests a mutation may be disobeying the grammar rules of the protein, and in turn have a detrimental effect on expression, but still can show enhanced ACE2 binding. Binding affinity is also dependent on the ACE2 receptor, which may have flexible methods of attachment with SARS-CoV-2. Our language model has no prior information of the ACE2 interface, which results in decreased correlation of output probability values against binding effects.

For RBD expression, the correlation of grammaticality is higher with BA.1 and BA.2 compared to the reference sequence. Due to the high circulation of Omicron sequences, more Omicron sequences will be present in the training set. The fine-tuned ProtBERT model effectively adjusts the SARS-CoV-2 dialect learned with a stronger bias towards Omicron sequences and subsequently shows a higher predictive value towards mutation protein stability effects for these sequences. Changing the context of the protein being mutated (BA.1 and BA.2 *vs* reference) has benefits for mutation prediction.

Grammaticality and DNMS are associated with higher experimental fitness; additionally, higher semantic change and higher attention change values are associated with lower experimental fitness values. Together, this gives more support to our ranking scheme of prioritizing high grammaticality values and small attention change and small semantic change for mutation prediction. DNMS combines these rankings for mutations and ultimately we prioritize higher DNMS scores for mutation prediction.

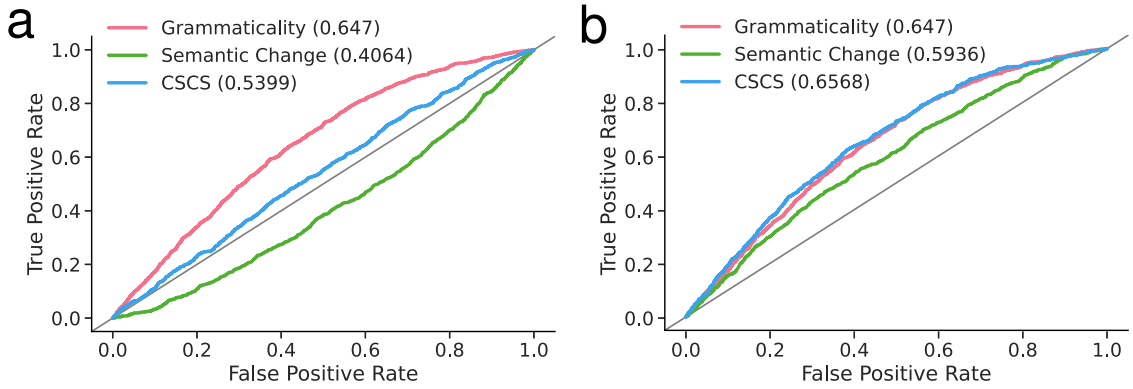

Fig. 6 | CSCS calculated from Hie et al.[28] biLSTM mutating the reference sequence and tested against the $n = 987$ novel mutations in the test set. A gray line indicates AUC=0.5, which is equivalent to random guessing. In (a) the original ranking method is shown and (b) shows an updated ranking method.

Comparison against previous work

In order to compare our results with previous work by Hie et al.[28], we obtain their biLSTM language model outputs of Grammaticality and Semantic Change using the reference sequence as the sequence of interest to mutate. We calculate AUC performance using these values with treating our test set of $n = 987$ mutations as ground truth mutations.

Figure 6 presents AUC scores and ROC curves for CSCS with the previous biLSTM model. First we consider the original ranking method (Fig. 6a), where semantic change is ranked from highest to lowest. In their previous study, this ranking method was used in order to find protein sequences that had greatest semantic changes which reflect a conformation change in the protein in order to produce an escape mutation, as their objective was solely to identify likely escape mutations. However, this ranking objective doesn't hold true when searching for novel mutations. To test our new ranking objective, we use an adjusted CSCS, with an updated ranking method (Fig. 6b), where semantic change is ranked from lowest to highest. Updating the ranking method shows a high performance improvement for Semantic Change alone and consequently the CSCS combined calculation of Semantic Change and Grammaticality.

These results indicate that when testing for novel future mutations, lower semantic change should be prioritized over higher semantic change. Switching the ranking scheme essentially flips the semantic change ROC curve from below AUC = 0.5 (random guessing) to above AUC = 0.5. These preliminary results together with the variant fitness analysis correlations give further validity to our DNMS ranking objective.

While adjusting the ranking method, (Fig. 6b), does show an improvement in performance, the AUC scores are still notably low. The biLSTM model was trained on spike protein sequences in the *Coronavidae* family towards the beginning of the pandemic (2021)[28]. Consequently, the model wasn't fine-tuned on the specific dialect of SARS-CoV-2 spike protein up to our cutoff point (January 1st, 2022). The lower performance here can also be attributed to the model lacking the context of predominate mutations that may aid novel mutation prediction.

In subsequent sections, we utilize an "adapted" version of CSCS that utilizes a different similarity metric calculation (ℓ_2 norm *vs* ℓ_1 norm), the updated ranking scheme for semantic change and utilizes a ProtBERT model. The inclusion of the "adapted" CSCS is to compare the difference between Grammaticality + Semantic Change and Grammaticality + Semantic Change + Attention Change (DNMS). For simplicity sake, the term "CSCS" will be used to refer to the adapted version of CSCS.

Deep novel mutation search results

In DNMS, we take a parent sequence from a phylogenetic tree and mutate it in silico to obtain all possible single-point amino acid substitutions. From the parent sequence, we calculate the protein (semantic) embedding and attention weight matrix from the ProtBERT model. For each mutation, we calculate the posterior probability (grammaticality) output from the ProtBERT model given the parent sequence input. For each mutated sequence with a single amino acid change, we calculate protein embedding and attention weight matrix. Semantic change is calculated as the difference from the parent sequence embedding to the mutated sequence embedding. Attention change is calculated as the difference from parent sequence attention matrix and mutated sequence attention matrix. Ground truth mutations are those in the child sequence(s) that were first recorded post-cutoff. Parent sequences are selected for those that have child sequence(s) with novel mutations. For a baseline comparison, we repeat the process using the reference sequence and all novel mutations in the test set.

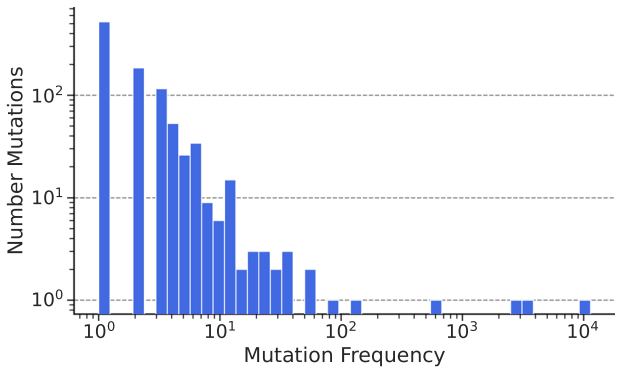

Fig. 7 | Distribution of mutation frequency in the test set. Histogram showing number of mutations in the test set against the mutation frequency.

In order to visualize the acquisition ranking objective of DNMS, Supplementary Fig. 7 displays Grammaticality, Semantic Change, Attention Change for a single parent sequence that had $n = 234$ unique mutations in the test set, novel mutations are marked with a red star.

For the $n = 987$ novel mutations in the test set, we record the mutation frequency, which is defined as the number of times a sequence contained that mutation in sequence database $\mathcal{S}$. A histogram showing the count of mutations at different frequencies is shown in Fig. 7. Where the majority of mutations only were recorded in database $\mathcal{S}$ once. With the wide genetic surveillance of SARS-CoV-2, it's potential that a single recorded mutation may be deleterious or largely neutral. While mutations with high frequency may also be neutral, they have a less chance of being deleterious, as deleterious mutations are quickly lost. Because of this, we first demonstrate our results with DNMS using an increasing threshold. Each step of the increased threshold excludes mutations if the frequency is below the threshold. In this way, we're able to visualize results for mutations that are of higher interest (greater frequency).

We compare five separate methods, Grammaticality, Semantic Change, Attention Change, CSCS (Grammaticality + Semantic Change) and DNMS (Grammaticality + Semantic Change + Attention Change)

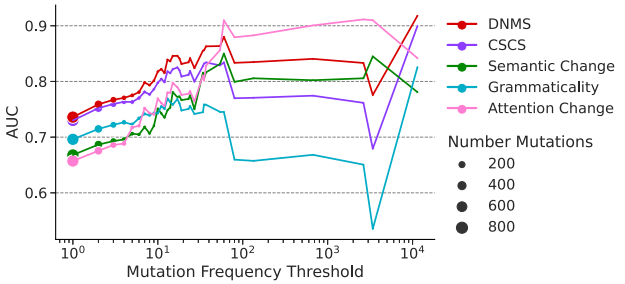

Fig. 8 | AUC scores for the different methods compared against increasing mutation frequency threshold. At each increase of the threshold, mutations with frequency lower than the threshold are excluded.

Table 1 | Average AUC results over all mutations (Global Average) and over increasing thresholds (Threshold Average) for the Baseline Reference and Parent Sequence experimental setups

	Reference sequence		Parent sequences	
	Global Average	Threshold Average	Global Average	Threshold Average
Grammaticality	0.6606	0.6458	0.6961*	0.7298*
Semantic Change	0.6338	0.5935	0.6680	0.7617*
Attention Change	0.6105	0.6670	0.6572*	0.7884*
CSCS	0.6923	0.6675	0.7300	0.7948*
DNMS	0.6974	0.6923	0.7360	0.8228*

A star * indicates where a method was significantly higher, ($p < 0.05$) than the reference baseline experiment after a Boneferonni corrected t-test.

against test set mutations in increasing frequency thresholds in Fig. 8. At point 10^0, all mutations are included. At each point in the graph, any mutation with a frequency lower than the threshold is excluded. In general, DNMS outperforms all methods except for a select few mutation frequency threshold sets. As DNMS contains all singular components (Grammaticality, Semantic Change, Attention Change), when one method shows poorer performance, the entire performance can show a corresponding decrease. In general, the addition of all three components shows the highest performance, as DNMS is able to utilize all available information.

To compare the five methods against the baseline reference sequence experiment *vs.* the parent sequence experiment in Table 1. We report two averages, a "global" average, simply an average of all single mutation results and a threshold average, the average of results collected over increasing thresholds (as displayed in Fig. 8). A star indicates where a method is significantly higher ($p < 0.05$) than the corresponding reference baseline experiment after a Bonferroni-corrected t-test.

In general, methods have higher performance in the parent sequence experiment, with all methods being significantly higher when considering the threshold average. Additionally, the combined methods (CSCS and DNMS) perform higher than the single methods (grammaticality, semantic change and attention change); with DNMS having the highest performance of all methods.

In order to compare the methods against each other, we perform a Friedman test comparing the methods against different datasets.

First, we consider each individual parent sequence ($n = 359$) tested as individual datasets. A Friedman test demonstrates a significant difference between the five methods, $\chi_F^2 = 152.642, p = 5.53\text{e}{-32}$. The Nemenyi post-hoc test, using $\alpha = 0.05$, results in Fig. 9a demonstrates that DNMS is the highest performant method, followed by CSCS, Attention Change, Grammaticality and Semantic Change.

Second, we consider the average of AUC scores over different mutation frequency thresholds as individual datasets ($n = 32$ separate threshold groups). A Friedman test demonstrates a significant difference between the five methods, $\chi_F^2 = 80.575, p = 1.32\text{e}{-16}$. The Nemenyi post-hoc test, using $\alpha = 0.05$, results in Fig. 9b demonstrates that DNMS is the highest performant method, followed by CSCS, Attention Change, Semantic Change and Grammaticality.

In the critical difference diagrams in Fig. 9, groups of methods that are not statistically significantly different are grouped together with a bar. Thus in Fig. 9a, Grammaticality and Semantic Change are not statistically significantly different and in Fig. 9 CSCS and Attention Change are not statistically significantly different, etc.

Together, these results demonstrate the superiority of DNMS compared to all other methods and also a higher performance for CSCS. This indicates one calculation alone (Grammaticality, Semantic Change or Attention Change) is not as performant as combining the information for novel mutation prediction.

Predictions for specific spike mutations of interest. To highlight specific individual mutation results, we identify mutations in the test set that are listed as spike mutations of interest for variants classified as VOC and/or VOI. Current VOCs, VOIs, Variants under monitoring and de-classified variants for SARS-CoV-2 and individual spike mutations of interest can be found at the European Centre for Disease Prevention and Control (ECDC) website, https://www.ecdc.europa.eu/en/covid-19/variants-concern. From this site, we confirmed four mutations in our test set are spike mutations of interest for variants classified as VOIs: D339H, K444T, N460K, and S486P. Additionally, we include mutation F486V, as it as a defining mutation for the Pango lineage BA.4 and BA.5. Both lineages are Omicron-descendent sublineages and were classified as VOC by ECDC from May 2022 until March 2023. Pango lineages are a hierarchical family tree naming convention to help identify unique subgroups within a larger classification system[51]. Current constellations (a collection of mutations found in a Pango lineage) for lineages of concern and genomically interesting regions can be found at https://github.com/cov-lineages/constellations.

For the five highlighted mutations, Supplementary Table 1 lists the mutation, the mutation *w.r.t.* the reference, the first date the mutation was found in a sequence in $\mathcal{S}$, the frequency of the mutation and its biological significance. Fig. 10 displays results for the five mutations, which shows averaged methods over all tested parent sequences that contained the corresponding mutation. All mutations are from Omicron sublineages.

Except for mutations N460K and S486P, DNMS is able to achieve over 0.9 AUC, indicating our method would be beneficial to predict clinically significant future mutations.

Comparison of transformer models. DNMS uses the term *Deep* to illustrate the usage of a deep transformer model, with multiple layers, each layer requires at least one attention head. With a transformer model, we're able to calculate the three components of DNMS, Grammaticality, Semantic Change and Attention change. In this paper and the results presented in earlier sections, we validate our method using a ProtBERT model, however other protein transformer models could be used in place of ProtBERT.

To validate our choice in ProtBERT, we test two other pre-trained protein transformer models, known as Evolutionary Scale Modeling (ESM). We investigate two ESM models, a 12 layer transformer model (ESM1 t12) and a 34 layer transformer model (ESM1 t34)[52].

The ESM models are first fine-tuned similarly to our ProtBERT model. We then check performance of the models by mutating the reference sequence and testing performance with the 987 mutations. We report the results of ProtBERT *vs.* ESM in Supplementary Fig. 8 which displays the global mutation average AUC against the reference sequence.

We show that performance for Semantic Change and Attention change is similar between the three models, except for ESM1 t34, with lower

Fig. 9 | Critical Difference (CD) Diagram showing statistical differences of AUC scores against five language model calculation methods. Critical Difference Diagram comparing methods based on (**a**) individual $n = 359$ parent sequences (CD=0.322) and (**b**) $n = 32$ mutation thresholds (CD=1.08).

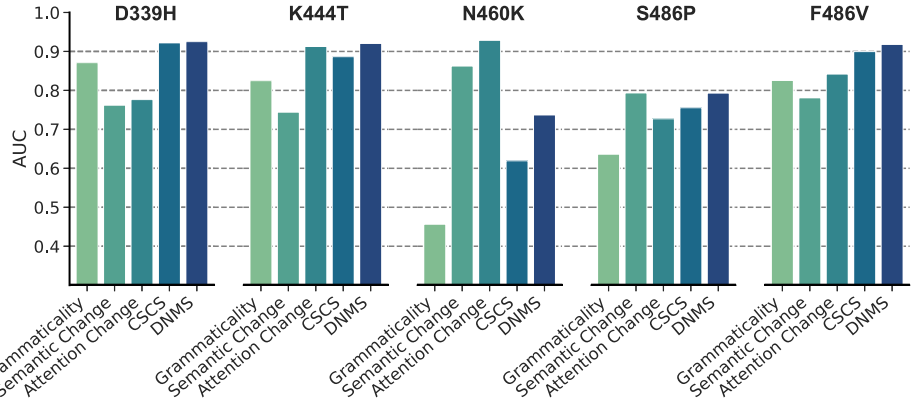

Fig. 10 | Highlighted mutation AUC scores. AUC scores for Grammaticality, Semantic Change, and Attention Change, CSCS, DNMS for five individual specific spike mutations of interest.

performance in attention change. Notably Grammaticality has lower performance with ESM1 t12 and in ESM1 t34, leading to dramatically decreased AUC in CSCS and DNMS for the ESM models.

While any protein transformer model is valid for our DNMS approach, these results validate our choice in ProtBERT.

Discussion

In this study, we aim to predict previously unseen mutations in SARS-CoV-2 spike protein using a fine-tuned language model that has learned the dialect of SARS-CoV-2 spike proteins. Potential mutations are ranked in order to determine those that have high grammaticality (posterior probability) and are similar to the protein of interest with lowest semantic change and attention change. When visualizing semantic embedding and attention matrices, we show that the language model is able to separate sequence groups, indicating similarity can be inferred with small semantic change and attention change. Viral fitness analysis demonstrates that high grammaticality is associated with higher viral fitness, while small semantic change and attention change are associated with higher viral fitness. For future mutation prediction, these are combined together in DNMS to create an acquisition ranking that prefers high grammaticality, low semantic change and attention change. Our statistical tests demonstrate that DNMS is superior for novel mutation prediction, as it combines all available information for the prediction.

Context matters

Changing the context of the protein of interest being predicted from the reference sequence to a parent sequence has higher performance as the reference sequence doesn't contain many of the mutations in sequences in the training set used for fine-tuning ProtBERT. For example, the mutation D339H at position 339, went through multiple steps in spike protein evolution, G339D - D339H. While the end amino acid H is still being predicted in the reference and a given parent sequence, the context of that prediction (G *vs* D) has changed. Additionally, there are many surrounding mutations that have occurred changing the context of the prediction. A known mutation that became predominate is D614G, this mutation isn't present in the reference sequence, but predominate in the training sequences. Thus when searching for novel mutations against the reference sequence, mutation D614G may be highest ranked as the fine-tuned model is more biased to this mutation. Consequently, the acquisition ranking for the mutations we are truly interested in (novel, unseen mutations) are decreased and the performance is lower.

Mutation fitness analysis

Our language model is purely predicting mutations based on the sequence data alone. There is no information given towards viral fitness of a mutation aside from indirectly by observing dominate mutations at a given position. As seen from the viral fitness correlation values, the language model outputs are more associated with RBD expression, a measure of protein stability, than ACE2 binding, which can be more variable due to the ACE2 receptor. While we analyzed correlation between ProtBERT calculations and variant fitness data, our main goal in this study was to determine what mutations are likely to happen in the future, not necessarily what the effect of those mutations will be. Variant effects are often epistatic, the accumulation of multiple substitutions often modify the effects of other mutations[49,53]. Because of this, when new mutations arise, it may be difficult to determine the resulting phenotype effects of any singular mutation. Thus when searching for novel mutations, it may be difficult to determine which mutations will have high clinically significance. Early variant detection models than can be useful to determine which mutations are likely to occur in the future, which can aid wet-lab experimentation to determine the functional effects of the mutation.

Applications for early variant detection system

In order to build a true early variant detection system, we utilize all novel $n = 987$ mutations that occurred after the cut-off date and had not been seen before in the training dataset. Due to the global pandemic, there was an unprecedented amount of genomic data captured for any virus[7]. With the high amount of genomic surveillance, it's likely that any single mutation recorded in our sequence database may have only occurred once or twice, suggesting it might have been a deleterious or largely neutral mutation. The

majority of our test set mutations occurred at low frequencies, with almost half of the test set occurring only once. While we don't have explicit fitness data on all mutations in our test set, we infer advantage by the frequency in our sequence database. In theory, the fitness advantage of a mutation is expressed by increased representation in a viral lineage[11]. While a mutation with high frequency may also be neutral, it's less likely for the mutation to be deleterious, as deleterious mutations are quickly lost. Thus we demonstrate predictive results against increasing threshold mutation test set. The increase in AUC from the threshold average compared to the global average suggests that overall all methods are more performant against mutations with potentially higher significance. With each step of the mutation frequency threshold, less mutations are considered, which highlights a weakness in some methods with variability in predictive performance for a few mutations. In particular, mutation N460K with frequency 3400 shows poor performance in grammaticality AUC. As grammaticality is a component of CSCS and DNMS, those methods also show a decrease in performance for this single mutation. However, attention change and semantic change AUC is high for N460K, which results in better performance for DNMS compared to CSCS. Together this shows that all three components (grammaticality, semantic change and attention change) are important for novel mutation predictions. Overall, grammaticality AUC is most variable when considering mutations at increasing thresholds. The posterior probability values may be variable for a few factors including lack of genetic variance in the training dataset at a given position. When looking at the entropy of positions in the training set *wrt* the performance of language model calculations at the same positions, see Supplementary Fig. 10, we do notice that at higher entropy levels (higher genetic diversity), grammaticality shows a higher performance trend. This indicates without observed changes in amino acids at any given position, the model may be too over-fit to the training data and will give lower probability to any given amino acid at a position. Whereas the overall global context of the sequence can still provide meaningful information towards the similarity calculations of semantic change and attention change. Due to the high predictive value of grammaticality overall the majority of the test set mutations, the overall average performance of DNMS is higher with giving higher weight to grammaticality in the calculation. However, considering single mutations such as N460K, it suggests a different weighting scheme may be beneficial to consider for some mutations.

Limitations and Future Direction

While our method has shown high predictive power for novel amino acid substitutions, it does not account for deletion and insertion mutations. In viral sequences, substitution mutations are the large majority of mutations compared to deletions and insertions[54]. In terms of a language model, a substitution is the equivalent of a token changing to a different token. For a deletion mutation, it could be modeled using a deletion token, "-"; however this isn't ideal for the ProtBERT model, which wasn't pretrained on sequences that contained this token. The deletion token doesn't contain inherent biological significance other than comparison to the reference sequence. The ProtBERT model can accurately determine patterns and semantic meaning from sequences of different lengths, thus the deletion token can be removed and two sequences of different lengths (with different deletion tokens) can be accurately compared without the explicit deletion token, by the surrounding sequence context alone. Without a deletion token, a deletion mutation is the equivalent as removing a token and shifting all tokens forward a single position. Similarly, an insertion mutation would be inserting a token and shifting all tokens backwards. While the methods of semantic change and attention change could be calculated similarly as for substitution mutations, modeling grammatically would be a more difficult task. Instead of considering the posterior probability of a single amino acid change at a given position, you would need to consider the posterior probability of amino acid changes of all tokens following the deletion/insertion which essentially shift the tokens at the affected positions. Ultimately, other methods such as generative protein models may be better suited to investigate deletion and insertion mutations.

Methods

Sequence data

SARS-CoV-2 nucleotide sequences are collected from NCBI Genebank up to January 11th, 2023. These are filtered for completed annotated sequences with non-ambiguous collection dates and a release date only after two months of the collection date. As the collection date is required for recording the first time a single mutation occurred, it is important to have high confidence in a sequence's collection date. Redundant nucleotide sequences (same per collection year and month) are removed.

Further filtering steps are performed including removing sequences with less than 29,000 bases and unknown nucleotide content greater than 0.05%. Additionally, sequences are filtered if the translated spike protein contained non-ambiguous amino acids. Lastly, Nextclade[55] is used to filter sequences for other quality issues. The end dataset contains 670,191 nucleotide sequences. Since this is an overwhelming amount of sequences and our concern is with the spike protein sequence, we further filter sequences to those that represent unique spike sequences (taking the first collected sequence). The end result is 35,943 sequences that represent unique spike proteins. These sequences and their collection date are shown in Fig. 3. The final 35,943 sequences are used to build our Nextstrain tree.

Of the final nucleotide dataset, 208,613 are pre-cutoff which represent 15,871 unique spike sequences. This set of sequences proved to be too many for fine-tuning the ProtBERT model. Thus we limit the training set to spike sequences that were observed at least twice in the data set. The end result is 6,256 unique spike sequences to fine-tune ProtBERT.

Variant fitness data

In order to analyze the relationship between the language model calculations and variant fitness, we obtain a DMS experimental dataset from Cao et al.[48]. ACE2 binding and RBD expression values are downloaded from https://github.com/jianfcpku/convergent_RBD_evolution; where we obtain binding and expression values for mutations in the RBD for three variants: reference sequence, BA.1 and BA.2.

To produce the language model calculations, for the reference we utilize Wuhan/Hu-1/2019. BA.1 (EPI_ISL_10000028) and BA.2 (EPI_ISL_10000005) are obtained from the GISAID database[56]. We use the Nextclade[55] tool to align the nucleotide sequences to the reference and save the resulting translated SARS-CoV-2 spike protein for inputs into the language model. From each of the three sequences, we calculate grammaticality, semantic change and attention change for all the RBD mutations that have corresponding experimental binding and/or expression values. In order to analyze the relationship between our language model calculations and the fitness values, we calculate the correlation and use a statistical test to determine significance.

Phylogenetic tree

Our phylogenetic tree is built with Nextstrain[47]. The Nextstrain data pipeline includes data filtering, alignment and masks certain positions that are known to show artifacts. A maximum likelihood phylogenetic tree is then built using IQ-Tree[57]; and refined using TimeTree[58]. In the refinement step; TimeTree infers ancestral sequences, resolves polytomies (when internal node is connected to more than three different nodes), and creates a time scaled phylogeny.

For the Nextstrain build, the 35,943 nucleotide sequences, which can be translated to unique spike protein sequences, are used to build the phylogenetic tree. Due to some filtering steps, the end tree has 35,818 leaf nodes (from sampled sequences) and 24,052 internal nodes. Supplementary Fig. 3 displays the full phylogenetic tree.

In the Nextstrain phylogenetic tree, internal nodes are inferred ancestral nodes, created using maximum likelihood methods. The leaf nodes are samples from the sequenced database. Internal nodes are "Parent" sequences of the "Child" leaf node. In DNMS, we mutate the parent sequences to predict novel mutations that are recorded in child leaf nodes. Node data from Nextstrain includes a list of mutations inherited from root (reference) to the node. In this way, we're able to re-create the parent nodes

by mutating the reference sequence with the given list of mutations. Deletion mutations are common, but our language models have not been trained on tokens to denote deletions (i.e. "-" token). Thus for deletion mutations, we simply remove the deleted amino acid from the re-constructed parent sequence.

Protein language model

Our method utilizes a pre-trained Bidirectional Encoder Representation from Transformers (BERT) model[44]. Pre-trained transformer models have many useful applications with fine-tuning to utilize transfer learning in downstream tasks. As previously shown with the Tasks Assessing Protein Embeddings (TAPE) collection of benchmark protein tasks, self-supervised pre-trained transformer models showed highest performance for a variety of protein prediction problems[59].

In this work, we use ProtBERT, previously pre-trained by Elnaggar et al. on UniRef100 database, which covers 216 million proteins from all areas of life[45]. UniRef100 clusters sequences from UniProt database at 100% sequence identity, ultimately removing duplicate sequences[60].

ProtBERT is a Masked Language Model (MLM). The MLM replaces a certain token with a [MASK] token and has an objective to recover the original token based on the full (left and right) context of the sequence. In contrast with a left-to-right model, such as a Long-Short-Term-Memory (LSTM) model, which attempts to determine a token given the left context. With Bi-directional LSTM (BiLSTM), the left-to-right context is concatenated with right-to-left context. However, these are still only trained with a single component of the context[44]. Thus BERT models have an advantage in that they are trained with the full context of the input sequence.

During pre-training 15% of tokens are randomly corrupted and the model is tasked to predict the selected tokens. If a token is selected for corruption, there's an 80% chance of being replaced with a [MASK] token; a 10% chance of replacing the token with a random token and 10% chance of keeping the token unchanged. Replacement with [MASK] token requires the transformer model to keep a contextual representation of every input token for the objective of recovering the original token. Replacing the [MASK] token with a random token (with a 10% chance) removes a potential mismatch between pre-training and fine-tuning tasks and utilizing the original token (with a 10% chance) allows a bias towards the true observed value[44].

ProtBERT was pre-trained for 300,000 steps for sequences with length $N = 512$ and then another 100,000 steps for sequences with length $N < 2000$. This allowed the model to learn useful representations from shorter sequence first for more efficient training on longer sequences later. Learning rate was set to 0.002 with weight decay of 0.01 and utilized the Lamb optimizer[45].

To fine-tune ProtBERT, we utilize the bio-transformers python wrapper (https://github.com/DeepChainBio/bio-transformers/). Fine-tuning allows us to use our training sequences to update model parameters with a bias towards SARS-CoV-2 sequences. If the pre-trained model is said to learn the global language of proteins over all areas of life, the fine-tuned model is analogous to learning the specific dialect of SARS-CoV-2 spike protein sequences. In this way, we're able to obtain a model that's learned the protein grammar rules and specific patterns unique to SARS-CoV-2 spike proteins.

During fine-tuning, 2.5% of tokens are randomly selected for corruption; with a 80% chance of being replaced with the [MASK] token and a 10% chance of being replaced with a random token. Fine-tuning utilized $n = 6, 256$ unique spike sequences that represent sequences which had frequency greater than 1 from the pre-cutoff sequence dataset. Training lasted for four epochs.

The ProtBERT model is shown in Supplementary Fig. 11. Input is a protein sequence of length N amino acids. The sequence is tokenized and encoded to feed into a 30 layer transformer model, each with 16 heads. Each head uses a self-attention mechanism[61]. Attention weights from pulled from each layer of the transformer model; ProtBERT has 30 layers and 16 heads, thus the attention weights have an initial size of (30, 16, N, N). The weights are max pooled to create an attention matrix of size (N,N). Individual hidden layers have a size of 1024 and the last hidden layer produces N amino acid embeddings of size 1024. These are concatenated to create a semantic protein embedding of size (N, 1024). For amino acid probability values, the amino acid embeddings are fed into a fully connected classification layer that utilizes the Gaussian Error Linear Unit (GELU) activation function, equation (1)[62], and normalizes outputs prior to a softmax layer to produce probabilities for all 20 amino acids at a given position.

$$\mathrm{GELU}\,(x) = x \cdot \mathbb{P}_{X\sim\mathcal{N}(0,1)}[X < x] \tag{1}$$

ESM model. To compare our ProtBERT model against other protein transformer models, we obtain pre-trained ESM models from Rives et al. 2019[52].

The first model, ESM1 t12 is a 12-layer transformer model, trained on UniRef50 representative sequences, which clusters UniProt at 50% sequence identity[52]. The second model, ESM1 t34 is a 34-layer transformer model trained on UniRef100[52], similarly to the pre-trained ProtBERT model.

Both models were fine-tuned with the same $n = 6, 256$ sequences as wth the ProtBERT model.

Formal problem formulation

Notation. After collecting and filtering SARS-CoV-2 nucleotide sequences, we obtain a nucleotide database $\mathcal{N}$. Nucleotide sequences are utilized to build the Nextstrain tree and are translated to create our spike protein sequence database, denoted by $\mathcal{S}$. As the nucleotide sequence encodes multiple other proteins, there may be a large number of duplicates of spike proteins in $\mathcal{S}$. Thus we only use unique spike proteins for $\mathcal{S}$, with taking the sequence of earliest collected date in the case of duplicates. For simplicity sake, the term "sequence" is used in this paper to denote a protein sequence, unless otherwise explicitly stated.

A protein sequence in sequence database $\mathcal{S}$ is denoted by $\mathbf{x}_k$, where k denotes the kth sequence. Protein sequences consists of a set of tokens x_i, $\mathbf{x}_k = (x_1, \ldots, x_N)$; where $x_i \in \Sigma$; Σ denotes the alphabet of the amino acid sequences; i denotes the ith token in $\mathbf{x}_k$, where $i \in [N]$ and N denotes the length of sequence $\mathbf{x}_k$.

The sequence Wuhan-Hu-1, Accession No. NC_045512.2[2], represents the first published sequence SARS-CoV-2 sequence and is referred to as the reference sequence and denoted by $\mathbf{x}_{\mathrm{ref}}$. Sequences $\mathbf{x}_k$ are mapped to $\mathbf{x}_{\mathrm{ref}}$ which becomes the root of the phylogenetic tree and a method to record variant mutations.

For any amino acid sequence $\mathbf{x}_k$, we use $\tilde{\mathbf{x}}_k[\tilde{x}_i]$ to denote a next strand mutated sequence of $\mathbf{x}_k$, mutated through mutation $\tilde{x}_i$; where $\tilde{\mathbf{x}}_k[\tilde{x}_i] = (\ldots, x_{i-1}, \tilde{x}_i, x_{i+1}, \ldots)$. Token $\tilde{x}_i$ denotes an amino acid substitution at position i from sequence $\mathbf{x}_k$ compared to sequence $\tilde{\mathbf{x}}_k$. $\mathbf{x}_k$ can either be the reference sequence, $\mathbf{x}_{\mathrm{ref}}$, or a sequence along the ancestry path from $\mathbf{x}_{\mathrm{ref}}$ to $\tilde{\mathbf{x}}_k$.

A single mutation, $\tilde{x}_i = [a : L : b]$, is denoted by a tuple with three values a L and b. a and b denotes the original and mutated amino acid respectively, and L denotes the actual location of the mutation in the sequence.

In practice, next strands often have a varying set of mutations, thus a mutant sequence of $\mathbf{x}_k$ is denoted by $\tilde{\mathbf{x}}_k$ where $\tilde{\mathbf{x}}_k = (\tilde{x}_1, \ldots, \tilde{x}_N)$ and the set of mutations are tokens in $\tilde{\mathbf{x}}_k$ that are in the same position as $\tilde{x}_k$ and disagree. The set of mutations between $\mathbf{x}_k$ and $\tilde{\mathbf{x}}_k$ are denoted by Eq. (2).

$$\mathcal{M}(\mathbf{x}_k, \tilde{\mathbf{x}}_k) = \{\tilde{x}_i | \tilde{x}_i \neq x_i\} \tag{2}$$

Given a sequence database $\mathcal{S}$, with sequences first aligned to reference sequence, $\mathbf{x}_{\mathrm{ref}}$, we use $\omega(\tilde{x}_i)$ to denote the frequency of mutation $\tilde{x}_i$. This indicates the number of times that a mutated sequence, $\tilde{\mathbf{x}}_k$ containing $\tilde{x}_i$, $(\tilde{\mathbf{x}}_k[\tilde{x}_i])$, appears in the sequence database and $\tilde{x}_i$ refers to mutated amino acid x at position i in $\mathbf{x}_{\mathrm{ref}}$. The frequency of a mutation, $\omega(\tilde{x}_i)$, can be

calculated as shown in Eq. (3).

$$\omega(\tilde{x}_i) = \sum_{\tilde{\mathbf{x}}_k[\tilde{x}_i] \in \mathcal{S}} 1 \tag{3}$$

For sequences in the database $\mathcal{S}$, their collection date t represents the date the sequence was sampled from the host. The collection dates are used to build training and test sets based on a cutoff point. For the entire data collected from a period t_0 to T, where T represents the last date a sampled is collected. The cutoff point is denoted by τ, such that $t_0 < \tau < T$. The pre-cutoff dataset, $\mathcal{S}_{\text{pre}}$ consists of samples that have a collection time point, t_k for the kth sample, before cutoff τ. The post-cutoff dataset, $\mathcal{S}_{\text{post}}$ contains samples with a collection time point after τ.

$$\mathcal{S}_{\text{pre}} = \{\mathbf{x}_k \in \mathcal{S} | t_0 \le t_k < \tau\}; \quad \mathcal{S}_{\text{post}} = \{\mathbf{x}_k \in \mathcal{S} | \tau \le t_k \le T\} \tag{4}$$

The test set mutations are denoted by $\mathcal{M}_{\text{test}}$. Mutations are recorded against the reference sequence, thus $\mathcal{M}_{\text{test}}$ represents the set of mutations that are variants of the reference sequence and are first recorded in sequences in $\mathcal{S}_{\text{post}}$ and not recorded in $\mathcal{S}_{\text{pre}}$, i.e. novel mutations.

$$\mathcal{M}_{\text{test}}(\mathbf{x}_{\text{ref}}, \mathcal{S}_{\text{post}}) = \{\tilde{x}_i | \tilde{x}_i \neq x_i; \quad \tilde{\mathbf{x}}_k[\tilde{x}_i] \in \mathcal{S}_{\text{post}}; \quad \tilde{\mathbf{x}}_k[\tilde{x}_i] \notin \mathcal{S}_{\text{pre}}\} \tag{5}$$

The sequence dataset denoted by $\mathcal{S}_{\text{test}}$, used for novel mutation prediction, contains parent sequences from the Nextstrain tree and a special case of the reference sequence. A parent sequence, $\mathbf{x}_k$, is counted in the set $\mathcal{S}_{\text{test}}$ if the next strand sequence (child sequence), $\tilde{\mathbf{x}}_k$ contains a mutation in the test set, $\mathcal{M}_{\text{test}}$. Thus the novel mutation is present in the child sequence, $\tilde{\mathbf{x}}_k$, but not parent sequence, $\mathbf{x}_k$.

$$\mathcal{S}_{\text{test}} = \{\mathbf{x}_k \in \mathcal{S} | \tilde{x}_i \in \mathcal{M}(\mathbf{x}_k, \tilde{\mathbf{x}}_k); \quad \tilde{x}_i \in \mathcal{M}_{\text{test}}\} \tag{6}$$

In order to build the set of sequences used to fine-tune ProtBERT, $\mathcal{S}_{\text{train}}$, we used a subset of $\mathcal{S}_{\text{pre}}$, where sequences have a frequency greater than 1 from the translated protein sequences in nucleotide sequence set $\mathcal{N}$. This represents more common and higher fittest members of pre-cutoff samples. For a given protein sequence, $\mathbf{x}_k$, let $\mathbf{g}_k$ represent the nucleotide (genomic) sequence in $\mathcal{N}$ that translates to (encodes) $\mathbf{x}_k$. The sequence frequency can be described with equation (7).

$$\omega(\mathbf{x}_k) = \sum_{\substack{\mathbf{g}_k \in \mathcal{N} \\ \mathbf{g}_k \text{ encodes } \mathbf{x}_k}} 1 \tag{7}$$

Thus our training dataset, $\mathcal{S}_{\text{train}}$, can be described with the following:

$$\mathcal{S}_{\text{train}} = \{\mathbf{x}_k \in \mathcal{S}_{\text{pre}} | \omega(\mathbf{x}_k) > 1\} \tag{8}$$

Problem statement. Our goal is to identify novel spike protein amino acid substitutions in $\mathcal{M}_{\text{test}}$. As our methods are intended for single point amino acid substitutions, insertion and deletion mutations are not a part of $\mathcal{M}_{\text{test}}$ or considered as ground truth mutations.

In order to identify novel mutations, we prioritize mutations if they have corresponding large posterior probabilities (grammatically acceptable tokens), and have similar similarity to the sequence of interest, as measured by semantic embedding and attention values.

In DNMS, a sequence $\mathbf{x}_k$ is selected for mutating if the next strand sequence $\tilde{\mathbf{x}}_k[\tilde{x}_i]$ has any mutation $\tilde{x}_i$ in $\mathcal{M}_{\text{test}}$. In practice, $\mathbf{x}_k$ may have multiple next strand sequences with different single mutations and/or a next strand that has multiple mutations. In either case, we consider all future single point amino acid mutations as the ground truth set that occur in next strand sequence(s) from $\mathbf{x}_k$.

Given a sequence $\mathbf{x}_k$ that we wish to predict mutations in $\mathcal{M}_{\text{test}}$, we mutate $\mathbf{x}_k$ in silico; we build the candidate set of mutations for prediction, $\mathcal{P}(\mathbf{x}_k)$, that contain every single point amino acid substitution for all positions $i = 1$ to N, the length of the sequence.

$$\mathcal{P}(\mathbf{x}_k) = \{\tilde{x}_i | x_i \in \mathbf{x}_k; \quad \tilde{x}_i \neq x_i; \quad x \in \Sigma\} \tag{9}$$

For each mutation in $\mathcal{P}(\mathbf{x}_k)$, we create a mutated sequence $\mathbf{x}_k[\tilde{x}_i]$ that contains the single point amino acid substitution $\tilde{x}_i$, such that $\tilde{\mathbf{x}}_k[\tilde{x}_i] = (x_1, \ldots, x_{i-1}, \tilde{x}_i, x_{i+1}, \ldots, x_N)$. The following section defines the language model calculations used for prediction.

Language model calculations. For the sequence of interest, $\mathbf{x}_k$, we first require a vector representation of all tokens x_i from $i = 1$ to N, denoted as e_i, to create X_0, which represents the input, in a vectorized matrix, to the first layer in the transformer model.

$$X_0 = \{e_1, e_2, \ldots, e_N\}$$

A transformer model consists of L layers, where the input to the layer ℓ, X_ℓ is the output of the previous layer (or X_0 if $\ell = 1$).

$$X_\ell = f_{\ell-1}(X_{\ell-1})$$

The function f_ℓ represents the layer ℓ transformation on input X. Layer transformations are accomplished using a multi-headed attention function (`MHAttention`) which utilizes $\mathcal{W}_\ell$, the collection of attention head weights for layer ℓ. A layer's transformation is shown in equation (10). The final output of layer ℓ additionally includes layer normalization and the GELU activation function, equation (1). For a more in-depth discussion and useful pseudocode on transformer models, please see ref. 62.

$$f_\ell(X) = X + \texttt{MHAttention}(X | \mathcal{W}_\ell) \tag{10}$$

A single attention head calculates Query, $\mathbf{Q}$, Key, $\mathbf{K}$ and $\mathbf{V}$ matrices from input X and input from previous layer $X_{\ell-1}$ as described in equation (11). Where $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_v$ are the weight matrices for query, key and value respectively and b_q, b_k, b_v are bias terms for query, key and value respectively.

$$\mathbf{Q} = \mathbf{W}_q X + b_q; \quad \mathbf{K} = \mathbf{W}_k X_{\ell-1} + b_k; \quad \mathbf{V} = \mathbf{W}_v X_{\ell-1} + b_v; \tag{11}$$

A query vector in query matrix $\mathbf{Q}$ maps a given token in input X, while the key and value vectors map the surrounding tokens in context of the input. The product, $\mathbf{QK}^T$ represents the contribution of each token to each other token in the sequence. The output of the attention head is an updated value matrix, $\mathbf{V}'$, shown in equation (12), where d_k is the dimension of key factors. For a multi-headed attention mechanism, this process is repeated H times, where H is the number of heads. Each head has an output that is concatenated together and multiplied by another weight matrix to obtain the output of the multi-head attention function.

$$\mathbf{V}' = \texttt{softmax}\left(\frac{\mathbf{QK}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{12}$$

The dot product of $\mathbf{W}_q$ and $\mathbf{W}_k$, query and key weights respectively, are also known as the score weights, the strength of the weights measure the relevance of each key-value pair to the query. In our model $\mathbf{W}_\ell^h$ represents the score weights of the hth attention head in layer ℓ. A single $\mathbf{W}_\ell^h$ matrix is size (N, N) and normalized row-wise such that each row adds to 1.

For the 30 layer transformer, each with 16 attention heads, we extract the matrix $\mathcal{W}, \mathcal{W} \in \mathbb{R}^{30 \times 16 \times N \times N}$. For a long sequence, individual attention weights may be very small, as each row is normalized to add to 1 in each individual $\mathbf{W}_\ell^h$ matrix. Thus we use max-max pooling to obtain attention matrix $\mathbf{A}$, $\mathbf{A} \in \mathbb{R}^{N \times N}$. Where cell $A_{i,j}$ is the maximum attention weight (and importance) the ith token had to the representation of the jth token for all layers.

$$\mathbf{A} = \max_{\ell=1}^{30} \max_{h=1}^{16} \mathcal{W}_\ell^h$$

For the sequence of interest, $\mathbf{x}_k$, $\mathbf{A}_k$ represents the attention matrix after passing input $\mathbf{x}_k$. For all mutated sequences, $\tilde{\mathbf{x}}_k[\tilde{x}_i]$ in $\mathcal{P}(\mathbf{x}_k)$, $\mathbf{A}[\tilde{x}_i]$ denotes the attention matrix after passing input $\tilde{\mathbf{x}}_k[\tilde{x}_i]$ to the ProtBERT model. We calculate the change in attention with equation (13), denoted by $\Delta\mathbf{A}[\tilde{x}_i]$, which represents the similarity in attention matrices calculated with the ℓ_2 norm.

$$\Delta\mathbf{A}[\tilde{x}_i] = \| \mathbf{A} - \mathbf{A}[\tilde{x}_i] \|_2 \tag{13}$$

Conceptually, $\Delta\mathbf{A}[\tilde{x}_i]$ represents a degree of similarity in how the model is "paying attention" to a sequence and a potential mutation of the sequence. Similar attention weights between two sequences indicate they share similar patterns. As our objective is to find the most likely future mutation, we desire this weight to be small in determining future mutations.

The output of the last layer in the transformer model produces N amino acid embedding vectors, each of size 1024, the hidden layer size in the model. These values are concatenated to create $\mathbf{Z}$, $\mathbf{Z} \in \mathbb{R}^{N \times 1024}$, which denotes the protein embedding. For simplicity, we refer to the full model (including tokenizing and vectorizing input $\mathbf{x}_k$) as $f_s(\mathbf{x}_k)$. Thus for the sequence of interest we're mutating, $\mathbf{Z} = f_s(\mathbf{x}_k)$, is the resulting protein embedding of the input sequence $\mathbf{x}_k$. For all mutated sequences $\tilde{\mathbf{x}}_k[\tilde{x}_i]$ in $\mathcal{P}(\mathbf{x}_k)$, $\mathbf{Z}[\tilde{x}_i]$ denotes the mutated protein embedding after passing input $\tilde{\mathbf{x}}_k[\tilde{x}_i]$ to the ProtBERT model, $\mathbf{Z}[\tilde{x}_i] = f_s(\mathbf{x}_k[\tilde{x}_i])$.

$\mathbf{Z}$ is a numerical embedding of the input sequence $\mathbf{x}_k$ and represents the protein semantics. In our goal to find likely future mutations in sequence $\mathbf{x}_k$, we calculate the semantic change, $\Delta\mathbf{Z}[\tilde{x}_i]$ with equation (14) which represents similarity in semantics from the sequence of interest to a given mutated sequence with the ℓ_2 norm.

$$\Delta\mathbf{Z}[\tilde{x}_i] = \| \mathbf{Z} - \mathbf{Z}[\tilde{x}_i] \|_2 = \| f_s(\mathbf{x}_k) - f_s(\mathbf{x}_k[\tilde{x}_i]) \|_2 \tag{14}$$

The output of the transformer model, $\mathbf{Z}$, is then fed into a final classification layer with weights $\mathbf{W}_f$ and bias $\mathbf{b}_f$ which uses the GELU activation function, equation (1), this creates classification layer output, X_c.

$$X_c = \texttt{GELU}(\mathbf{W}_f \mathbf{Z} + \mathbf{b}_f)$$

The classification layer output, X_c is normalized and fed into a final softmax layer with un-embedding weights, $\mathbf{W}_u$. The final output is a probability distribution, $\mathbf{P}$; which consists of probability values for tokens in the alphabet, with length N_Σ over the length of input sequence, $\mathbf{x}_k$ with length N, $\mathbf{P} \in (0, 1)^{N_\Sigma \times N}$.

$$\mathbf{P} = \texttt{softmax}(\mathbf{W}_u X_c)$$

For a given token x at the ith position, $p(x_i|\mathbf{x}_k)$ denotes the posterior probability of the token at that position given input sequence $\mathbf{x}_k$. Probabilities are calculated using forward mode, where a single forward pass of the input sequence is fed into the transformer model to output posterior probabilities.

$$p(x_i|\mathbf{x}_k) = p(x_i|x_1, \ldots, x_i, \ldots, x_N) \tag{15}$$

For all mutations in $\mathcal{P}(\mathbf{x}_k)$, probability values are calculated by inputting sequence $\mathbf{x}_k$. The grammaticality of the mutation, $p(\tilde{x}_i|\mathbf{x}_k)$ represents the probability of mutated amino acid $\tilde{x}$ at the ith position. This value represents the likelihood of the amino acid at the position and represents how well the mutation confers to the grammar rules of the protein learned by the language model.

CSCS: constrained semantic change search. In this section, we discuss the influential method, Constrained Semantic Change Search (CSCS) introduced by Hie et al.[28] In the original paper, they utilized a biLSTM model trained on SARS-CoV-2 spike protein and homologous Betacoronavirus sequences.

The original published CSCS method is shown in equation (16), with another minor difference of using the ℓ_1 norm in the semantic change calculation, (equation (14)).

$$CSCS'(\tilde{x}_i; \mathbf{x}_k) = \texttt{rank}(\Delta\mathbf{Z}[\tilde{x}_i]) + \beta \times \texttt{rank}(p(\tilde{x}_i|\mathbf{x}_k)) \tag{16}$$

Where β is a weighting parameter, in their study and results discussed in Section 2.5, $\beta = 1$.

From preliminary results discussed in the paper, it was found this ranking method isn't as performant for novel mutation search, thus we adjust the ranking in an adjusted CSCS calculation. The function $\texttt{rank}(x)$ ranks the items in array x from highest to lowest. We use $\texttt{rank}(-x)$ to demonstrate a ranking of items in array x from lowest to highest.

To do a comparison against CSCS and DNMS, we use an adjusted version of CSCS, defined in equation (17), where we use the ℓ_2 norm in semantic change calculation, and adjust the ranking scheme to best fit our objective. Additionally we add α weighting parameter for semantic change calculation. In results comparison of DNMS and CSCS, the adjusted version in equation (17) is utilized.

$$CSCS(\tilde{x}_i; \mathbf{x}_k) = \alpha \times \texttt{rank}(-\Delta\mathbf{Z}[\tilde{x}_i]) + \beta \times \texttt{rank}(p(\tilde{x}_i|\mathbf{x}_k)) \tag{17}$$

DNMS: deep novel mutation search. DNMS is formally defined in equation (18), where we combine grammaticality, semantic change and attention change rankings.

$$\begin{aligned} DNMS(\tilde{x}_i; \mathbf{x}_k) = {} & \alpha \times \texttt{rank}(-\Delta\mathbf{Z}[\tilde{x}_i]) + \beta \times \texttt{rank}(p(\tilde{x}_i|\mathbf{x}_k)) \\ & + \gamma \times \texttt{rank}(-\Delta\mathbf{A}[\tilde{x}_i]) \end{aligned} \tag{18}$$

Where α, β, and γ are weighting parameters for Semantic Change, Grammaticality and Attention Change respectively. To determine what values are best fitting for the weighting parameters, we performed a grid search and determined that $\alpha = 1.5$, $\beta = 3.0$ and $\gamma = 1.0$ show the best results for DNMS. The same α and β values are used for CSCS, equation (17) for a fair comparison.

Performance calculations. All possible mutations, $\tilde{x}_i \in \mathcal{P}(\mathbf{x}_k)$, are given an acquisition ranking, based on $DNMS(\tilde{x}_i; \mathbf{x}_k)$, a_{DNMS}, where the highest scores of $DNMS(\tilde{x}_i; \mathbf{x}_k)$ are given priority and identified as the most likely candidates for future novel mutations.

$$a_{DNMS}(\tilde{x}_i; \mathbf{x}_k) = \texttt{rank}(DNMS(\tilde{x}_i; \mathbf{x}_k)) \tag{19}$$

Similarly for CSCS, we calculate the acquisition ranking based on $CSCS(\tilde{x}_i; \mathbf{x}_k)$.

$$a_{CSCS}(\tilde{x}_i; \mathbf{x}_k) = \texttt{rank}(CSCS(\tilde{x}_i; \mathbf{x}_k)) \tag{20}$$

For grammaticality, we consider the rank of probability values alone for acquisition ranking, as shown in equation (21).

$$a_{gram}(\tilde{x}_i; \mathbf{x}_k) = \texttt{rank}(p(\tilde{x}_i|\mathbf{x}_k)) \tag{21}$$

Similarly for semantic change and attention change, equations (22) and (23) respectively, we consider the ranking for those values separately, but smaller values are prioritized and ranked higher than higher values. Note these ranking values are all components of $DNMS(\tilde{x}_i; \mathbf{x}_k)$, without the weighting parameters.

$$a_{sem}(\tilde{x}_i; \mathbf{x}_k) = \texttt{rank}(-\Delta\mathbf{Z}[\tilde{x}_i]) \tag{22}$$

$$a_{attn}(\tilde{x}_i; \mathbf{x}_k) = \texttt{rank}(-\Delta\mathbf{A}[\tilde{x}_i]) \tag{23}$$

For each method, all mutations $\tilde{x}_i \in \mathcal{P}(\mathbf{x}_k)$ are given a corresponding acquisition ranking score, a_i for range $i = 1$ to M, where M represents length of possible mutation set, $\mathcal{P}(\mathbf{x}_k)$; $M = |\mathcal{P}(\mathbf{x}_k)|$. This is the same as $N \times 19$; where N denotes the length of $(\mathbf{x}_k)$ and 19 represents 19 potential amino acid

mutations, $\tilde{x}_i$, for the ith position in $(\mathbf{x}_k)$. Acquisition rankings range from 1 to M, where the highest value represents the most likely mutations as specified by the method and 1 represents the least likely mutations. Starting from 1 to M, increasing thresholds are created. At each threshold, we regard all mutations with rank lower than the threshold as positive class (future mutation) and compare these to the ground truth future novel mutations, thus we're able to calculate a true positive rate and a false positive rate. Plotting the false positive rate *vs.* true positive rate, we can achieve the receiver operating characteristic (ROC) curve. The AUC is the area under the ROC curve. For AUC values, if the acquisition ranking is perfect, all positive class samples (future mutations) will be ranked higher than the negative class and AUC will equal 1. Deviations from this ranking decrease AUC; and AUC value of 0.5 indicates random ranking.

Statistics and reproducibility

In order to determine if methods differ with statistical significance, a Friedman test is done comparing the methods results first on different parent sequences, and second on average AUC scores over different mutation frequency thresholds.

A Friedman test is a non-parametric version of the repeated-measures ANOVA[63]. The test considers ranks for each data set (different parent sequence or different mutation frequency threshold results) separately, methods are ranked from highest AUC to lowest AUC.

For k methods tested on n datasets, equation (24) determines the average ranks of the jth method, where r_i^j is the rank of the jth method on the ith dataset.

$$R_j = \frac{1}{n}\sum_{i=1}^{n} r_i^j \tag{24}$$

The null hypothesis states that there is no difference between algorithms, thus their average ranks (R_j) over different datasets will not be different[63]. Thus the Friedman statistic„ χ_F^2 is shown in equation (25).

$$\chi_F^2 = \frac{12n}{k(k+1)}\left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{25}$$

In our analysis, we compare $k = 5$ different methods first on $n = 359$ separate datasets (parent sequences) then $n = 32$ mutation frequency threshold averages.

After rejecting the null-hypothesis, that methods are equivalent, we continue to perform a Nemenyi post-hoc test for comparing all methods to each other. Two methods are significantly different if the average ranks differ by at least the critical difference (CD), as defined in equation (26). Where q_α is the Studentized range statistic, with $k = 5$ methods and $\alpha = 0.05$, $q_\alpha = 2.728$[63].

$$CD = q_\alpha\sqrt{\frac{k(k+1)}{6n}} \tag{26}$$

To display the results from the Nemenyi post-hoc test, we display Critical Difference diagrams in Fig.9. Where methods are ranked in descending order of performance with methods on the left (closer to 1) having higher performance. Two methods are statistically significantly different if their average ranks differ by at least the CD value. Methods that are not statistically significantly different are grouped together with a bar.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

Data used in analysis can be downloaded as instructed in our code repository: https://github.com/maggieelkin/CovMutation. We provide the required data to reproduce analysis and figures as published. Source data for figures and analysis is also provided in Supplementary Data 1–8.

We also make use of publicly available databases and tools:

- NCBI Virus for SARS-CoV-2 Nucleotide sequences: https://www.ncbi.nlm.nih.gov/labs/virus/vssi/#/
- Nextclade CLI for alignment and mutation information[55]: https://docs.nextstrain.org/projects/nextclade/en/stable/index.html
- Grammaticality and Semantic Change values for mutations against the reference sequence from biLSTM language model trained by Hie et al.[28] downloaded from: https://github.com/brianhie/viral-mutation
- Fitness single-residue DMS of Spike RBD from Cao et al.[48] downloaded from: https://github.com/jianfcpku/convergent_RBD_evolution and also available at https://github.com/maggieelkin/CovMutation.

Code availability

All code used in analysis is written in Python and available in our code repository: https://github.com/maggieelkin/CovMutationand on Zenodo at https://doi.org/10.5281/zenodo.14015344[64]. Where we also provide links to raw and processed data and our fine-tuned ProtBert model. We provide examples of usage to reproduce analysis and figures as published.

Received: 19 May 2024; Accepted: 13 November 2024;
Published online: 21 January 2025

References

1. Wang, H. et al. The genetic sequence, origin, and diagnosis of SARS-CoV-2. *Eur. J. Clin. Microbiol. Infectious Dis.* 1–7 (2020).
2. Wu, F. et al. A new coronavirus associated with human respiratory disease in China. *Nature* **579**, 265–269 (2020).
3. Wang, H., Pipes, L. & Nielsen, R. Synonymous mutations and the molecular evolution of SARS-CoV-2 origins. *Virus Evol.* **7**, veaa098 (2021).
4. Sanjuán, R. & Domingo-Calap, P. Genetic diversity and evolution of viral populations. *Encyclopedia of Virology* 53–61 https://doi.org/10.1016/B978-0-12-809633-8.20958-8 (2021).
5. Otto, S. P. et al. The origins and potential future of SARS-CoV-2 variants of concern in the evolving COVID-19 pandemic. *Curr. Biol.* **31**, R918–R929 (2021).
6. Groves, D. C., Rowland-Jones, S. L. & Angyal, A. The D614G mutations in the SARS-CoV-2 spike protein: Implications for viral infectivity, disease severity and vaccine design. *Biochem. Biophys. Res. Commun.* **538**, 104–107 (2021).
7. Markov, P. V. et al. The evolution of SARS-CoV-2. *Nat. Rev. Microbiol.* **21**, 361–379 (2023).
8. Jacob, J. J. et al. Evolutionary tracking of SARS-CoV-2 genetic variants highlights an intricate balance of stabilizing and destabilizing mutations. *mBio* **12**, e0118821 (2021).
9. Kryazhimskiy, S. & Plotkin, J. B. The population genetics of dN/dS. *PLoS Genet.* **4**, e1000304 (2008).
10. Kim, K. et al. The roles of APOBEC-mediated RNA editing in SARS-CoV-2 mutations, replication and fitness. *Sci. Rep.* **12**, 14972 (2022).
11. Zahradník, J., Nunvar, J. & Schreiber, G. Perspectives: SARS-CoV-2 spike convergent evolution as a guide to explore adaptive advantage. *Front. Cell. Infect. Microbiol.* **12**, 748948 (2022).
12. Shang, J. et al. Cell entry mechanisms of SARS-CoV-2. *Proc. Natl Acad. Sci.* **117**, 11727–11734 (2020).
13. Harvey, W. T. et al. SARS-CoV-2 variants, spike mutations and immune escape. *Nat. Rev. Microbiol.* **19**, 409–424 (2021).
14. Galloway, S. E. et al. Emergence of SARS-CoV-2 B.1.1.7 Lineage—United States, December 29, 2020–January 12, 2021. *Mmwr. Morbidity Mortal. Wkly. Rep.* **70**, 95–99 (2021).
15. Focosi, D., Quiroga, R., McConnell, S., Johnson, M. C. & Casadevall, A. Convergent evolution in SARS-CoV-2 spike creates a variant soup

from which new COVID-19 waves emerge. *Int. J. Mol. Sci.* **24**, 2264 (2023).
16. Maher, M. C. et al. Predicting the mutational drivers of future SARS-CoV-2 variants of concern. *Sci. Transl. Med.* **14**, eabk3445 (2022).
17. Lässig, M., Mustonen, V. & Walczak, A. M. Predicting evolution. *Nat. Ecol. Evol.* **1**, 0077 (2017).
18. Wang, Y., Yadav, P., Magar, R. & Farimani, A. B. Bio-informed protein sequence generation for multi-class virus mutation prediction. *bioRxiv* https://doi.org/10.1101/2020.06.11.146167 (2020).
19. Fowler, D. M. & Fields, S. Deep mutational scanning: a new style of protein science. *Nat. methods* **11**, 801–807 (2014).
20. Narayanan, K. K. & Procko, E. Deep mutational scanning of viral glycoproteins and their host receptors. *Front. Mol. Biosci.* **8**, 636660 (2021).
21. Dey, T., Chatterjee, S., Manna, S., Nandy, A. & Basak, S. C. Identification and computational analysis of mutations in SARS-CoV-2. *Comput. Biol. Med.* **129**, 104166 (2021).
22. Obermeyer, F. et al. Analysis of 6.4 million SARS-CoV-2 genomes identifies mutations associated with fitness. *Science* **376**, 1327–1332 (2022).
23. Kepler, L., Hamins-Puertolas, M. & Rasmussen, D. A. Decomposing the sources of SARS-CoV-2 fitness variation in the United States. *Virus Evol.* **7**, veab073 (2021).
24. Beguir, K. et al. Early Computational detection of potential high risk SARS-CoV-2 variants. *Comput. Biol. Med.* 155, 106618 (2023).
25. de Hoffer, A. et al. Variant-driven early warning via unsupervised machine learning analysis of spike protein mutations for COVID-19. *Sci. Rep.* **12**, 9275 (2022).
26. Zhou, B. et al. TEMPO: A transformer-based mutation prediction framework for SARS-CoV-2 evolution. *Comput. Biol. Med.* **152**, 106264 (2023).
27. Rodriguez-Rivas, J., Croce, G., Muscat, M. & Weigt, M. Epistatic models predict mutable sites in SARS-CoV-2 proteins and epitopes. *Proc. Natl Acad. Sci.* **119**, e2113118119 (2022).
28. Hie, B., Zhong, E. D., Berger, B. & Bryson, B. Learning the language of viral evolution and escape. *Science* **371**, 284–288 (2021).
29. Pathan, R. K., Biswas, M. & Khandaker, M. U. Time series prediction of COVID-19 by mutation rate analysis using recurrent neural network-based LSTM model. *Chaos Solitons Fractals* **138**, 110018 (2020).
30. Rong, M. L. K., Kuruoglu, E. E. & Chan, W. K. V. Modeling SARS-CoV-2 nucleotide mutations as a stochastic process. *PLoS ONE* **18**, e0284874 (2023).
31. Nawaz, M. S., Fournier-Viger, P., Shojaee, A. & Fujita, H. Using artificial intelligence techniques for COVID-19 genome analysis. *Appl. Intell.* **51**, 3086–3103 (2021).
32. Darooneh, A. H., Przedborski, M. & Kohandel, M. A novel statistical method predicts mutability of the genomic segments of the SARS-CoV-2 virus. *QRB Discov.* **3**, e1 (2022).
33. Saldivar-Espinoza, B. et al. Prediction of recurrent mutations in SARS-CoV-2 using artificial neural networks. *Int. J. Mol. Sci.* **23**, 14683 (2022).
34. Ofer, D., Brandes, N. & Linial, M. The language of proteins: NLP, machine learning & protein sequences. *Comput. Struct. Biotechnol. J.* **19**, 1750–1758 (2021).
35. Bepler, T. & Berger, B. Learning the protein language: Evolution, structure, and function. *Cell Syst.* **12**, 654–669.e3 (2021).
36. Jumper, J. et al. Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
37. Outeiral, C. & Deane, C. M. Codon language embeddings provide strong signals for use in protein engineering. *Nat. Mach. Intell.* **6**, 170–179 (2024).
38. Wu, Z., Johnston, K. E., Arnold, F. H. & Yang, K. K. Protein sequence design with deep generative models. *Curr. Opin. Chem. Biol.* **65**, 18–27 (2021).
39. Shin, J.-E. et al. Protein design and variant prediction using autoregressive generative models. *Nat. Commun.* **12**, 2403 (2021).
40. Meier, J. et al. *Language Models Enable Zero-shot Prediction of the Effects of Mutations on Protein Function*, Vol. 34, 29287–29303 (Curran Associates, Inc., 2021).
41. Marquet, C. et al. Embeddings from protein language models predict conservation and variant effects. *Human Genet.* **141**, 1629–1647 (2021).
42. Riesselman, A. J., Ingraham, J. B. & Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nat. Methods* **15**, 816–822 (2018).
43. Jiang, T., Fang, L. & Wang, K. Deciphering the Language of Nature: a transformer-based language model for deleterious mutations in proteins. *Innovation* **4**, 100487 (2023).
44. Kenton, J. D. M.-W. C. & Toutanova, L. K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT 2019* (eds Burstein, J. et al.) **1**, 4171–4186 (Association for Computational Linguistics, 2019).
45. Elnaggar, A. et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**, 7112–7127 (2022).
46. Vig, J. et al. BERTology meets biology: interpreting attention in protein language models. Preprint at https://arxiv.org/abs/2006.15222 (2021).
47. Hadfield, J. et al. Nextstrain: Real-time tracking of pathogen evolution. *Bioinformatics* **34**, 4121–4123 (2018).
48. Cao, Y. et al. Imprinted SARS-CoV-2 humoral immunity induces convergent Omicron RBD evolution. *Nature* **614**, 521–529 (2023).
49. Starr, T. N. et al. Deep mutational scans for ACE2 binding, RBD expression, and antibody escape in the SARS-CoV-2 Omicron BA.1 and BA.2 receptor-binding domains. *PLOS Pathog.* **18**, e1010951 (2022).
50. Starr, T. N. et al. Deep mutational scanning of SARS-CoV-2 receptor binding domain reveals constraints on folding and ACE2 binding. *Cell* **182**, 1295–1310.e20 (2020).
51. Rambaut, A. et al. A dynamic nomenclature proposal for SARS-CoV-2 lineages to assist genomic epidemiology. *Nat. Microbiol.* **5**, 1403–1407 (2020).
52. Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl Acad. Sci.* **118**, e2016239118 (2021).
53. Starr, T. N. et al. Shifting mutational constraints in the SARS-CoV-2 receptor-binding domain during viral evolution. *Science* **377**, 420–424 (2022).
54. Sanjuán, R., Nebot, M. R., Chirico, N., Mansky, L. M. & Belshaw, R. Viral Mutation Rates. *J. Virol.* **84**, 9733–9748 (2010).
55. Aksamentov, I., Roemer, C., Hodcroft, E. B. & Neher, R. A. Nextclade: Clade assignment, mutation calling and quality control for viral genomes. *J. Open Source Softw.* **6**, 3773 (2021).
56. Shu, Y. & McCauley, J. GISAID: Global initiative on sharing all influenza data—from vision to reality. *Eurosurveillance* **22**, 30494 (2017).
57. Nguyen, L.-T., Schmidt, H. A., von Haeseler, A. & Minh, B. Q. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Mol. Biol. Evol.* **32**, 268–274 (2015).
58. Sagulenko, P., Puller, V. & Neher, R. A. TreeTime: Maximum-likelihood phylodynamic analysis. *Virus Evol.* **4**, vex042 (2018).
59. Rao, R. et al. Evaluating protein transfer learning with TAPE. *Adv. Neural Inf. Process. Syst.* **32**, 9689–9701 (2019).
60. Consortium, T. U. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.* **47**, D506–D515 (2018).
61. Vaswani, A. et al. Attention is all you need. In *Adv. Neural Inf. Process. Syst.* **30** (Curran Associates, Inc., 2017).
62. Phuong, M. & Hutter, M. Formal algorithms for transformers. Preprint at https://arxiv.org/abs/2207.09238 (2022).

63. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**, 1–30 (2006).
64. Elkin, M. DNMS: Deep novel mutation search. https://doi.org/10.5281/zenodo.14015344 (2024).

Acknowledgements

This work is partially sponsored by the U.S. National Science Foundation under grant Nos. IIS-2236579, IIS-2302786, and IOS-2430224.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42003-024-07262-7.

Correspondence and requests for materials should be addressed to Magdalyn E. Elkin or Xingquan Zhu.

Peer review information *Communications Biology* thanks Santiago Garcia Vallve and Amalio Telenti for their contribution to the peer review of this work. Primary Handling Editors: Laura Rodríguez Prrez and Joao Manuel de Sousa Valente. A peer review file is available.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025