CovMutEx: An Extensible Software Framework for Exploring SARS-CoV-2 Genome-Wide Mutation Probabilities

Anthony Yua Ior[1], Ali Kerem Yildiz[1], Huzeyfe Ayaz[2], and Ali Cakmak[1]*
[1]Istanbul Technical University, Dept. of Computer Eng., Maslak 34469 Istanbul, Türkiye
[2] Technical University of Munich, Dept. of Computer Eng., Arcisstraße 21, 80333 München, Germany

*Corresponding author: Ali Cakmak, ali.cakmak@itu.edu.tr

Abstract—As SARS-CoV-2 continues to evolve, there is a critical need for integrated bioinformatics platforms that bridge the gap between complex mutation forecasting and real-time genomic surveillance. We present CovMutEx (COVID-19 Mutation Explorer), an interactive web-based software framework designed for the genome-wide exploration of mutation probabilities. Unlike static trackers, CovMutEx offers a model-agnostic architecture that integrates deep learning methods, including the state-of-the-art PRIEST model, within a modular system to analyze user-specified lineages. The platform features a dedicated Variant Hotspot Explorer, which provides a quantitative framework for validating model-derived signals against documented ground truth for notable Variants of Interest. By reformulating variant interpretation as a lineage-conditioned ranking problem, the tool natively computes Precision, Recall, and F1-scores, supporting both positional fidelity and regional biological relevance. Our benchmarks demonstrate that the system maintains stable rendering performance across the 30 kb genome through efficient data decimation and list virtualization. Released as open-source software, CovMutEx provides researchers with a transparent and extensible hub to build, test, and visualize the inferences by the integrated predictive models, thereby enhancing preparedness for the continuing pandemic.

Index Terms—SARS-CoV-2, mutation prediction, genome visualization, interactive analytics

INTRODUCTION

THE worldwide spread of SARS-CoV-2 has shown the need for ongoing surveillance and timely public health responses. Vaccines remain central to prevention efforts, but the virus mutates rapidly [1], which can reduce vaccine effectiveness over time [2, 3]. Because the virus keeps changing over time [4], vaccine development is difficult, since vaccine designs must keep up with a target that does not stay the same. There is an urgent need for computational tools that can offer early, actionable insight into likely mutation trends and support current clinical decision-making. Prior studies have expanded what is known about viruses, including core transmission dynamics, estimated nucleotide substitution rates, and broader evolutionary patterns reported during the pandemic [5, 6, 7]. The literature has documented that a range of computational methods has changed our capacity to combat the virus [8]. Specialized bioinformatics tools are now central to the analysis of SARS-CoV-2 molecular and genomic profiles [9, 10]. By scanning genomic sequences for recurring mutations [11], these tools help track how the virus has changed over time [4]. The integration of new algorithm methods and data analysis techniques has enabled researchers to predict the severity and clinical outcomes of emerging variants [12], which can help public health officials plan for future viral threats. Developing and applying these computational tools has reduced the time needed to respond to the SARS-CoV-2 crisis [13]. These tools serve as early warning systems by flagging potential mutations before they become widespread. They support the development of targeted medical interventions and inform decisions about how to distribute scarce healthcare resources [3]. They help clarify how the virus changes over time and provide a basis for targeted containment and mitigation measures [14]. Because the virus mutates over time, researchers need visualization tools that allow to explore future genetic changes interactively and provide information to support informed public health decisions. Although online resources have expanded, there is still no platform designed specifically to interactively explore and visualize model-derived mutation probabilities across the full SARS-CoV-2 genome. This paper presents CovMutEx (COVID-19 Mutation Explorer), an interactive web-based tool developed to address this gap. It provides a unified framework for researchers to examine and compare potential mutation patterns over time. Rather than proposing a novel predictive algorithm, we directly integrate pre-trained deep learning models from the literature into CovMutex,including sequence-conditioned models [15] and the state-of-the-art prevalence-anchored PRIEST model [16], into a unified exploratory framework. The platform offers an interface that most users can operate with little training. Researchers can set key parameters, including the variant being studied, the deep learning model used for inference, and the time elapsed since the variant first appeared. After these parameters are defined, the system starts an automated feature extraction step that uses the chosen variant sequence together with baseline information

from the reference genome [17]. The tool uses k-merization with a sliding window approach to produce local feature representations. The features are processed in advance and then passed to pre-trained models, which produce position-specific mutation probabilities that can be displayed in real time. The results are presented as interactive charts that support close inspection at the residue level. This feature allows researchers to identify protein regions that are likely to mutate and to evaluate the broader biological implications of those changes. Drawing on existing genomic research, CovMutEx seeks to address a practical gap: allowing researchers to visually investigate how future viral mutations may affect vaccine effectiveness. Traditional genome browsers offer extensive information, but they often do not include integrated prediction models required for current surveillance work. We devoted substantial engineering work to optimize the system architecture for high-density visualization and real-time inference. By implementing list virtualization and data decimation strategies, the platform maintains a highly responsive user experience even when analyzing large-scale genomic datasets. The data preprocessing and feature extraction steps are set up to keep response times short. We also use dynamic data loading so that the client does less computation when it needs to render complex visualizations [18]. It is important to clarify that this study does not aim to present one best-performing predictive model. CovMutEx is built as a flexible, expandable, interactive system that supports the study of possible future changes in the SARS-CoV-2 genome through the integrated prediction models. One of the core innovations of this platform is the newly introduced Variant Hotspot Explorer. This module provides a quantitative, temporally separated validation framework that natively computes Precision, Recall, and F1-scores, allowing researchers to benchmark model-derived signals directly against documented lineage-defining mutations. Although the platform offers several pre-trained models to illustrate what it can do, its modular open-source design lets researchers add their own architectures or integrate third-party models reported in prior studies[19]. Integrating external models requires careful attention to data preprocessing and differences in feature representation, but we view this flexibility as the main strength of CovMutEx and an important route for future collaborative development. To demonstrate this aspect, we integrated a state-of-the-art model [16], and benchmark it against other integrated models [15]. The remainder of the manuscript is structured to present the background and technical details of our study. We begin with a detailed review of prior studies relevant to this field. The following section examines the CovMutEx architecture, covering both the frontend and backend implementations. Next, we will present empirical results to assess the platform's performance and then discuss the system's functional and non-functional requirements. We close by outlining possible directions for future research.

Literature Review

This section reviews current tools for genomic visualization and mutation analysis and explains the technical differences that distinguish CovMutEx from prior approaches. In this review, we group prior work into two broad areas: widely used genome browsers and tools that focus on mutation analysis.

Genome Browsers

Genome browsers are the main reference maps that researchers use to examine genome regions and interpret genomic data. For SARS-CoV-2, a range of platforms now serve as key resources, and each offers different functions with clear trade-offs. For instance, the UCSC SARS-CoV-2 Genome Browser [20, 21][22] offers an integrated interface for examining reference sequences and gene interaction graphs. That level of detail can also create problems, because for many researchers the interface is complex enough to slow down quick, focused exploration. Other browsers are designed to serve narrower purposes. JBrowse [23, 24] is often used because it can display many types of genomic data and supports several visualization modes. In contrast, the CoV Genome Tracker [25] emphasizes mutations at the protein domain level and presents them through a phylogenetic tree interface. Platforms such as the WashU Virus Genome Browser [26] support comparative genomic analysis across multiple viruses, including Ebola, SARS, MERS, and SARS-CoV-2. Alongside these virus-specific tools, general-purpose engines still play an important role. Ensembl [27] supports many species and provides an API that is well suited to extension. Nextstrain [28] is widely used for ongoing, near real-time monitoring of pathogen evolution. IGV [29] is a widely used tool in research, often chosen to inspect SNP predictions and to load custom data tracks. GBrowse [30] still provides strong annotation functions, but its software and system requirements can at times restrict access on current computing platforms. 3D Genome Browser 2.0 [31] is an online platform for visualizing and analyzing 3D genome architecture. Despite their strong visual appeal, these browsers mainly function as static records of existing knowledge, not as tools that visualize and browse the what may come next as future mutations.

Tools for Mutation Analysis

Genomic analysis includes many different tools, and each one is designed to serve a particular purpose. This discussion examines several widely used platforms and compares their specialized architectures with the visual and interactive features of CovMutEx. Some of these tools focus on specific aspects of viral behavior. EVESCAPE [32], for example, uses biophysical and structural data to estimate the likelihood of immune-escape mutations. Although it is essential for antibody research, it mainly works at the level of protein regions and does not provide the nucleotide-level possible future mutation detail. In the same line of inquiry, Schwerdt et al. report related findings.The method described in [33] supports phylogenetic tracking of SARS-CoV-2 mutations, but it functions mainly as a retrospective monitoring tool rather than a system for forecasting future changes. We developed CovMutEx to address this limitation by automating the full data-processing pipeline, allowing users to focus on the analysis rather than manual preparation. For visual inspection of sequence variants, platforms such as VIMVer [34] provide high-resolution views at both the nucleotide and amino acid levels. In practice, VIMVer

is limited to mutations that have already been observed, and it relies on very strict input formatting, which can slow down rapid exploratory work. CovMutEx removes these formatting constraints and facilitates the exploration of inferred future mutational states. Related comparative studies, including Cedeño-Pérez and Gómez-Romero [35], identify protected genomic regions, but they depend on third-party tools for visualization and do not integrate predictive models. Tools such as ViralVar [36] and CovSeq [37] focus on lineage dynamics and on reporting specific amino-acid substitutions. Although the technology is sound, these systems often do not provide support for integrating predictive tools or interactive features to support what-if hypothesis generation. Machine learning is increasingly used in monitoring tools such as Covidex [38], which applies random forest classifiers to k-mer databases, and Coronapp [39], which generates mutational maps from GISAID data [40, 41]. Still, neither platform is mainly designed to visualize and explore future mutations through integrated models.

Viral Mutation Prediction Models

Recent work has introduced several deep learning architectures [42] for mutation forecasting MLAEP [43], PRIEST [16], DNMS [44]. Ma et al. [45][42] presents a lightweight language model that combines the regularity and randomness of viral mutations to predict future SARS-CoV-2 variants. It successfully identified strains like XBB.1.16, EG.5, JN.1, and BA.2.86 before their emergence. The PRIEST [16] introduces an interpretable deep-learning framework that uses temporal evolutionary information to predict SARS-CoV-2 mutations with immune escape potential. MLAEP [43] is a machine learning-guided framework that integrates structural modeling, multi-task learning, and genetic algorithms to predict SARS-CoV-2 antigenic evolution. It successfully forecasts high-risk mutations and synthetic variants with enhanced immune evasion. These models study key evolutionary drivers, including antigenic drift and mutation grammar, by using structural measures and fitness-based parameters. Although these approaches are technically strong, they are often designed for specific biological questions and do not offer a public, interactive interface that supports broad exploration. CovMutEx is not designed to outperform these high-performance models on standard predictive accuracy measures. Rather, it provides a complementary web-based platform that supports intuitive exploration of mutation probabilities across the entire genome. CovMutEx offers an open-source framework that allows researchers to add their own models, making it a flexible tool for genomic research. To demonstrate this capability, we integrated the pre-trained PRIEST model [16] alongside several sequence-conditioned models from Ayaz et al.'s study [15] into CovMutEx. PRIEST was selected as our state-of-the-art benchmark due to its temporal focus and open availability, which made it highly compatible with our web-based inference engine. While other notable high-performance models like MLAEP [43] and DNMS [44] offer robust methodological approaches, such as multi-task learning and structural genetic algorithms, they were not natively integrated into the platform in this iteration. Instead, we discuss them as alternative approaches to highlight the diversity of predictive architectures that future researchers could plug into CovMutEx's modular framework.

Comparison of features across existing tools

Although the tools described above offer useful ways to study viral diversity, most do not bring together the kind of forward-looking analysis needed to anticipate future SARS-CoV-2 changes across the full genome. Most current resources focus on cataloging variants after they have been observed, instead of supporting the prospective inference model integration that CovMutEx provides. Our comparison suggests that EVEscape [32] best matches the a candidate prediction model that could be integrated into our platform. EVEscape estimates the likelihood of immune escape in the Spike protein by combining evidence from structural exposure, evolutionary constraints, and biochemical differences. While it is well suited to identifying pathways of antibody escape, it remains a specialized tool and should not be treated as a general future mutation explorer. By comparison, CovMutEx proposes a genome-wide framework that visualizes estimated probabilities at the level of individual positions. Our system is open source and transparent, allowing researchers to deploy the provided pre-trained models or integrate their own architectures for analysis. To make these technical differences clear, we compare CovMutEx with several widely used genomic analysis platforms in Table 1. This overview reports which core functions are present and which are missing in the tools we reviewed. Unlike ViralVar [36], VimVer [34], CovSeq [37], and CoronApp [39], which mainly report observed data, CovMutEx allows users to interactively explore modeled future mutation paths by setting time points and variant-related parameters. EVEscape is mainly aimed at forecasting immune escape, whereas CovMutEx covers a broader range of genomic data. Interactivity is another point of divergence. Our tool supports dynamic data loading and smooth panning and zooming, which makes it easier to examine large datasets. Although CoronApp includes some of these interface elements, they are not consistently present in the other tools that were evaluated. These interactive features are necessary for researchers to examine high-dimensional viral evolution data in a precise and controlled way. By integrating future mutation inference models with advanced interactive visualization, CovMutEx offers a clear addition to current bioinformatics tools. By going beyond static observation, the platform supports proactive development of biological hypotheses and helps build a more detailed understanding of SARS-CoV-2 evolution.

Materials and Methods

CovMutEx was developed using a method that focused on fast data rendering and on integrating multiple prediction models within one workflow. We found early in the project that showing close to 30,000 genomic positions in real time required more than standard web components; it depended on careful coordination between the front-end display logic and the back-end inference pipeline. This section describes the system design, explains how the frontend and backend

Features	ViralVar	VimVer	CovSeq	CoronApp	Evescape	CovMutEx
Visualization of Predicted Mutations	✗	✗	✗	✗	Partial	✓
WebLogo view	✗	✗	✗	✗	✗	✓
Tree representation	✗	✗	✗	✓	✓	✗
Input data	User-Provided	Selection	User-Provided	User-Provided	Selection	Selection
Region-level analysis	✓	✗	✓	✓	✓	✓
Position-level analysis	✗	✗	✗	✗	✓	✓
View type	Chart	Table	Chart	Chart	Chart	Chart
Dynamic data loading	✗	✗	✗	✓	✗	✓
Pan	✓	✗	✗	✓	✓	✓
Variant Hotspot Profiling	✗	✗	✗	✗	✗	✓
Zoom	✓	✗	✗	✓	✓	✓

TABLE 1
Comparison of CovMutEx with existing tools

were implemented, and outlines the main technical details of the mutation prediction model workflow.

System Design

CovMutEx was built as a modular three-tier system, with a React frontend, a Django backend, and a set of deep learning models, where each layer is responsible for a distinct part of the workflow. This separation of concerns helps keep the interface smooth. The server handles the compute-heavy genomic feature extraction, while the browser renders the visuals using well-tested client-side libraries, so the system stays responsive during demanding analyses.

- **Frontend (React):** The frontend, built with React, provides the project's main interface for user interaction. It controls the user interface and the application state, so researchers can set lineage-specific parameters and then examine the resulting mutation probabilities using responsive charts drawn on a canvas.
- **Backend (Python/Django):** The backend serves as the system's traffic controller by routing requests, enforcing rules, and coordinating data flow between the database and the application logic[46][47]. It routes incoming requests, reads data derived from GISAID datasets, and performs the compute-intensive step of converting raw sequences into feature vectors used by the models.
- **Integrated Prediction Models:** These models form the main analytical component integrated into the platform. The models take preprocessed genomic vectors as input and output probability distributions across genomic positions; these results are then sent back to the client interface for visualization.

Figure 1 presents the system architecture and shows how the components relate to one another, as well as the order in which they exchange data.

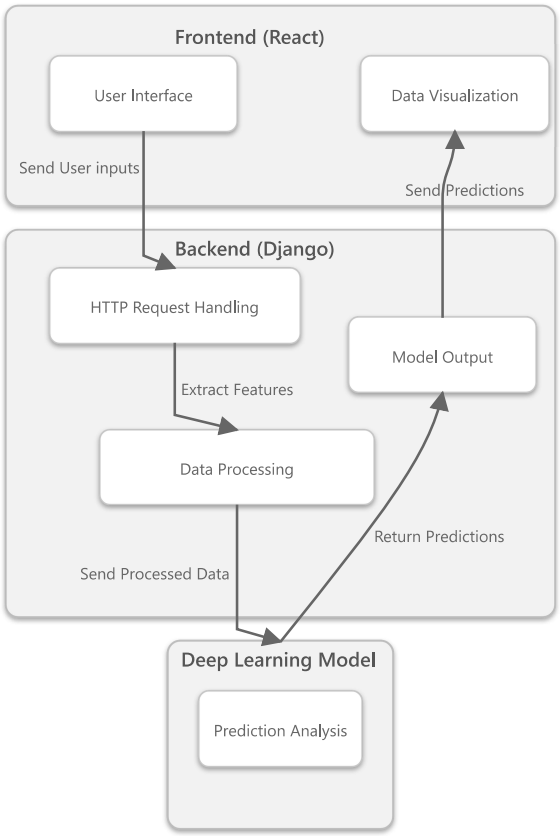

Fig. 1. CovMutEx System Architecture

We next describe the roles involved and the technology stacks used for the frontend and backend implementations.

Frontend Modules

This section outlines the main interface components and how they are built and integrated to support the required user tasks. To address the big data constraints in genomic visualization, we chose a current JavaScript stack that prioritizes fast rendering and reliable state management. We employed React because its component-based design supports a stateful interface in which different widgets, such as

dropdown menus and charts, stay synchronized [48]. Redux Toolkit was used to manage the data flow among these components by maintaining a central store that serves as a single source of truth. This approach ensures that when a researcher zooms in on a specific genomic region, the update is reflected immediately across the application state [49]. Visualizing 29,903 genomic positions at the same time is difficult and places heavy demands on the rendering pipeline. Chart.js draws to an HTML5 canvas instead of relying on SVG elements in the DOM[50][51]. This way, we aim to avoid browser-side lag when users navigated at high zoom levels [50]. Tailwind CSS was used to apply utility-based styles that support a responsive and clean interface, and the layout remains consistent across different screen sizes [52, 53].

Scalable User Input Capturing

Users begin by completing a parameter specification form, shown in Figure 2. Using this interface, users set up the prediction by choosing a model, a specific SARS-CoV-2 lineage [40][54], and the time period to be predicted.

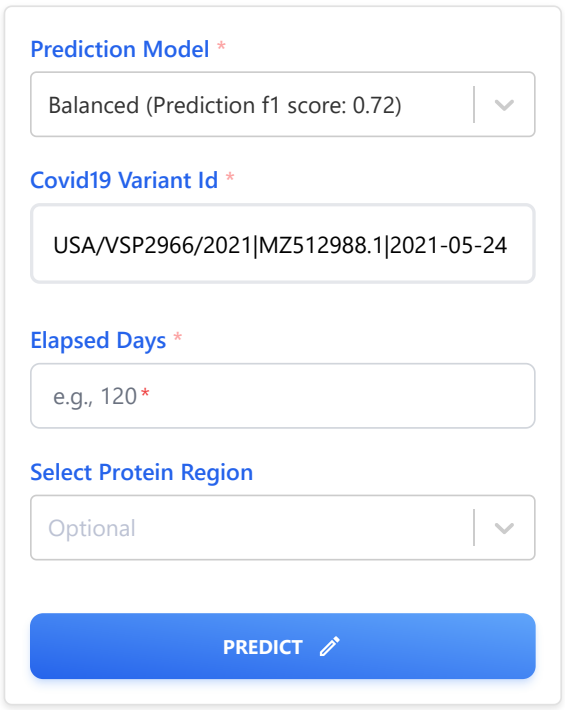

Fig. 2. Mutation prediction parameters form

A central engineering challenge in this work was handling the display of about 600,000 viral variants. In a typical dropdown menu, the browser tries to place every option into the Document Object Model (DOM). With a list of this size, that approach would quickly exhaust available memory and could cause the application to fail. We evaluated a server-side search approach, but the API response time was too slow to support a smooth user experience. We instead implemented list virtualization [55]. This method draws only the entries that are visible in the user's viewport, while keeping the remaining hundreds of thousands of entries stored virtually rather than rendered on the screen. This approach yields a fast list that allows real-time filtering, as shown in Figure 3.

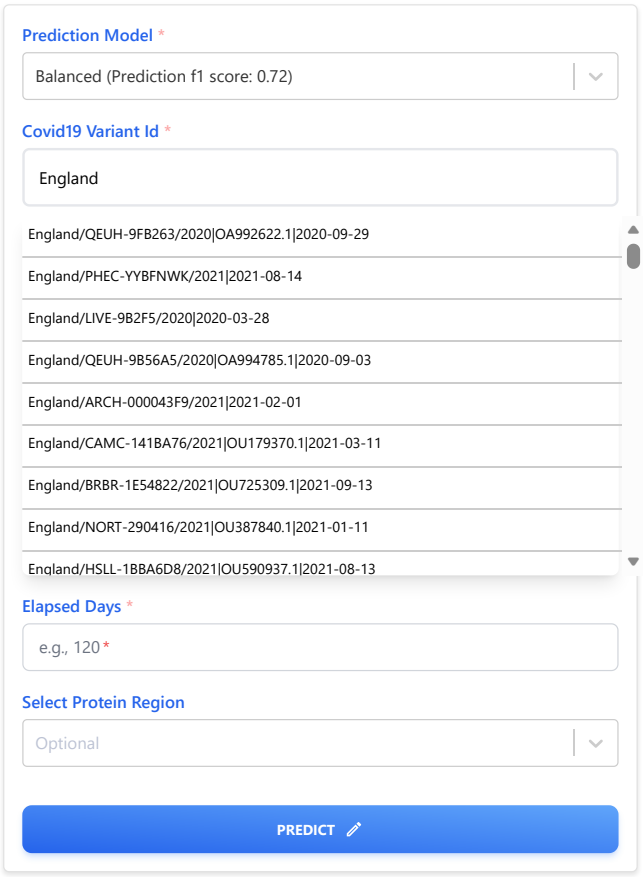

Fig. 3. Dynamic variant list virtualization

Interactive Charts and Data Reduction

After the parameters are submitted, the system processes the variant data and outputs a probability matrix with shape (4, 29903). To keep the visualization interactive, we used a data decimation approach. By default, the frontend groups dense probability regions into chunks using a factor of 25. This approach reduces rendering bottlenecks while still giving a clear overview of mutational trends across the genome (Figure 4). The rationale for this decimation is described in Algorithm 1. This approach lets the interface shift between a summary view and detailed bars at the position level as the researcher zooms in.

When users zoom in past a set threshold, the chart switches from the chunked view to a position-level view that shows finer detail. This approach provides clear, detailed results while reducing the risk of slowdowns when rendering large datasets.

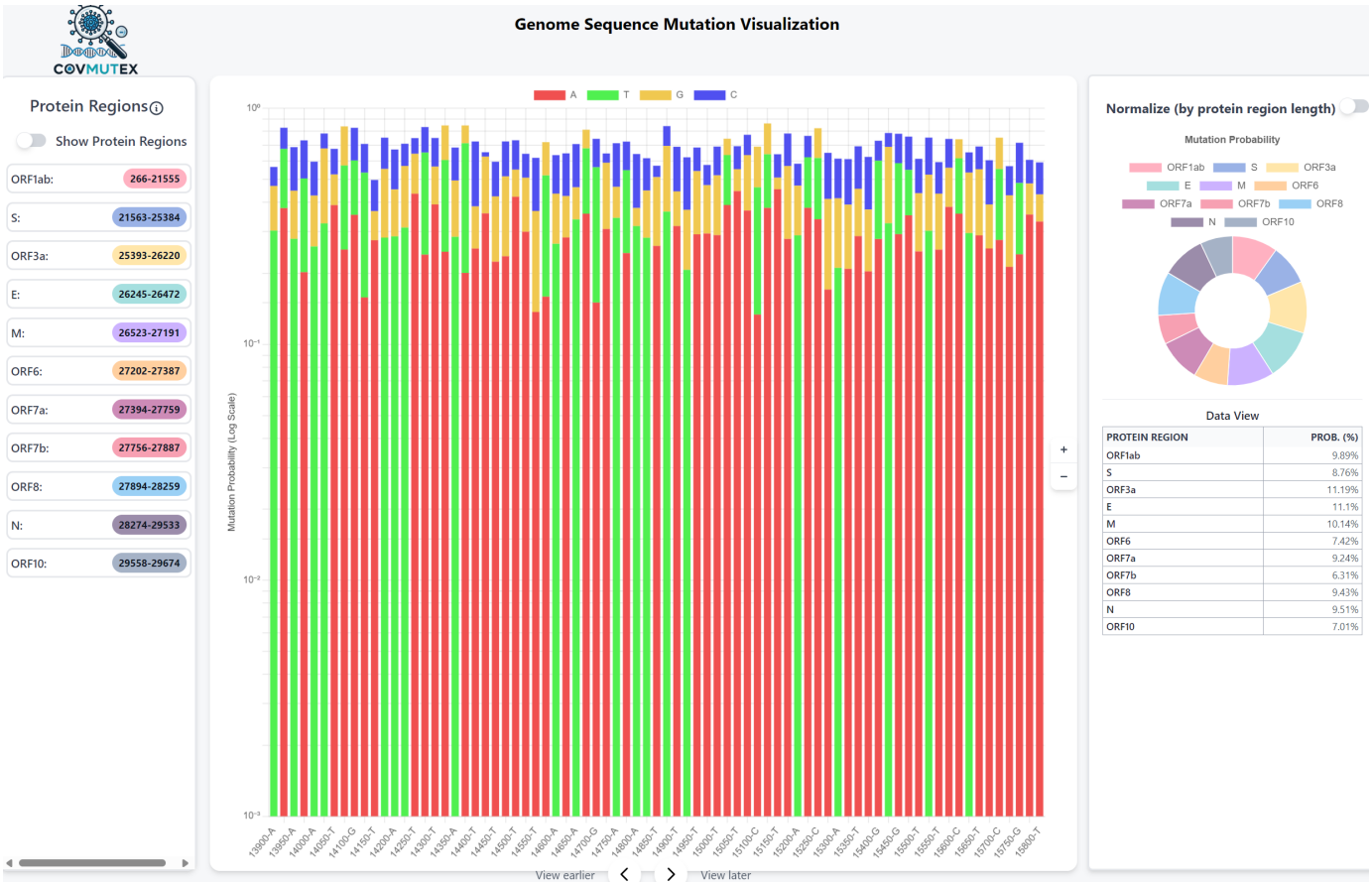

Fig. 4. Predicted mutation visualization

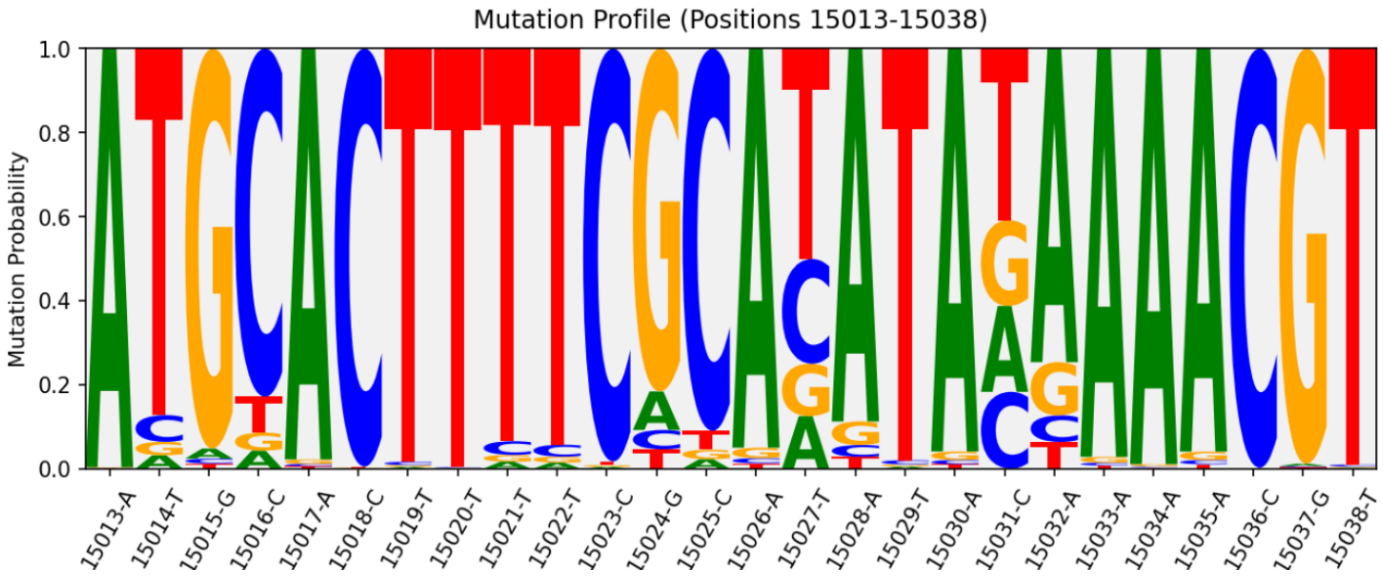

Fig. 5. WebLogo style position-level mutation visualization

To support smooth panning at the position level, where the full dataset must be loaded, we expanded the rendering widgets by adding padding on both sides and added event listeners that track mouse movement during interaction. This supports efficient data loading that follows the user's navigation within the chart. We also use a WebLogo-style view [56] at the highest zoom level to examine mutations at a single position, as shown in Figure 5. We use Logomaker [57] to generate sequence logo images during runtime. The stacked bar chart appears alongside a side panel and a

Algorithm 1: DecimateData

Input : D, a list of datasets,
$factor$, the decimation factor
Output: D', the decimated datasets

Initialize: $D' \leftarrow$ empty list;
foreach *dataset in* D **do**
  Initialize: decimatedDataset $\leftarrow$ empty list;
  foreach *index in dataset* **do**
    if *index* $\mod$ *factor* $= 0$ **then**
      slice $\leftarrow$ extract factor elements from dataset starting at index;
      avg $\leftarrow$ average of elements in slice;
      Append avg to decimatedDataset;
  Append decimatedDataset to D';
return D';

doughnut chart (Figure 4). The doughnut chart summarizes mutation probabilities across protein regions and lets users turn normalization on or off to support targeted analysis. Selecting a segment of the doughnut chart causes the stacked bar chart to mark the same region for comparison. The side panel shows clickable annotations that mark protein region boundaries and give context for the genomic data. Users can choose to show or hide these boundaries directly within the chart. Each protein region is shown in a distinct color, used both in the chart background and in the side panel annotations.

Backend Modules

CovMutEx's back end is implemented in Python, using machine learning libraries selected to support data processing and systematic model evaluation. Django is a Python web framework that supports building web applications in a structured way, with design choices that favor clear code and practical development [58]. Because it is built on Python, it is well suited for data preprocessing tasks on the back end. Django REST Framework extends Django to support the development of web APIs, enabling data exchange between the server and the front end [59]. The project uses PyTorch, NumPy, Pandas, and Scikit-learn to support large-scale data processing, numerical computation, data preparation, and the implementation of machine learning models. When the system receives input from the user, the back end starts a set of processing steps. If a specific protein region is given, the analysis is limited to that region only, which reduces computational cost and shortens response time. If not, the full genome sequence is processed.

Prediction Models Integrated From Literature

As a proof-of-concept, we integrated four pre-trained models from the literature, specifically from [16] and [15], as summarized below.

- PRIEST [16]: PRIEST (Prevalence-based Baseline): A state-of-the-art model that predicts viral mutations with immune escape capability by utilizing temporal evolutionary information. In CovMutEx, it is utilized to provide a prevalence-anchored ranking matched to specific lineage dates, serving as a biological reference for exact residue recovery.
- Balanced model [15]: The network has two fully connected layers. The first layer maps the 205-dimensional input feature vector to 128 dimensions. The second layer produces a binary output that indicates whether the sample is mutated or not.The model applies a ReLU activation function in the hidden layer and a sigmoid function at the output layer, and it is trained with the Adam algorithm using binary cross-entropy as the loss function.The model was trained on 100 balanced datasets, each formed by sampling the same number of mutated and non-mutated observations.
- Multi-input ensemble model [15]: Ten identical submodels are trained in parallel, each matching the base model's architecture. Their outputs are concatenated and then passed to the final prediction layer. Each submodel includes batch normalization, dense layers (205 - 256 - 128), and dropout layers (0.5 and 0.3 rates).The concatenated output, consisting of 640 features, is then fed into later layers to generate the final prediction.
- Single-input ensemble model [15]: This variant reduces the complexity of a multi-input ensemble by merging the inputs first and then passing the combined representation into one shared model.This model includes batch normalization and dense layers, but it applies L1 and L2 regularization in the early layers rather than dropout.A large dropout (0.8) is applied after concatenation, followed by a dense layer with 32 units and another dropout before the final output.

Probability Model: Let $\mathbf{x}_i \in R^{205}$ represent the input feature vector for a given candidate nucleotide at position i. The hidden layer representation $\mathbf{h}_i \in R^{128}$ in the neural network is computed using the Rectified Linear Unit (ReLU) activation:

$$\mathbf{h}_i = \max(0, \mathbf{W}_1 \mathbf{x}_i + \mathbf{b}_1)$$

where $\mathbf{W}_1 \in R^{128\times 205}$ is the learned weight matrix and $\mathbf{b}_1 \in R^{128}$ is the bias vector. The final position-specific mutation probability is generated by applying the sigmoid activation function to the output layer, constraining the estimation to a valid probability space $[0, 1]$:

$$P(m_{i,b}|S,t) = \frac{1}{1+\exp(-(\mathbf{W}_2 \mathbf{h}_i + b_2))}$$

where $\mathbf{W}_2 \in R^{1\times 128}$ and b_2 are the weights and bias of the output layer.

We directly employ pre-trained models from Saha et al. (2024) and Ayaz et al. (2025). Hence, we did not train models for this study. In the original studies, these models were trained on a large-scale curation of SARS-CoV-2 genomic and phylogenetic data, primarily sourced from the GISAID EpiCoV database [40]. PRIEST was trained on the spike protein sequence data from the final quarter of 2019 to the starting half of 2022. Ayaz et al.'s models were trained on all observed mutations between Dec. 30, 2019, and Oct. 01, 2021. The tested lineages do not overlap with these training periods, and they span from October 2022 to January 2025.

Integration and Optimization of External Models for Web Deployment

Deploying external deep learning models within an interactive web platform introduces challenges extending beyond model selection, encompassing framework compatibility, feature space standardization, and inference scalability. All adaptations implemented in CovMutEx were exclusively deployment-oriented; no model weights were retrained, pruned, quantized, or structurally modified after their original development.

Architectural Compatibility. Legacy Keras model artifacts (`.keras` or `.h5`) [60] were made compatible with the platform's current TensorFlow runtime [61, 62] through a custom loading and reconstruction procedure that intercepts model configurations at initialization, resolves tensor routing for Functional API architectures, and maps layer weights through a fallback hierarchy of signature matching, configuration-order loading, and layer-name resolution. This preserves the full structural and parametric integrity of each integrated model, preventing silent failures or tensor dimension mismatches during inference.

Feature Space Standardization. Raw biological sequences are dynamically transformed into a standardized 205-dimensional feature matrix to ensure live genomic inputs precisely mirror each model's original training distribution [15]. A 30-nucleotide sliding window is applied across the genome [63, 64], computing evolutionary and biochemical metrics on-the-fly, including PAM250 substitution scores [65] and 18 physicochemical amino-acid properties. All features are then bulk-preprocessed as matrix operations, where categorical variables are one-hot encoded [66, 67] and numerical features are Z-score standardized [68], ensuring strict conformity to each model's expected input distribution.

Inference Optimization. Three complementary strategies enable real-time inference across the full $\sim$30,000 base-pair genome. First, model objects are cached in memory after initial load, eliminating repeated deserialization overhead. Second, a differential feature extraction strategy restricts computation to only the nucleotide positions affected by a selected variant's mutations, merging the resulting partial feature vectors into a precomputed HDF5 reference matrix [69] rather than recomputing the full genome-wide matrix on every request. Third, candidate instances are evaluated in a single batched forward pass [61], and where applicable the prediction search space is narrowed to user-selected protein regions, further reducing latency without altering any learned model parameters.

Overall, the optimization strategy in CovMutEx prioritizes efficient serving of external models through compatibility adaptation, in-memory caching, differential feature computation, batched inference, and region-restricted evaluation, enabling real-time genomic inference within the browser without compromising the statistical integrity of any integrated model.

Quantitative Hotspot Analysis Framework

To address the need for empirical validation, we developed a framework that evaluates whether model signals concentrate on residues genuinely involved in viral evolution. Using Algorithm 2, we reconstruct lineage consensus genomes for seven notable variants (e.g., XBB.1.5, BA.2.86, and XFG) from GISAID frequency data. The framework benchmarks model performance using different metrics. Rather than predicting a single future mutation, this framework tests whether the strongest model-derived signals concentrate on Spike residues genuinely involved in the mutational architecture of a selected SARS-CoV-2 lineage, establishing a biologically grounded retrieval benchmark.

Algorithm 2: Lineage Consensus Reconstruction

Input: Lineage-specific sequence alignments
Output: M_ℓ, the lineage-specific reference mutation set

Initialize: Consensus sequence $\leftarrow$ empty list;
foreach *genomic position* g **do**
  Compute allele frequencies from lineage-specific sequence alignment;
  Identify alternate allele $a^* = \arg\max_a \text{freq}(a)$;
  if *freq*$(a^*) \geq 0.50$ **then**
    Incorporate a^* into consensus at position g;
  else
    Retain reference allele (suppress low-frequency variants);
Project nucleotide-level mutations to 1-based Spike amino-acid coordinates;
return M_ℓ *with* $|M_\ell| \in [43, 75]$;

Lineage Consensus Construction

For each of seven post-2022 lineages spanning October 1, 2022 (XBB.1.5) through January 1, 2025 (NB.1.8.1, XFG), we reconstruct a representative consensus genome from lineage-specific nucleotide mutation frequency tables that are derived from [70] based on sequences observed between Jan. 1, 2023 and April 8, 2026. The numbers of sequences used for mutation observation were 2124 for NB.1.8.1, 91453 for XBB.1.5, 130 for BA.2.86, 773 for XFG, 3150 for KP.2, 1553 for KP.3, and 9179 for XBB.1.16. The consensus construction process operates as follows:

The majority-frequency threshold of $\geq 50\%$ suppresses low-frequency background noise and yields a compact representation of the dominant mutational state. The filtered mutation catalog is then collapsed onto 1-based Spike amino-acid coordinates to define the comparison set of known mutation sites M_ℓ, with reference sets ranging from 43 to 75 Spike sites across the seven lineages 2.

Hotspot Scoring and Protein-Level Aggregation

Model inference within the CovMutEx pipeline is performed over the Spike coding region (nucleotides 21,563–25,384). Each nucleotide position is evaluated against four candidate nucleotides using feature representations including 30-nt sequence context, nucleotide and amino-acid substitution scores, elapsed time, phylogenetic depth, synonymous/non-synonymous status, and amino-acid biochemical descriptors. The model produces four mutation scores per nucleotide position.

To obtain protein-level hotspot scores, nucleotide-level outputs are aggregated at the codon level by summing non-reference score mass across the three codon nucleotides and averaging:

$$H(a) = \frac{1}{3}\sum_{i=1}^{3} \sum_{b\in\{A,T,G,C\}\setminus\{r_i\}} s_{i,b} \tag{1}$$

where $H(a)$ is the hotspot score for Spike amino-acid site a, r_i is the reference nucleotide at codon position i, and $s_{i,b}$ is the integrated sequence-conditioned model score for candidate base b. Sites are ranked by this raw codon-level score; min–max normalization to $[0, 1]$ is applied only for visualization.

Baseline and Evaluation Metrics

The Hotspot Case Study page supports both mutation probability-based scoring and a PRIEST-based (prevalence) baseline. Under the prevalence mode, each Spike position is assigned a quarter-specific site prevalence score matched to lineage date, with fallback to global prevalence when unavailable, enabling direct comparison under an identical evaluation framework.

Biological agreement is assessed using two complementary metrics. **Exact overlap** counts a hotspot as recovered only if its amino-acid coordinate exactly matches a known lineage mutation site. **Proximity overlap** accommodates clustered adaptive change by counting a hotspot as supported if it lies within ± 3 amino acids of a known site. For each threshold K, we compute:

$$\text{Precision@}K = \frac{\text{Top-}K \text{ positions with overlap}}{K} \tag{2}$$

$$\text{Recall@}K = \frac{\text{Top-}K \text{ positions with overlap}}{|M_\ell|} \tag{3}$$

$$\text{F1@}K = 2 \cdot \frac{\text{Precision@}K \cdot \text{Recall@}K}{\text{Precision@}K + \text{Recall@}K} \tag{4}$$

We report both exact-match and proximity metrics for all lineages and models, with performance highlighted at the F1-optimal operating point.

Interactive Analysis Interface and Backend Architecture

As illustrated in Figure 6, the frontend provides a detailed visual breakdown of the mutational landscape. The base plot aligns the full length of the Spike protein along the X-axis (amino-acid coordinates 1–1273) against a display-normalized version of the codon-aggregated prediction score on the Y-axis (ranging from 0 to 1). A continuous black trace line visualizes this computed hotspot score across every position, illustrating the overall topography of predicted mutational pressure.

To visually evaluate the model's performance against the established metrics, the interface overlays a color-coded, interactive marker system:

- **Explorer hotspots (Blue circles):** Denote the Top-K ranked positions, representing distinct peaks where the model predicts the highest localized mutational pressure.
- **Known variant sites (Yellow diamonds):** Mark the lineage-specific reference mutation set (M_ℓ), derived from the majority-frequency consensus algorithm.
- **Overlap hits (Red stars):** Visually represent the *Exact overlap* metric, appearing with a red dashed drop-line when an Explorer hotspot precisely matches the coordinate of a Known variant site.
- **Near-hits $\pm$3aa (Green circles):** Visually represent the *Proximity overlap* metric, appearing with a green dashed drop-line when a predicted hotspot falls within a 3-amino-acid sliding window of a known lineage mutation.

The Case Study analysis page is backed by a dedicated REST API endpoint that serves as the computational backbone for both this interactive exploration and offline analysis. The API returns:

- **Per-site scores**: Complete hotspot score vector $\mathbf{H} = [H(1), \ldots, H(1273)]^\top$
- **Ranked results**: Hotspot table sorted by score with lineage membership annotations
- **Metadata**: Consensus date, sequence counts, allele frequency distributions
- **Reference definitions**: Explicit lineage-specific mutation site enumeration M_ℓ
- **Overlap statistics**: Exact and proximity F1 scores at all Top-K thresholds
- **Complete Top-K sweep**: Full precision@K, recall@K, F1@K trajectories

The same backend payload powers both the interactive React frontend and offline analysis pipelines, ensuring deterministic consistency between the visual overlays shown in the web tool and the quantitative metric figures. This unified architecture eliminates discrepancies between exploratory and publication-grade results.

EXPERIMENTAL RESULTS

This section evaluates CovMutEx in terms of scalability, capacity under load, usability, and performance relative to existing tools. We first present an empirical evaluation of performance, then discuss a qualitative assessment, and end with a use case that shows how CovMutEx is typically applied.

Performance evaluation

To evaluate CovMutEx performance, we recorded preprocessing time, prediction time, rendering time, and total response time while varying genome length and the number of concurrent users. All evaluations were run on a PC with the following specifications. The system is equipped with 2 x i7-7700HQ 2.8GHz CPU, 16 GB of RAM, and an NVIDIA GeForce GTX 1050 Ti graphics card with 4 GB of dedicated video memory.

Genome length analysis

We first examine how the combined time for preprocessing, prediction, and rendering varies as genome length, measured in nucleotides, increases. Figure 7 shows that rendering time stays roughly constant as genome length varies, which supports quick visualization of mutation

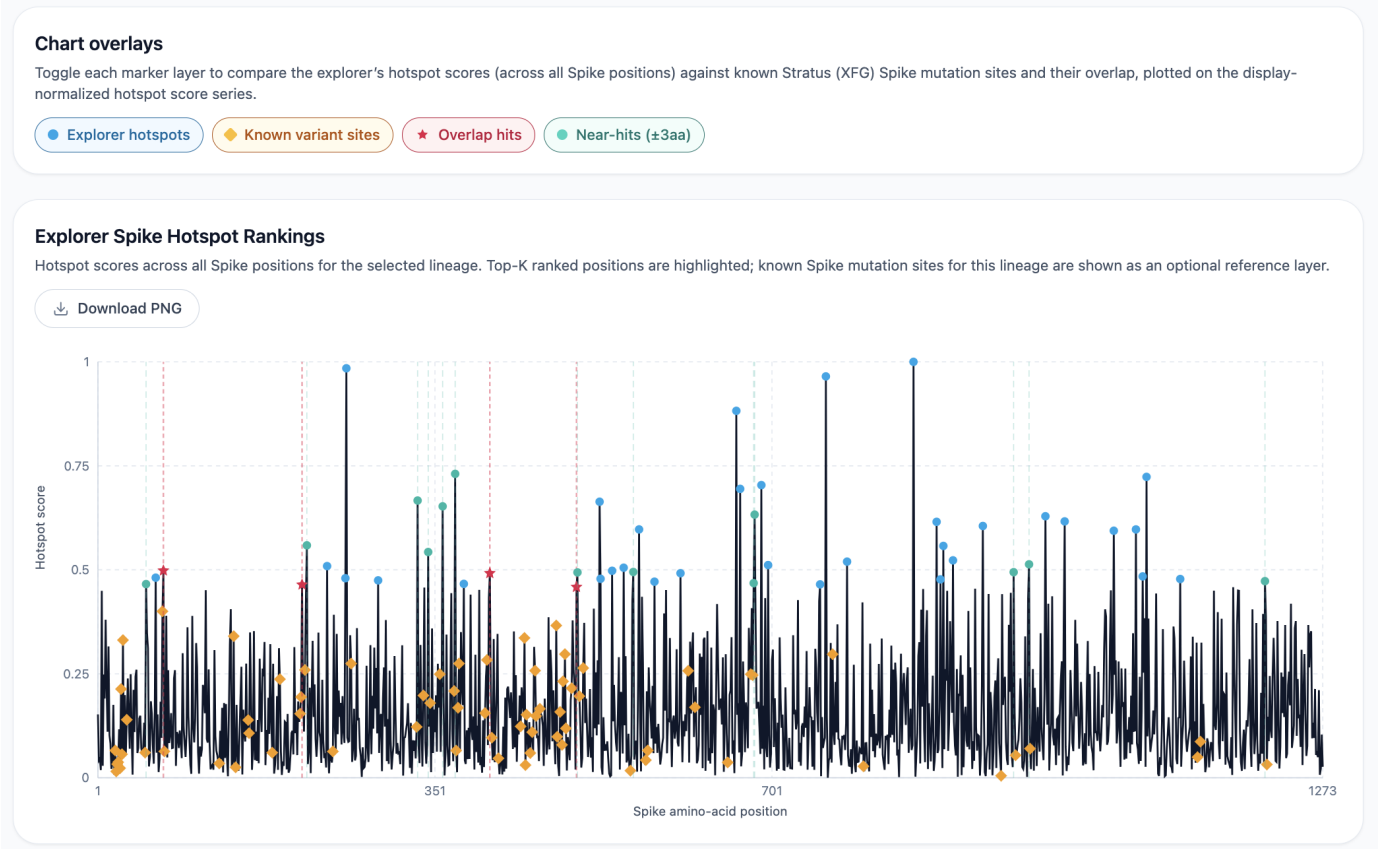

Fig. 6. Screenshot of the Hotspot Analysis page. The interface provides a high-resolution, full-length graphical representation of how the integrated sequence-conditioned model's mutational pressure predictions align with established biological ground truth, shown here for the Stratus (XFG) lineage.

probabilities. This consistency likely results from how the data were first divided into chunks for the visualization. CovMutEx splits the genome into fixed-size chunks, so the number of chunks stays the same in the visualization even when the total genome length differs. For instance, if a genome has 30,000 nucleotides and is split into blocks of 30, it produces 1,000 chunks. If a genome has 100,000 nucleotides and is split into blocks of 100, it also produces 1,000 chunks. Rendering time stays roughly constant, while prediction and preprocessing make up most of the total processing time, and prediction time rises more quickly as genome size increases. With the SARS-CoV-2 genome at about 30,000 nucleotides, CovMutEx performs at a level that is appropriate for its intended application.

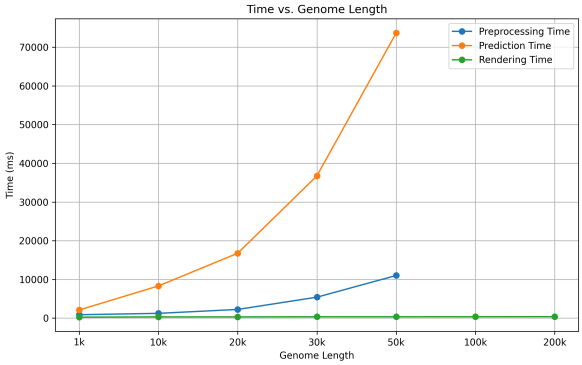

Fig. 7. Scalability of CovMutEx as genome length increases

Scalability with User Load

Figure 8 shows how CovMutEx response time changes as the number of concurrent users increases.As user load rises, the average, minimum, and maximum response times increase moderately. CovMutEx remains reasonably responsive even when many users access it at the same time, which suggests it can support varied user needs with only minor declines in performance. CovMutEx shows good performance, with low time costs for preprocessing, prediction,

and rendering. Because it can process genomes of different lengths and scale to different sample sizes and levels of user demand, the tool can support research aimed at understanding SARS-CoV-2 and how it changes over time.

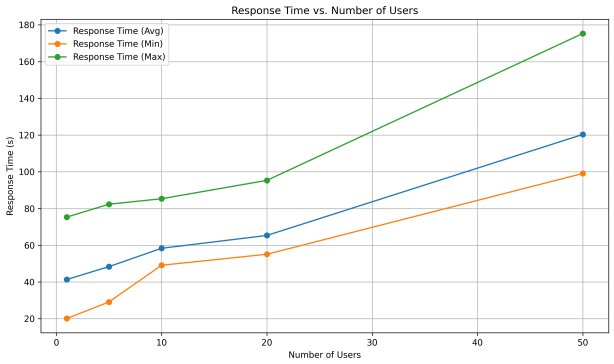

Fig. 8. Scalability of CovMutEx as the number of concurrent users increases

Case Study on Notable Variants of Interest

Our analysis across seven lineages indicates distinct utility profiles for the integrated models. To specifically address predictive performance against known major variants, all seven lineages evaluated in this study (e.g., Kraken/XBB.1.5, Pirola/BA.2.86, and the FLiRT variants) are progressive descendants of the Omicron variant. This allows us to rigorously assess how well the models track the continuous antigenic drift of the most dominant global variant family. While the PRIEST [16] baseline excels at exact residue recovery due to its prevalence-anchored ranking (Mean Exact $F1 = 0.264$), Ayaz et al.'s models [15] demonstrate superior performance in identifying biologically coherent neighborhoods. For instance, the integrated multi-input model surpasses PRIEST in proximity recall for the Kraken (XBB.1.5) variant. The balanced model proved most robust for recently emerged variants like KP.3 and XFG, reflecting its ability to map evolutionary plasticity through sequence-conditioned insights rather than historical prevalence alone. All comparative analyses were conducted at a Top-$K = 50$ threshold, operationalizing six complementary metrics per model-lineage pair (Figures 9, 11, 12, 13, 14, and 15): exact overlap (EO), exact precision@K (EP), exact recall@K (ER), proximity overlap within ± 3 amino acids (PO), proximity precision@K (PP), and proximity recall@K (PR). Exact metrics quantify strict residue-level recovery of lineage-defining mutations, while proximity metrics assess whether a model correctly delineates functionally permissive mutational neighborhoods, including flexible surface loops and antigenic epitopes. Together they provide a two-dimensional view of model utility, one axis measuring positional fidelity and the other measuring regional biological relevance.

Exact Residue Recovery. PRIEST dominates exact-site identification across all seven lineages, recovering a mean of 12.0 exact residues within its Top-50 list, more than three times the rate of the strongest integrated sequence-conditioned model (multi-input, mean EO = 3.71). The contrast is sharpest for Pirola (BA.2.86), where PRIEST captures

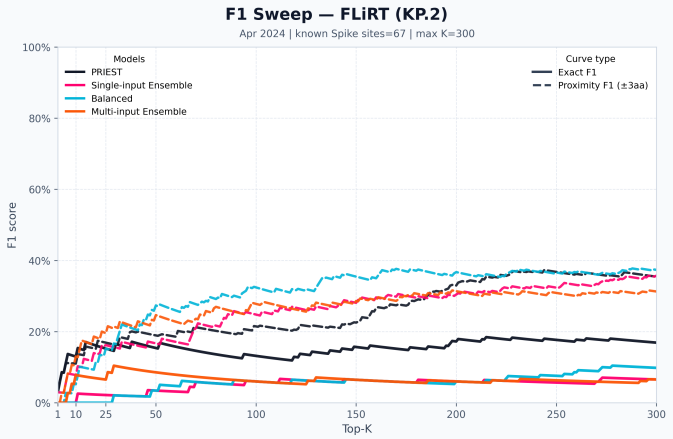

Fig. 9. Comparative Top-K F1 Profiling of Mutation Hotspot Prediction Models – FLiRT (KP.2).

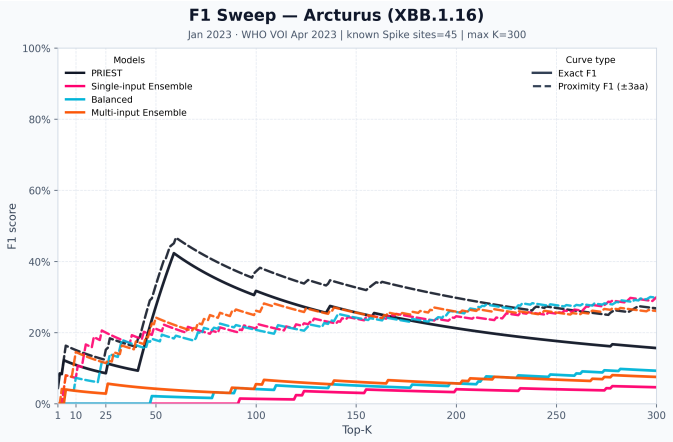

Fig. 10. Comparative Top-K F1 Profiling of Mutation Hotspot Prediction Models – Arcturus (XBB.1.16).

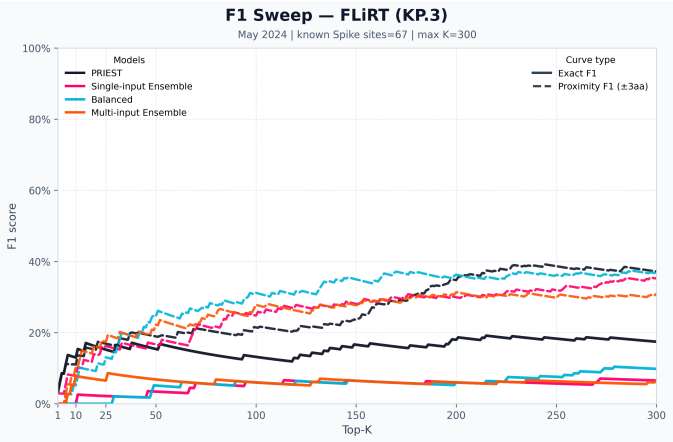

Fig. 11. Comparative Top-K F1 Profiling of Mutation Hotspot Prediction Models – FLiRT (KP.3).

27 exact overlaps (EP = 54.0%, ER = 42.9%) versus only 4 for integrated multi-input model. This reflects PRIEST's

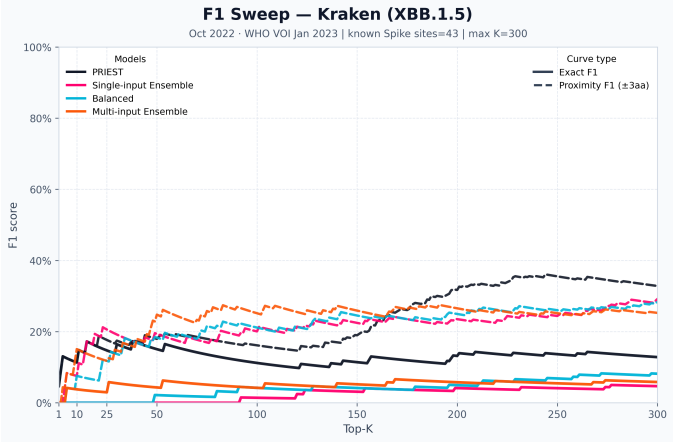

Fig. 12. Comparative Top-K F1 Profiling of Mutation Hotspot Prediction Models – Kraken (XBB.1.5).

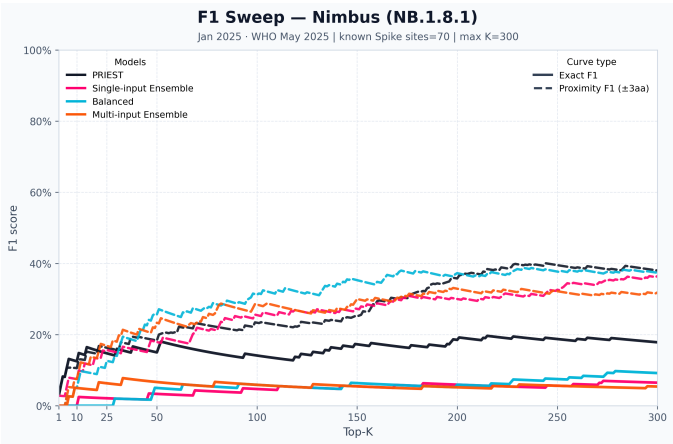

Fig. 13. Comparative Top-K F1 Profiling of Mutation Hotspot Prediction Models – Nimbus (NB.1.8.1).

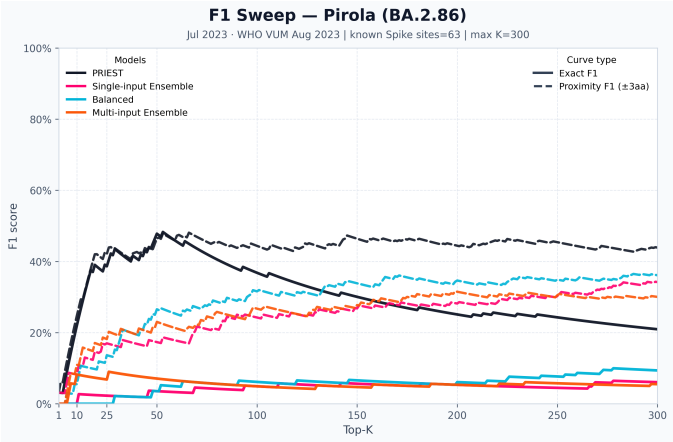

Fig. 14. Comparative Top-K F1 Profiling of Mutation Hotspot Prediction Models – Pirola (BA.2.86).

quarter-specific prevalence ranking, which is inherently optimized to recover lineage-defining sites that co-localize with recurrently selected population-level positions.

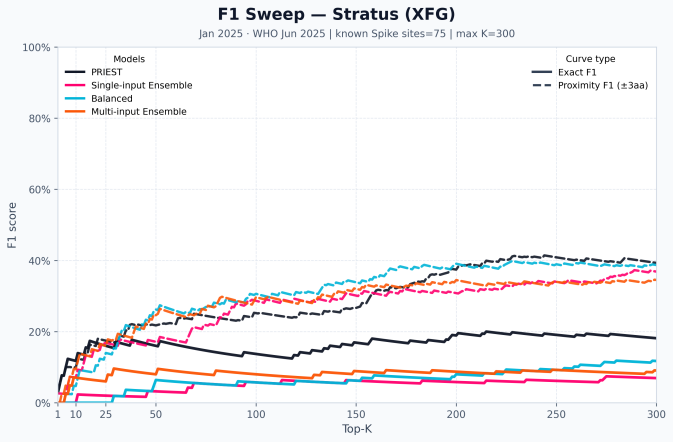

Fig. 15. Comparative Top-K F1 Profiling of Mutation Hotspot Prediction Models – Stratus (XFG).

Biologically Coherent Neighborhoods. Under proximity metrics, the performance gap narrows substantially and in several contexts reverses. For Kraken (XBB.1.5), integrated multi-input model surpasses PRIEST across all three proximity dimensions (PO: 12 vs. 10; PP: 24.0% vs. 20.0%; PR: 25.6% vs. 16.3%). For later-emerging variants, KP.2, KP.3, XFG, and NB.1.8.1, integrated balanced model matches or exceeds PRIEST in proximity recall, reflecting robust identification of the broader mutational pressure zones characteristic of progressive antigenic drift. This advantage is rooted in architecture: integrated sequence-conditioned models infer site-specific pressure against an explicit lineage consensus by integrating local sequence context, biochemical properties, and temporal covariates, granting interpretive capacity unavailable to prevalence-driven approaches.

Intra-Family Differentiation. Within the integrated sequence-conditioned model family, the multi-input ensemble achieves the highest exact-site recovery, functioning as a high-specificity filter suited to targeted confirmatory analysis. The balanced model exhibits broader score distribution and superior proximity recall for recently emerged variants, which is advantageous when mapping spatially clustered mutational events. The single-input ensemble consistently underperforms both, confirming that multidimensional contextual integration is a necessary architectural requirement.

Calibration Under Variable Thresholds. At each model's F1-optimal threshold, PRIEST retains overall leadership (mean best exact F1 = 0.264, proximity F1 = 0.413). Within the integrated sequence-conditioned model cohort, the balanced model is the most robust variant (exact F1 = 0.101, proximity F1 = 0.355), surpassing multi-input (0.086 and 0.311, respectively), an inversion relative to fixed-threshold results that underscores the importance of aligning model selection with the analytical objective. PRIEST excels at recovering prevalence-anchored hotspots with high positional precision, while the others provide sequence-conditioned, neighborhood-aware insight into evolutionary plasticity. Their integration offers interpretive power that substantially exceeds either model class in isolation.

Usability Evaluation

This section describes how usability was assessed for the system. It outlines the evaluation goals, the participant group, the tasks used during testing, and the measures collected to judge ease of use. It also explains how the study was conducted and how the findings were analyzed to identify usability issues and guide revisions. To assess the usability of CovMutEx, we ran a structured survey with 25 participants. All participants watched a training video that explained the purpose of the study. All participants provided informed consent before taking part. In particular, they were informed that their responses would be used for system evaluation and that data would be collected anonymously. All procedures were conducted in line with the relevant guidelines and regulations. The survey consisted of ten Likert-scale items adapted from the System Usability Scale (SUS) [71]. These items assessed ease of use, how well the system worked with other tools, perceived complexity, and users' confidence while using it. The survey is available at https://forms.gle/pedTnt51HzAF9QSC8. Figure 16 presents visual summaries of the responses.

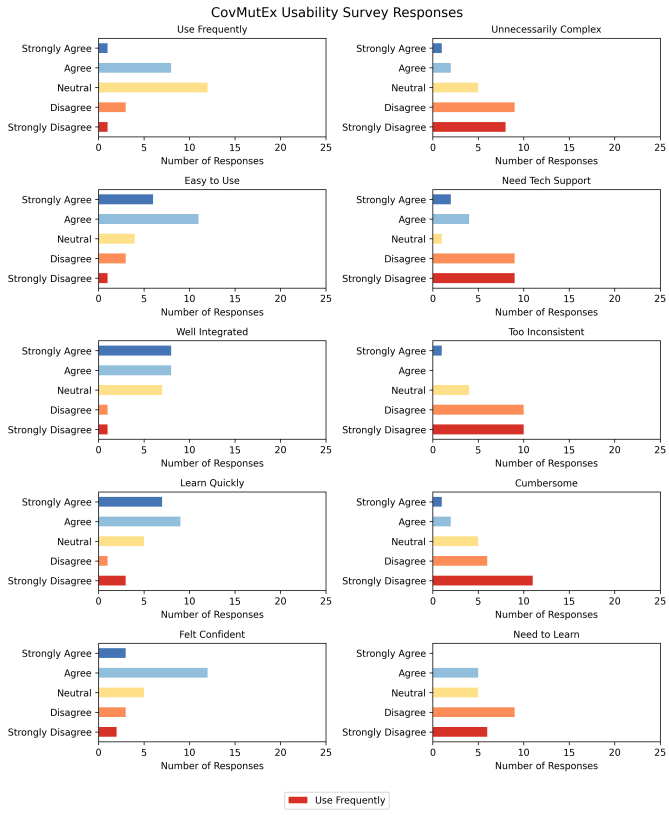

Fig. 16. Summary of usability responses from 25 participants evaluating CovMutEx. The chart visualizes agreement levels across ten usability statements adapted from the System Usability Scale (SUS). The charts on the left column belong to positive-toned questions (e.g., the tool is easy to use), and the ones on the right belong to negative-toned questions (e.g., the tool is cumbersome).

Study Design and Tasks

Before completing the survey, participants were asked to watch an approximately five-minute training video on the About page and then complete two typical tasks in CovMutEx.

- Task 1: Estimate mutation probabilities at a given genomic position.Participants applied CovMutEx to estimate mutation probabilities at a specified site in the SARS-CoV-2 genome.
- Task 2 compares normalized mutation probabilities for ORF8. Participants compared normalized mutation probabilities in the ORF8 protein region across several prediction models.

The tasks were set up to mirror realistic use cases and to ensure that participants worked with both the position-level and protein-region-level features of the tool.

Participant Demographics

The study included participants from a range of demographic backgrounds. In our sample, 52% of respondents reported backgrounds in the life sciences (for example, molecular biology, genetics, and medicine), while 48% reported backgrounds in the computational sciences including computer science and software engineering. In the sample, 64% of participants held a Master's degree, 12% held a PhD, and 24% held a Bachelor's degree. Participants reported varying levels of experience in bioinformatics and/or genomics: 72% had 1–5 years, 24% had 5–10 years, and 4% had more than 10 years.

Analysis of System Usability Scale (SUS) scores

We computed the System Usability Scale (SUS) score (0–100) from responses to 10 Likert-scale items rated from 1 to 5 [71]. In particular, we apply the following numerical mapping: "Strongly Disagree" was coded as 1. In the coding scheme, the response category "Disagree" was assigned the value 2. The category "Neutral" received a score of 3. The response category "Agree" was coded as 4 and "Strongly agree" as 5. For negatively worded items, the SUS scoring procedure inverts the item score before calculating the total. To compute the SUS score for each participant, for each of the 10 questions, we convert the response into a numerical value. For the positively keyed items (Q1, Q3, Q5, Q7, and Q9), we compute the adjusted score by subtracting 1 from the response value. For negatively worded items (Q2, Q4, Q6, Q8, and Q10), we reverse-score each response by subtracting it from 5 (e.g., a response of 5 is recoded as 0, and a response of 1 is recoded as 4). Finally, we sum the contributed scores, with each score ranging from 1 to 5, and then multiply the total by 2.5 to convert the result to a score from 0 to 100. The main findings are summarized as follows. The average System Usability Scale (SUS) score was 74.5, which indicates good usability and exceeds the industry average score of 68 [72][73][74][75]. Score distribution was as follows. A score of 80 or higher, classified as excellent, was achieved by 20% of participants. Scores between 70 and 79 (Good) account for 52% of the sample. Scores in the 60–69 range were classified as average and represented 16% of the sample. Scores below 60, classified as poor, accounted for 12% of the sample.

Quantitative results from the analysis

The usability responses pointed to a few clear patterns in the data. In the survey, 68% of respondents agreed or

strongly agreed that the system was easy to use. Regarding feature integration, 64% of respondents reported that the system's functions were well integrated. 64% of respondents reported that most people would be able to learn how to use the system quickly. 60% of participants reported that they felt confident using the system. In the survey, 20% of respondents described the system as unnecessarily complex, and 16% reported that it was cumbersome. Task 1 Accuracy was 84%, meaning that 84% of the responses were accurate. Task 2 accuracy: 68% of participants produced correct or partly correct solutions.

Interpretation of the results and their implications

The usability feedback was generally positive, suggesting that CovMutEx is accessible to users with backgrounds in both computational and life sciences. The high confidence and learnability scores indicate that many researchers are likely to adopt the tool with little formal training. A smaller group of users reported problems with complexity or inconsistency, which suggests areas that may need improvement in future work, such as, improve onboarding and tutorial features to support new users during initial use, revising the interface so users can reach core functions with fewer steps and less searching, improving the consistency in visual elements and the terms used across the document. In response to user feedback, we made the following changes to the tool:

- A table was added beneath the protein region mutation probability pie chart to report the probability for each region in explicit terms. Before this change, the probabilities for each region were available only in the tooltip that appeared when the cursor hovered over the relevant slice of the pie chart. Several users reported that the interface was not intuitive to use. The addition of the table improved the visibility of this information. This revision aims to improve the lower success rate observed in Task 2.
- When a protein is selected in the leftmost pane, the view now automatically zooms to the corresponding protein region. In earlier versions, this action only marked the selected area and did not change the zoom level or focus. Several participants noted that users should not need to manually zoom in to view the relevant protein region.
- In some participants' sessions, the coding region selection on the input page did not respond as expected.We revised the front-end implementation for this component to improve its responsiveness.

Use Case: Exploring SARS-CoV-2 Variant Evolution with CovMutEx

Suppose a researcher is examining how different SARS-CoV-2 variants might change over time and which genetic changes could arise under different selective pressures. The researcher use CovMutEx to examine possible genetic mutations and to estimate which variants may emerge in the future. When the researcher opens the CovMutEx interface, an input form appears (Figure 2). The user chooses a SARS-CoV-2 variant from the drop-down menu (Figure 3), enters the number of days since the variant emerged, and then selects a pre-trained machine learning model to generate a prediction. The prediction results are shown in a stacked bar chart, which first presents binned mutation probabilities across the full genome (Figure 4). In the bar plot, each color corresponds to a different nucleotide, and the height of each colored segment shows the mutation probability at that position. When reviewing the visualization, the researcher notes areas with high mutation probability, which may indicate changes in the protein sequence. When the "Show Protein Region" option is turned on, the relevant regions are marked with annotations, making their biological role clearer. The researcher notes that ORF1ab accounts for a large share of the genome, which suggests that it may influence viral evolution. To develop a more detailed understanding, the researcher uses CovMutEx's interactive features. By examining the data at a finer scale, she can assess mutation probabilities at each position and identify nucleotides that are more likely to change. This analysis helps the user identify potential genomic hotspots. This analysis also allows the researcher to identify plausible mutation pathways for the selected SARS-CoV-2 variant. The visualization shows areas with high mutation probabilities, which may help identify targets for vaccine development and therapeutic interventions.

DISCUSSION

The emergence of SARS-CoV-2 showed the need for computational tools that support genomic surveillance and guide public health planning. Many web-based tools support genomic analysis and visualization, but most do not allow users to visualize model-predicted mutation probabilities across a viral genome; instead, they mainly report changes that have already been observed in existing data [20, 24, 25, 26]. CovMutEx was designed to fill this need by combining predictive deep learning models with an interactive visualization platform that supports flexible exploration of results. While CovMutEx remains primarily a software-oriented explorer, the integration of SOTA models and the addition of the Hotspot Explorer provide a rigorous framework for tool evaluation. The interface enhances interpretability through a color-coded marker system: 'Red stars' signify exact overlap hits, while 'Green circles' denote near-hits within the functional neighborhood. This visualization bridges the gap between raw algorithmic output and actionable surveillance, allowing researchers to choose the most effective model for a specific lineage context.

Key contributions and how this work differs from earlier studies

This study does not aim to introduce new mutation prediction models or to claim state-of-the-art predictive accuracy. Its main contribution is a flexible, extensible, interactive platform for visualizing and exploring plausible future mutations in the SARS-CoV-2 genome.

- Bridging future mutations and visualization: Current genome browsers such as UCSC and Ensembl, and mutation trackers such as ViralVar and VimVer, mainly report mutations that have already been observed. CovMutEx instead visualizes model-predicted mutation probabilities across the full genome at single-position

resolution and presents these findings in a form suited for visual inspection. Users can set inputs such as Variant ID and Elapsed Days and then inspect inferred mutation probabilities using stacked bar charts and WebLogo-style plots.
- Extensibility and flexibility are supported through an open-source, modular design that allows researchers to add their own models for analysis (with some effort).
- CovMutEx supports interactive work with the SARS-CoV-2 genome by allowing dynamic data loading, along with panning and zooming, which helps users move through large genomic datasets more efficiently.The interface shows the genome in chunks for broad, genome-wide inspection and, at the highest zoom level, switches automatically to a position-level WebLogo view, so users can locate candidate hotspot regions.

Validation of system performance and usability

Empirical performance benchmarks showed that CovMutEx runs efficiently and scales well. The implementation kept the visualization rendering time fairly stable as genome length increased by splitting the input data into chunks at the outset. Improving server-side data preprocessing and the handling of batch prediction requests reduces end-to-end latency and keeps response times within an acceptable range, even when many users submit requests at the same time. A structured usability study with 25 participants from varied backgrounds found that the platform was accessible. The mean System Usability Scale (SUS) score was 74.5, which indicates good usability and is above the industry benchmark of 68, suggests that researchers from different backgrounds can adopt the tool with minimal training. The high confidence and learnability scores support the design choices intended to improve the user experience.

Implications for Viral Research

CovMutEx integrates predictive models with interactive features and may be a useful tool for genomic analysis, especially when users need both model-based predictions and direct exploration of results. It links integrated model provided future mutation probability estimations with an interface that allows interactive examination, helping researchers form hypotheses and support public health decisions about changes in viral lineages over time.

CONCLUSION

The SARS-CoV-2 pandemic has shown that public health needs computational systems that can translate genomic surveillance data into timely, preventive action. Although viral modeling research has advanced, there remains a practical lack of tools that support hands-on, interactive exploration of predicted mutation pathways. We developed CovMutEx to address this specific problem. It serves as a web-accessible hub that places priority on system performance, modular design, and the researcher's experience rather than on introducing a single fixed algorithm. Our evaluation, based on performance benchmarks and a usability study with domain specialists, indicates that the platform meets technical requirements and can be used by a wider group of researchers. A System Usability Scale (SUS) score of 74.5 suggests that users rate the system's usability as above average in typical benchmark terms. This high usability score validates the design choices we made so that researchers with different backgrounds can work through complex 30 kb genomic datasets with clarity.

LIMITATIONS OF THE STUDY AND DIRECTIONS FOR FUTURE RESEARCH

No analytical framework is free of limitations, and CovMutEx is no exception. At this stage, our feature extraction and preprocessing procedures are custom-tailored specifically for the SARS-CoV-2 genome. Adapting this platform to other RNA viruses, such as Influenza or MERS, would require specific architectural revisions to the data processing logic rather than simple configuration changes. Specifically, the pipeline must be expanded to accommodate different reference genome alignment protocols, modified codon mapping logic for segmented or alternative reading frames, and updated biochemical feature arrays tailored to the unique mutational grammar of the target pathogen. Expanding the framework to support cross-species adaptability is a primary objective for future iterations.

Besides, while the current platform enhances interpretability through 1D sequence annotations and quantitative hotspot overlays, it currently lacks a functional annotation module for mapping predictions onto 3D protein structures. Consequently, another target feature for the next major iteration is the integration of 3D structural biology viewers to contextualize mutations within known functional domains (e.g., receptor-binding interfaces and immune escape epitopes). Furthermore, we plan to incorporate real-time clinical and epidemiological metadata—such as geographic distribution, localized transmission rates, and patient symptom severity so that predicted mutational hotspots can be interpreted within a comprehensive public health context. Finally, to maximize rendering speed and minimize server load, CovMutEx is currently deployed as a stateless, privacy-preserving analytical explorer. Consequently, persistent user sessions and collaborative sharing functions were not included in the initial system architecture. Recognizing the critical role of team collaboration in modern bioinformatics research, the development of secure workspace-sharing features and exportable analytical sessions stands as a highly valuable direction for future platform enhancements.

An additional limitation concerns the platform's reliance on GISAID-derived sequence data for prevalence-anchored baseline evaluations. While GISAID is an indispensable global resource, it is subject to inherent spatial and temporal sampling biases. Disparities in genomic surveillance infrastructure across different countries frequently result in delayed sequencing uploads from underrepresented global regions. Consequently, these geographic and temporal data bottlenecks can skew the apparent frequency of emerging mutations, which may impact the real-time accuracy and generalizability of prevalence-anchored predictive models. Future iterations of the platform will explore integrating statistically weighted sampling methods to help mitigate these database-driven biases.

DECLARATIONS

Ethics approval and consent to participate

The usability study was conducted as an anonymous evaluation of software functionality. For non-invasive software usability testing, formal ethics committee approval was not required. Informed consent was obtained from all participants.

AI usage

Language-editing tools were used for grammar and formatting. All technical content was authored, checked, and approved by the authors, who take responsibility for the manuscript.

Consent for publication

Not applicable.

Availability of data and materials

Data are available at GISAID (https://gisaid.org/). Source code: https://github.com/itu-bioinformatics-database-lab/CovMutEx.

Availability and requirements

Project name: CovMutEx
Project home page: https://covidmutex.itu.edu.tr
Operating system(s): Platform independent
Programming language: Python
Other requirements: Python 3.8+
License: GNU GPL
Any restrictions to use by non-academics: licence needed.

Competing interests

None declared.

Funding

Supported by Istanbul Technical University (ITU BAP) grant FHD-2024-45740 and the National Center for High-Performance Computing (UHEM) grant 1009742021.

LIST OF ABBREVIATIONS

API: Application Programming Interface
CSS: Cascading Style Sheets
DOM: Document Object Model
GISAID: Global Initiative on Sharing All Influenza Data
ML: Machine Learning
ORF: Open Reading Frame
RNA: Ribonucleic Acid
SARS-CoV-2: Severe Acute Respiratory Syndrome Coronavirus 2
SUS: System Usability Scale
UML: Unified Modeling Language

Authors' contributions

AKY implemented the frontend, AYI implemented the backend, HY implemented the ML models, AC conceived the study and acquired the funding. All authors contributed to writing and approved the final manuscript.

REFERENCES

[1] Markov PV, Ghafari M, Beer M, Lythgoe K, Simmonds P, Stilianakis NI, et al. The evolution of SARS-CoV-2. Nature Reviews Microbiology. 2023;21(6):361-79.
[2] Carabelli AM, Peacock TP, Thorne LG, Harvey WT, Hughes J, de Silva Thushan I 6 CGUC, et al. SARS-CoV-2 variant biology: immune escape, transmission and fitness. Nature Reviews Microbiology. 2023;21(3):162-77.
[3] Oluwagbemi OO, Oladipo EK, Kolawole OM, Oloke JK, Adelusi TI, Irewolede BA, et al. Bioinformatics, computational informatics, and modeling approaches to the design of mRNA COVID-19 vaccine candidates. Computation. 2022;10(7):117.
[4] Azzeri A, Mohamed NA, Wan Rosli SH, Abdul Samat MN, Rashid ZZ, Mohamad Jamali MA, et al. Unravelling the link between SARS-CoV-2 mutation frequencies, patient comorbidities, and structural dynamics. Plos One. 2024;19(3):e0291892.
[5] Añonuevo LE, Lachica ZPT, Amistas DA, Lato JIE, Bontilao HLC, Catalan JMG, et al. Transmission dynamics and baseline epidemiological parameter estimates of Coronavirus disease 2019 pre-vaccination: Davao City, Philippines. Plos one. 2023;18(4):e0283068.
[6] Lv JX, Liu X, Pei YY, Song ZG, Chen X, Hu SJ, et al. Evolutionary trajectory of diverse SARS-CoV-2 variants at the beginning of COVID-19 outbreak. Virus Evolution. 2024;10(1):veae020.
[7] Misawa K, Ootsuki R. A simple method for estimating time-irreversible nucleotide substitution rates in the SARS-CoV-2 genome. NAR Genomics and Bioinformatics. 2024;6(1):lqae009.
[8] Napolitano F, Xu X, Gao X. Impact of computational approaches in the fight against COVID-19: an AI guided review of 17 000 studies. Briefings in bioinformatics. 2022;23(1):bbab456.
[9] Zhang X, Wu J, Luo Y, Wang Y, Wu Y, Xu X, et al. CovEpiAb: a comprehensive database and analysis resource for immune epitopes and antibodies of human coronaviruses. Briefings in Bioinformatics. 2024;25(3):bbae183.
[10] Jahshan Z, Yavits L. ViTAL: Vision TrAnsformer based low coverage SARS-CoV-2 lineage assignment. Bioinformatics. 2024:btae093.
[11] Gong Z, Zhu JW, Li CP, Jiang S, Ma LN, Tang BX, et al. An online coronavirus analysis platform from the National Genomics Data Center. Zoological research. 2020;41(6):705.
[12] Yang Q, Song W, Reheman H, Wang D, Qu J, Li Y. PANoptosis, an indicator of COVID-19 severity and outcomes. Briefings in Bioinformatics. 2024;25(3):bbae124.
[13] Moradi M, Golmohammadi R, Najafi A, Moghaddam MM, Fasihi-Ramandi M, Mirnejad R. A contemporary review on the important role of in silico approaches for managing different aspects of COVID-19 crisis. Informatics in Medicine Unlocked. 2022;28:100862.
[14] Cariou M, Picard L, Guéguen L, Jacquet S, Cimarelli A, Fregoso OI, et al. Distinct evolutionary trajectories of SARS-CoV-2-interacting proteins in bats and primates identify important host determinants of COVID-19. Proceedings of the National Academy of Sciences. 2022;119(35):e2206610119.
[15] Ayaz H, Ibrahimzada AR, Adebali O, Fejzullahu A, Azgari C, Baysan M, et al. Predicting Future SARS-CoV-2 Mutations using Deep Learning. bioRxiv. 2025. Available from: https://www.biorxiv.org/content/10.1101/2025.07.12.664533v1.
[16] Saha G, Sawmya S, Saha A, et al. PRIEST: predicting viral mutations with immune escape capability of SARS-CoV-2 using temporal evolutionary information. Briefings in Bioinformatics. 2024;25(3):bbae218.
[17] Wang C, Liu Z, Chen Z, Huang X, Xu M, He T, et al. The establishment of reference sequence for SARS-CoV-2 and variation analysis. Journal of medical virology. 2020;92(6):667-74.
[18] Ulmer A, Angelini M, Fekete JD, Gillmann C, Hauser H, Schulz HJ, et al. A survey on progressive visualization. IEEE Transactions on Visualization and Computer Graphics. 2024 Sep;30(9):6447-67. Available from: https://doi.org/10.1109/TVCG.2023.3346641.
[19] Wilson G, Bryan J, Cranston K, Kitzes J, Nederbragt L, Teal TK. Good enough practices in scientific computing. PLoS Computational Biology. 2017 Jun;13(6):e1005510. Available from: https://doi.org/10.1371/journal.pcbi.1005510.
[20] Navarro Gonzalez J, Zweig AS, Speir ML, Schmelter D, Rosenbloom KR, Raney BJ, et al. The UCSC genome browser database: 2021 update. Nucleic acids research. 2021;49(D1):D1046-57.
[21] Casper J, Speir ML, Raney BJ, Perez G, Nassar LR, Lee CM, et al. The UCSC genome browser database: 2026 update. Nucleic Acids Research. 2026;54(D1):D1331-5.
[22] Fernandes JD, Hinrichs AS, Clawson H, Karolchik D, Lee BT, Nassar LR, et al. The UCSC SARS-CoV-2 Genome Browser. Nature Genetics. 2020 Oct;52(10):991-8. Available from: https://doi.org/10.1038/s41588-020-0700-8.
[23] Buels R, Yao E, Diesh CM, Hayes RD, Munoz-Torres M, Helt G, et al.

JBrowse: a dynamic web platform for genome visualization and analysis. Genome biology. 2016;17(1):1-12.

[24] Diesh C, Stevens GJ, Xie P, De Jesus Martinez T, Hershberg EA, Leung A, et al. JBrowse 2: a modular genome browser with views of synteny and structural variation. Genome biology. 2023;24(1):74.

[25] Akther S, Bezrucenkovas E, Sulkow B, Panlasigui C, Li L, Qiu W, et al. CoV Genome Tracker: tracing genomic footprints of Covid-19 pandemic. bioRxiv. 2020. Available from: https://www.biorxiv.org/content/early/2020/04/14/2020.04.10.036343.

[26] Flynn JA, Purushotham D, Choudhary MNK, Zhuo X, Fan C, Matt G, et al. Exploring the coronavirus pandemic with the WashU Virus Genome Browser. Nature Genetics. 2020 Oct;52(10):986-91. Available from: https://doi.org/10.1038/s41588-020-0697-z.

[27] Yates AD, Austine-Orimoloye O, Azov AG, Barba M, Barnes I, Barrera-Enriquez VP, et al. Ensembl 2026. Nucleic Acids Research. 2026;54(D1):D1053-60.

[28] Hadfield J, Megill C, Bell SM, Huddleston J, Potter B, Callender C, et al. Nextstrain: real-time tracking of pathogen evolution. Bioinformatics. 2018 05;34(23):4121-3. Available from: https://doi.org/10.1093/bioinformatics/bty407.

[29] Thorvaldsdóttir H, Robinson JT, Mesirov JP. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Briefings in Bioinformatics. 2012 04;14(2):178-92. Available from: https://doi.org/10.1093/bib/bbs017.

[30] Donlin MJ. Using the generic genome browser (GBrowse). Current protocols in bioinformatics. 2009;28(1):9-9.

[31] Yu S, Fu Y, Wong JHy, Wang J, Zhao H, Zhao J, et al. The 3D Genome Browser 2.0: an enhanced online platform for visualizing and analyzing 3D genome architecture. Nucleic Acids Research. 2026;54(D1):D48-54.

[32] Thadani NN, Gurev S, Notin P, Youssef N, Rollins NJ, Ritter D, et al. Learning from prepandemic data to forecast viral escape. Nature. 2023 Oct;622(7984):818-25. Available from: https://doi.org/10.1038/s41586-023-06617-0.

[33] Schwerdt J, Tersteegen A, Marquardt P, Kaasch AJ, Nürnberger A. An Explorative Tool for Mutation Tracking in the Spike Glycoprotein of SARS-CoV-2. In: 2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS); 2021. p. 1-6.

[34] Wilde V, Canard B, Ferron F. Viral Instant Mutation Viewer (VIMVer) : a tool to speed up the identification and analysis of new SARS-CoV2 emerging variant and beyond. Viruses. 2023 Aug;15(8):1628. Available from: https://amu.hal.science/hal-04149011.

[35] Cedeño-Pérez LF, Gómez-Romero L. CovDif, a Tool to Visualize the Conservation between SARS-CoV-2 Genomes and Variants. Viruses. 2022;14(3). Available from: https://www.mdpi.com/1999-4915/14/3/561.

[36] Alisoltani A, Jaroszewski L, Godzik A, Iranzadeh A, Simons LM, Dean TJ, et al. ViralVar: A Web Tool for Multilevel Visualization of SARS-CoV-2 Genomes. Viruses. 2022;14(12). Available from: https://www.mdpi.com/1999-4915/14/12/2714.

[37] Liu B, Liu K, Zhang H, Zhang L, Bian Y, Huang L. CoV-Seq, a New Tool for SARS-CoV-2 Genome Analysis and Visualization: Development and Usability Study. J Med Internet Res. 2020 Oct;22(10):e22299. Available from: https://www.jmir.org/2020/10/e22299.

[38] Cacciabue M, Aguilera P, Gismondi MI, Taboga O. Covidex: An ultrafast and accurate tool for SARS-CoV-2 subtyping. Infection, Genetics and Evolution. 2022;99:105261.

[39] Mercatelli D, Triboli L, Fornasari E, Ray F, Giorgi FM. Coronapp: A web application to annotate and monitor SARS-CoV-2 mutations. Journal of Medical Virology. 2021;93(5):3238-45. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/jmv.26678.

[40] Khare S, Gurry C, Freitas L, Schultz MB, Bach G, Diallo A, et al. GISAID's role in pandemic response. China CDC Weekly. 2021;3(49):1049.

[41] Khare S, Gurry C, Freitas L, Schultz MB, Bach G, Diallo A, et al. GISAID's role in pandemic response. China CDC weekly. 2021;3(49):1049.

[42] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 30; 2017. Available from: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[43] Han W, Chen N, Xu X, et al. Predicting the antigenic evolution of SARS-CoV-2 with deep learning. Nature Communications. 2023;14(1):3982.

[44] Elkin ME, Zhu X. Paying attention to the SARS-CoV-2 dialect: a deep neural network approach to predicting novel protein mutations. Communications Biology. 2024;7(1):7262.

[45] Ma E, Guo X, Hu M, et al. A predictive language model for SARS-CoV-2 evolution. Signal Transduction and Targeted Therapy. 2024;9(1):353.

[46] Zaharia M, Xin RS, Wendell P, Das T, Armbrust M, Dave A, et al. Apache Spark: a unified engine for big data processing. Communications of the ACM. 2016 Oct;59(11):56-65. Available from: https://doi.org/10.1145/2934664.

[47] Ahmad T, Nawab F, Paton NW. Benchmarking Apache Arrow Flight: a wire-speed protocol for batch and streaming data. In: Proceedings of the ACM International Conference on Management of Data (SIGMOD); 2022. Available from: https://doi.org/10.1145/3527199.3527264.

[48] Chen S, Thaduri UR, Ballamudi VKR. Front-end development in react: an overview. Engineering International. 2019;7(2):117-26.

[49] Johnson R. Practical Redux Engineering: Definitive Reference for Developers and Engineers. HiTeX Press; 2025.

[50] Da Rocha H. Learn Chart. js: Create interactive visualizations for the web with chart. js 2. Packt Publishing Ltd; 2019.

[51] Satyanarayan A, Moritz D, Wongsuphasawat K, Heer J. Vega-Lite: a grammar of interactive graphics. IEEE Transactions on Visualization and Computer Graphics. 2017 Jan;23(1):341-50. Available from: https://doi.org/10.1109/TVCG.2016.2599030.

[52] Suhendar A, Razani MN, Sutowo MHA, Pratama NFP, Darmawan ZFAF, Ambo SN, et al. Fast and Beautiful Web Development Using React and Tailwind Css. Dedication: Jurnal Pengabdian Masyarakat. 2026;10(1):21-32.

[53] Bhat K. Ultimate Tailwind CSS Handbook: Build sleek and modern websites with immersive UIs using Tailwind CSS (English Edition). Orange Education Pvt Ltd; 2023.

[54] Rambaut A, Holmes EC, O'Toole Á, Hill V, McCrone JT, Ruis C, et al. A dynamic nomenclature proposal for SARS-CoV-2 lineages to assist genomic epidemiology. Nature Microbiology. 2020 Nov;5:1403-7. Available from: https://doi.org/10.1038/s41564-020-0770-5.

[55] Atul Banwar. List Virtualization in React. 2023.

[56] Crooks GE, Hon G, Chandonia JM, Brenner SE. WebLogo: a sequence logo generator. Genome research. 2004;14(6):1188-90.

[57] Tareen A, Kinney JB. Logomaker: beautiful sequence logos in Python. Bioinformatics. 2020;36(7):2272-4.

[58] Thakur P, Jadon P. Django: Developing web using python. In: 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE). IEEE; 2023. p. 303-6.

[59] Hillar GC. Django RESTful Web Services: The Easiest Way to Build Python RESTful APIs and Web Services with Django. Packt Publishing Ltd; 2018.

[60] Manaswi NK. Understanding and working with Keras. In: Deep learning with applications using Python: Chatbots and face, object, and speech recognition with TensorFlow and Keras. Springer; 2018. p. 31-43.

[61] Abadi M, et al. TensorFlow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI); 2016. p. 265-83.

[62] Min S, Lee B, Yoon S. Deep learning in bioinformatics. Briefings in bioinformatics. 2017;18(5):851-69.

[63] Raza K, et al. A survey of k-mer methods and applications in bioinformatics. Computational and Structural Biotechnology Journal. 2024;23:2289-303.

[64] Orozco-Arias S, et al. K-mer-based machine learning method to classify LTR-retrotransposons in plant genomes. PeerJ. 2021;9:e11456.

[65] Dayhoff MO, Schwartz RM, Orcutt BC. A model of evolutionary change in proteins. In: Atlas of Protein Sequence and Structure. vol. 5. National Biomedical Research Foundation; 1978. p. 345-52.

[66] Hancock JT, Khoshgoftaar TM. Survey on categorical data for neural networks. Journal of Big Data. 2020;7(1):1-41.

[67] Potdar K, Pardawala TS, Pai CD. A comparative study of categorical variable encoding techniques for neural network classifiers. International Journal of Computer Applications. 2017;175(4):7-9.

[68] Pedregosa F, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 2011;12:2825-30.

[69] Zheng H, Vishwanath V, Koziol Q, et al. HDF5 Cache VOL: Efficient and scalable parallel I/O through caching data on node-local storage. In: 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid). IEEE; 2022. p. 61-70.

[70] Chen C, Nadeau S, Yared M, Voinov P, Xie N, Roemer C, et al. CoV-Spectrum: analysis of globally shared SARS-CoV-2 data to identify and characterize new variants. Bioinformatics. 2022;38(6):1735-7.

[71] Brooke J, et al. SUS-A quick and dirty usability scale. Usability evaluation in industry. 1996;189(194):4-7.

[72] Lewis JR, Sauro J. Item benchmarks for the system usability scale. Journal of User Experience. 2018;13(3).

[73] Hyzy M, Bond R, Mulvenna M, Bai L, Dix A, Leigh S, et al. System usability scale benchmarking for digital health apps: meta-analysis. JMIR mHealth and uHealth. 2022;10(8):e37290.

[74] Lewis JR. The System Usability Scale: past, present, and future. International Journal of Human–Computer Interaction. 2018 Feb;34(7):577-90. Available from: https://doi.org/10.1080/10447318.2018.1455307.

[75] Hertzum M. System Usability Scale: a meta-analysis of how SUS is used and what it measures. International Journal of Human–Computer Interaction. 2026. Available from: https://doi.org/10.1080/10447318.2026.2625260.