Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Interpretable Predictive Modeling of Host Specificity of Rabies lyssavirus from Glycoprotein Sequences

Arjun Rao

arjun041008@gmail.com

The Hyderabad Public School https://orcid.org/0009-0001-2323-0664

Research Article

Keywords: RABV, Saliency Maps, ProtBERT, LSTM

Posted Date: March 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9015851/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Interpretable Predictive Modeling of Host Specificity of *Rabies lyssavirus* from Glycoprotein Sequences

Arjun Rao

Abstract

Rabies lyssavirus (RABV) is a zoonotic virus that affects a wide range of mammalian hosts and has a 100% mortality rate following the onset of symptoms. Let $s = (a_1, \ldots, a_L)$ denote the amino acid sequence of the RABV G protein and let $y \in \mathcal{Y}$ denote the host isolate. This study learns a family of parameterized mapping functions $f_\theta : \mathcal{X} \to \Delta^{|\mathcal{Y}|}$ that map an encoded sequence $x = \phi(s) \in \mathcal{X}$ to a predictive host probability vector $\hat{\boldsymbol{p}} = f_\theta(x)$. Among the learned mappings, the long short-term memory (LSTM) mapping composed with an integer-encoding operator achieved the highest test accuracy of 90.0%. Additionally, interpretability analyses of f_θ recover known antigenic sites and identify a strong association between residue positions 460–520 and host specificity. Overall, the learned maps and their sensitivity analyses provide groundwork toward targeted vaccines and containment.

1 Introduction

Rabies lyssavirus (RABV) is a neglected infection caused by a single-stranded RNA virus of the family Rhabdoviridae and is responsible for approximately 60,000 human deaths annually [1]. It affects mammals and is typically transmitted via bites or scratches from infected animals. Approximately 95% of human RABV deaths occur after contact with an infected dog; however, chiropterans (bats), Mephitidae (skunks and stink badgers), *Procyon lotor* (raccoons), and *Vulpes* spp. (foxes) are common reservoirs, especially in North America where most deaths are attributed to bat-associated variants [1]. Further adding to the severity of the virus is its near-100% case fatality rate once symptoms appear, making post-exposure prophylaxis the only effective way to prevent death [1].

RABV consists of five structural proteins: 1) nucleoprotein (N), 2) phosphoprotein (P), 3) matrix protein (M), 4) glycoprotein (G), and 5) large subunit (L). The nucleoprotein is relatively conserved, whereas the surface-exposed G protein contains the ectodomain (including antigenic sites) and is a primary target of neutralizing antibodies [2]. This property makes the G protein particularly informative for host-specific variation.

As an RNA virus, RABV has a high mutation rate due to low replication fidelity of its RNA-dependent RNA polymerase. This influences viral pathogenicity and invasiveness [3]. Cross-species transmission can further increase genetic diversity in RABV—particularly in the G protein—and drive phenotypic variation across host species [4].

These properties of the G protein make it possible to predict the host species from the G protein sequence. A better understanding of virus–host interactions can support the design of more effective vaccines for specific viral quasispecies, whose efficacy can vary across mutations. This, in turn, can enable improved containment strategies in *Homo sapiens* and other mammals.

Current research on RABV host specificity often predicts reservoirs based on physical characteristics of species using logistic regression and other machine learning (ML) techniques [5]. Other work stud-

ies host-shift events to characterize spread from a common host such as *Canis lupus familiaris* to less common hosts such as *Vulpes vulpes* [6]. In contrast, the use of proteomic sequences for host prediction remains in its infancy in RABV research.

Multiple instance learning has proven successful for viral host prediction, as shown by Liu et al. in the development of EvoMIL. Although an AUC of 0.92 is promising, most targeted hosts were prokaryotic, and Liu et al. noted that eukaryotic host prediction is more difficult [6]. Beyond this example of deep learning, most conventional methods use the BLAST framework, which analyzes similarity between a query proteome and sequences in a reference database; this can be ineffective for novel RABV strains. Other attempts at host classification using machine learning focus on broad sets of hosts and bacteriophages, limiting residue-level analysis of proteomic factors affecting host specificity [7, 8].

In the context of RABV, host prediction will allow effective containment of the virus and development of post-exposure prophylaxis vaccines to minimize the impact of human-animal conflict. From an applied-mathematics perspective, this task can be posed as learning a mapping $f : \mathcal{X} \to \mathcal{Y}$ (or, probabilistically, $f : \mathcal{X} \to \Delta^{|\mathcal{Y}|}$) from an encoded protein sequence space $\mathcal{X}$ to a host label space $\mathcal{Y}$. Understanding host–protein interaction through analysis of the learned mapping (e.g., residue-wise sensitivity) improves mechanistic insight into residue-level contributions to host specificity across hosts. Thus, this study as opposed to relying on black-box ML models, proposes new interpretability pipelines that can be used to generate hypotheses on factors that determine protein-host dynamics.

2 Background

2.1 Protein-Host Interaction

Invading viruses must interact with host cells, as they depend on the host's metabolic processes to replicate and invade additional cells. In *Rabies lyssavirus* (RABV), this interaction is primarily mediated by the glycoprotein (G protein), which is located on the surface of the virus and consists of 524 amino acid residues. Due to its exposed position, the G protein plays a critical role in binding to host receptors and antibodies, making it a key determinant of host specificity and immune recognition.

There are three main antigenic sites on the G protein where neutralizing antibodies bind: antigenic site I (residues 226–231), antigenic site II (residues 34–42 and 198–200), and antigenic site III (residues 330–338) [9]. Mutations within these regions, as well as in other parts of the G protein, have been shown to significantly affect the pathogenicity of different RABV strains. Takayama-Ito et al. demonstrated that the region between residues 164 and 303 plays an important role in RABV lethality in adult mice, concluding that pathogenicity is influenced by the combined effect of multiple amino acid residues rather than a single mutation [10].

Additional studies highlight the importance of specific residue positions, such as 255 and 333, in modulating viral pathogenicity. For instance, the D225G mutation was shown to reduce neurotropism in adult mice, leading to observable differences in disease progression [11]. Changes in pathogenicity can also manifest as altered permeability of the blood–brain barrier, while other mutations have been shown to attenuate virulence [7]. A broad pattern that emerges from these studies is that many functionally important residues are located within the antigenic sites of the G protein, which themselves are situated on the ectodomain of the protein.

The identification of these critical residues has largely relied on wet-lab techniques such as flow cytometry and in vitro assays measuring the growth and spread of RABV-infected cells. In contrast, computational approaches have not yet been widely used to model host–protein interactions or to systematically quantify residue-level importance at scale. Developing such computational frameworks

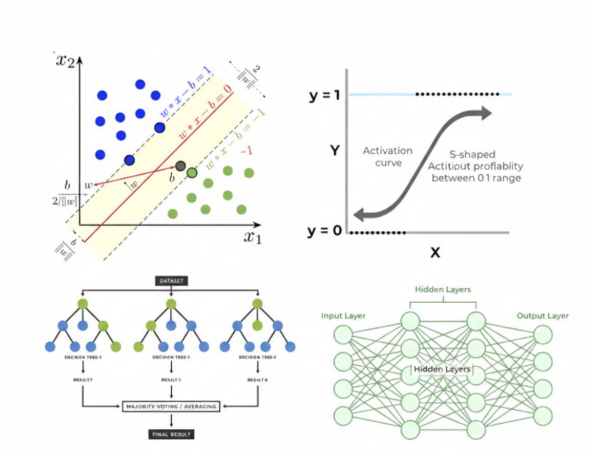


Figure 1: Architecture of Support Vector Machines, Logistic Regression, Random Forest Classifier and Artificial Neural Network. These models use various algorithmic methods to discriminate between various host sequences.

is particularly relevant in a clinical context, as they can inform the design of protein-subunit vaccines, which aim to induce immunity through the expression of the G protein [9]. Because vaccines against RABV are often species-specific, mutations in the G protein sequence can directly impact vaccine efficacy against particular viral strains, underscoring the need for a deeper residue-level understanding of G protein function [12].

2.2 Protein Embeddings

While using machine and deep learning models on protein sequences, it is necessary to convert the sequence into a form that can be used by these models for training. The process of converting protein sequences to numerical vectors can be done using hot encoding of the protein or using embeddings. Hot encoding is a straightforward way of converting the protein sequence into a binary vector or encoding the amino acids with a numerical value between 1 and 20 [13]. While this is the simpler strategy, hot encoding does not fully capture the relationship between the various amino acids.

The other strategy would involve using unsupervised learning to generate embeddings that capture the relation between the amino acids in proteins [14]. This method involves using processes from the field of Natural Language processing (NLP) such as k-mers splitting of protein sequence and production of embeddings based on the similarity of protein sequences [15]. Unsupervised protein embeddings have proven to greater accuracy in machine learning models trained on them compared to traditional methods [16].

This study makes use of the embedding model PROTBert, which was proposed by Brandes et al. and makes use of the BERT (Biderectional Encoder Representations from Transformers) architecture that is used in NLP. The main advantage of this model is that it uses local and global representations in embedding the data and does not impose any limits on the length of the amino acid sequence [17]. This makes this model both versatile and makes it better learn rules for embedding proteins. The use of PROTBert for detecting transcription factors along with Support Vector Machines has shown better performance than the use of other probabilistic vector representations [18]. Overall, while literature on PROTBert is limited the architecture of the model and the performance of BERT on NLP tasks, makes it a reliable choice for RABV host prediction.

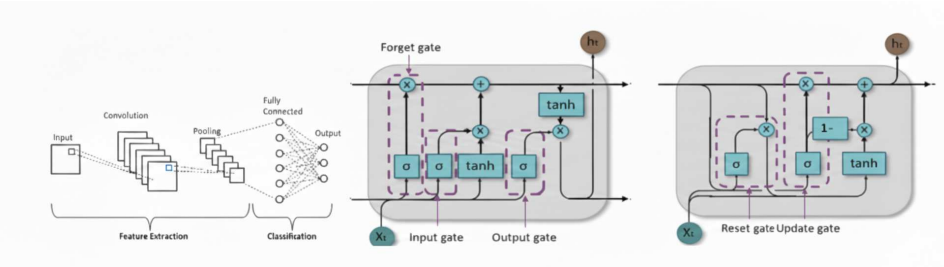


Figure 2: Architecture of CNN, LSTM and GRU. The CNN consists of multiple convolutional layers, the LSTM and GRU have reset, update and forget gates to iteratively learn patterns in sequences.

2.3 Predictive Model Architecture

This study, aiming to develop a pipeline for predicting the host given the protein sequence of the G protein, trains a range of machine and deep learning models, and this section aims to summarize the various model architectures, as seen in Figure 1. In terms of machine learning, logistic regression is a commonly used method for binary classification where the linear regression equation is passed through the sigmoid function to obtain a set of probabilities. In the case of multinomial logistic regression, the regression equation is passed through a softmax equation to model probabilities across multiple classes [19]. Random Forests are another popular model for classification tasks, that use multiple decision trees to split the classification problem into a series of simpler tasks [19]. Support Vector Machines are an ML model that separate the data by drawing an artificial hyperplane to separate the data and are regularly used in protein classification tasks [20].

While these machine learning models can make predictions, it is difficult for models to train on data with a large number of dimensions, like embeddings, due to increased computational load and the sparsity of data, preventing the models from detecting patterns [21]. For such dataset, dimensionality reduction techniques like Principal Component Analysis (PCA) are used. PCA involves computing the top eigenvectors of the covariance matrix of the given data, thus reducing the number of dimensions while also broadly representing the patterns in data [22].

Dimensionality is not a very significant issue, however, in the context of deep learning, Specific neural networks, called Recurrent Neural Networks (RNNs), like Gated Recurrent Unit (GRU) and Long-Short Term Memory (LSTM) models allow the processing the prediction of sequential data [23]. LSTMs are models that learn the training parameters in the context of the previous hidden cell. These models consist of a forget, input and output gate for each neural unit of the cell, which removes irrelevant data, takes in new data and send the learned parameters to the next unit [24]. A GRU has a simpler architecture than an LSTM, as it only consists of two gates. The reset gate determines how much of the data from the previous hidden unit is to be forgotten and the update gate determines how much data from the current unit is to be sent to the next unit [25].

GRUs and LSTMs are commonly used in NLP and are other techniques that have been adapted for protein classification. Sharma and Maheshwari (2024) have successfully shown the usage of Convolutional Neural Networks (CNNs) for protein-related tasks, the architecture of which are in Figure 2. These models are typically used for image classification tasks, and in the context of protein, they model the spatial relationship between the protein tensors that are produced by the convolutional layers [26]. Overall, the architectures of these machine and deep learning models indicates the potential for them to perform optimally in tasks like host prediction.

2.4 Mathematical formalism and interpretability operators

Host prediction, in this study, is viewed as learning a composition $f_\theta \circ \phi$ that maps an amino-acid sequence $s = (a_1, \ldots, a_L) \in \mathcal{A}^L$ (with $|\mathcal{A}| = 20$) to a host distribution in $\Delta^{|\mathcal{Y}|}$. This composition is built from a sequence-to-vector encoding stage and a learned predictor; therefore, we first formalize encodings. A one-hot encoding operator ϕ_{OH} maps each residue a_i to $\boldsymbol{x}_i \in \{0,1\}^{20}$ with $(\boldsymbol{x}_i)_k = \mathbf{1}[a_i = \alpha_k]$, and the whole sequence to $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L] \in \{0,1\}^{20\times L}$ [13]. Here α_k denotes the kth amino acid in a fixed ordering and $\mathbf{1}[\cdot]$ is the indicator function. Alternatively, an integer encoding operator ϕ_{int} maps $a_i \mapsto e(a_i) \in \{0, \ldots, 19\}$ and yields a categorical sequence $\boldsymbol{e} \in \{0, \ldots, 19\}^L$ [13].

To incorporate contextual information across residues, transformer-based embeddings are used; abstractly, ProtBERT defines

$$\boldsymbol{h} = \phi_{\mathrm{BERT}}(s) \in \mathbb{R}^{1024}, \tag{1}$$

where $\boldsymbol{h}$ is a fixed-length vector representation produced by a transformer encoder [17]. Because $\boldsymbol{h}$ is high-dimensional, PCA is applied as a linear projection,

$$\boldsymbol{z} = \phi_{\mathrm{PCA}}(\boldsymbol{h}) = W_k^\top(\boldsymbol{h} - \boldsymbol{\mu}), \qquad W_k \in \mathbb{R}^{1024\times k}, \tag{2}$$

where $\boldsymbol{\mu}$ is the empirical mean embedding and W_k contains the top k principal directions (eigenvectors of the sample covariance) [22].

For interpretability, learned internal representations (e.g., hidden states) are visualized by manifold learning. In UMAP, low-dimensional points $\{\boldsymbol{y}_i\}_{i=1}^n \subset \mathbb{R}^d$ are learned by minimizing a cross-entropy-type objective between fuzzy neighbor graphs in the original and embedded spaces [27, 28, 29]:

$$\mathcal{L}_{\mathrm{UMAP}}(\{\boldsymbol{y}_i\}) = \sum_{i<j} \Big[w_{ij} \log \frac{w_{ij}}{\hat{w}_{ij}} + (1 - w_{ij}) \log \frac{1 - w_{ij}}{1 - \hat{w}_{ij}} \Big]. \tag{3}$$

This objective is typically minimized with stochastic gradient descent (SGD) [30], which (for generic parameters θ) updates

$$\theta_{t+1} = \theta_t - \eta \, \nabla_\theta \ell(\theta_t; \mathcal{B}_t), \tag{4}$$

for stepsize η and mini-batch $\mathcal{B}_t$.

Finally, residue-level importance can be quantified by saliency. For an input encoding x and class score $S_c(x)$, a standard saliency signal is

$$\boldsymbol{s}_c(x) = \nabla_x S_c(x), \tag{5}$$

so that large-magnitude components indicate input coordinates that most strongly affect the score locally [31].

3 Methodology

3.1 Data Collection

Data used to train the machine and deep learning models was collected from the National Center for Biotechnology Information (NCBI) Virus Database [32]. For the dataset, those G protein sequences of RABV were selected in which the host was known, and only those hosts were considered in which the number of FASTA protein sequences per host was greater than 100. Additionally, to reduce the number of host classes, the species name was replaced with the genus, as in combining Vulpes vulpes and Vulpes lagopus to create the Vulpes class. The final dataset of 6764 protein sequences consisted of the following hosts: 1) Bos taurus (domestic cattle), 2) Canidae (dog-like mammals except the domestic dog and foxes), 3) Canis lupus familiaris (domestic dog), 4) Eptesicus fuscus (big brown

bat), 5) Felis catus (cat), 6) Melogale (ferret badger), 7) Mephitidae (skunks and stink badgers), 8) Procyon lotor (raccoon), 9) Homo sapiens (humans), and 10) Vulpes (foxes). Canis lupus familiaris was not classified under Canidae, as it is the most common host of the virus, and under-sampling was not performed to enable the model to understand Canis-specific features in the G proteome.

3.2 Preprocessing

After obtaining the data, standard data hygiene practices were followed to remove null values. To prevent phylogenetic leakage that might inflate accuracy of the model, clustering of sequences was performed at 95% across the training and testing datasets. For learning the host mapping $f_\theta \circ \phi$, two preprocessing operators ϕ were used to trade off accuracy and interpretability. In the first pipeline, the FASTA sequence s was mapped to a dense feature vector $\boldsymbol{h} = \phi_{\text{BERT}}(s) \in \mathbb{R}^{1024}$ using ProtBERT. Since $\boldsymbol{h}$ is high-dimensional, PCA was applied to obtain a low-dimensional representation $\boldsymbol{z} = \phi_{\text{PCA}}(\boldsymbol{h}) \in \mathbb{R}^{10}$. The resulting features $\boldsymbol{z}$ were then used to learn three downstream mapping functions f_θ (logistic regression, random forest, and support vector classification). In addition to this pipeline, a second pipeline was used because PCA compression reduces residue-level interpretability.

The other preprocessing method involved converting each position of the G protein amino acid sequence into a column and encoding the amino acids alphabetically as an integer between 0 and 19. The issue of having a large number of dimensions in that dataset was overcome by using deep learning models that can train on these models. The advantage conferred by this method is that it allows greater interpretability at a residue level.

3.3 Model Training

For both pipelines, the target variable was the host $y \in \mathcal{Y}$, and an 80/20 training–testing split was used. Each trained predictor is represented as a composition

$$\hat{\boldsymbol{p}}(\cdot \mid s) = f_\theta(\phi(s)) \in \Delta^{|\mathcal{Y}|}, \qquad \hat{y} = \arg\max_{c \in \mathcal{Y}} \hat{p}(c \mid s), \tag{6}$$

where ϕ is a preprocessing/encoding operator (e.g., ProtBERT→PCA or integer encoding) and f_θ is the learned map.

The three machine learning mapping functions used on PCA-transformed embeddings were multinomial logistic regression, random forest, and support vector classification. The simplicity of these maps is also a good fit for PCA features, which are linear projections. In terms of the deep learning architectures, the convolutional neural network used for training the label-encoded data consisted of one embedding layer to reduce the dimensionality of the data and two one-dimensional convolutional layers with 64 and 128 units, respectively. A max pooling layer was used between the convolutional layers to reduce the spatial dimensions of the tensor. For compilation, the Adam optimizer was used with the sparse categorical cross-entropy loss function [33]. This CNN was trained on 15 epochs, which was optimized for maximum accuracy.

The architecture of the GRU and LSTM used in this study is very similar. Both consist of an embedding layer and the recurrent layers of 128 units followed by a dense layer with 64 units and the output layer with softmax activation. Dropout was also used to prevent overfitting in this model by deactivating 30% of the units, and the model was trained for 5 epochs. This architecture is in line with that used in literature and ensures optimal performance by targeting common pitfalls like high dimensionality and overfitting.

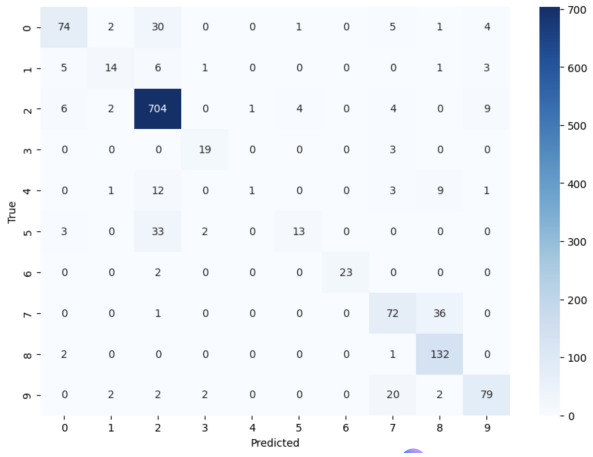


Figure 3: Confusion Matrix of CNN. This graphically represents the relation between the model's predicted output and the true output for each class.

4 Results

4.1 Model Classification Reports

In comparing the accuracy of the various models, precision, recall, F1, and accuracy scores were considered. Additionally, as the dataset was not balanced, it is also important to look at the weighted average accuracy and host-specific F1 scores, which account for the differences in the class size. For a given host class, let TP, FP, TN, and FN denote the entries of the one-vs-rest confusion matrix (true positives, false positives, true negatives, and false negatives). Then the standard metrics are [34]

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{8}$$

$$\text{F1} = \frac{2\,\text{Precision}\,\text{Recall}}{\text{Precision} + \text{Recall}}, \tag{9}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{10}$$

To aggregate across hosts, we report the weighted average metric (weighted by class support), e.g.

$$\text{F1}_{\text{weighted}} = \sum_{c \in \mathcal{Y}} \frac{n_c}{\sum_{c' \in \mathcal{Y}} n_{c'}} \text{F1}_c, \tag{11}$$

where n_c is the number of test samples in class c and F1_c is the class-wise F1 score. To analyze host-specific accuracy, the weighted average scores coupled with the confusion matrix represent the ability of the model to correctly predict a particular host.

In the first model pipeline, where the three machine learning models were trained on PCA-transformed protein embeddings, the Random Forest model had the highest accuracy. The Random Forest Classifier had an overall accuracy of 83.0% compared to the Logistic Regression's 71.0% and the Support Vector Classifier's 79.0% accuracy. All three models showed greater ability in being able to predict Canis lupus familiaris due to a greater number of data points and less accuracy in predicting hosts like Canidae and Felis catus, for which there was less training data, as can be seen in Table 1. It's also

worth pointing out that an ANN was trained on this pipeline; however, it was unable to make predictions, possibly due to vanishing gradients causing the performance to not improve with greater training.

Under the second pipeline, where the amino acid sequences were integer encoded, the CNN had an accuracy of 84.0% and a weighted average accuracy of 82.0%, with further model performance analysis as seen in Figure 3. The LSTM had the highest accuracy of 90.0% and weighted average accuracy of 87.0%, indicating that it was the most successful at predicting the host. While the GRU had a validation accuracy of 84.0%, the model underfit considerably, as it had a training accuracy of 76.3%, and the weighted average accuracy was also considerably lower than the macro accuracy. For this reason, the GRU was not used for subsequent analysis.

4.2 Baseline Comparison

To baseline the accuracy results obtained from the two pipelines used, classical bioinformatics methods were also used. A k-mer logistic regression model was used as a baseline comparison that draws linear boundaries between tokens formed by the combination of amino acid sequences. This model assigns a bias matrix defined by $Host = \beta^T x$. This model achieved an accuracy of 82.0% and weighted F1 score of 81.0%. The LSTM and Random Forest Model achieved a greater accuracy and weighted F1 score that is significant given that the original data was imbalanced. This indicates that host specificity depends on a combination of sequence ordering and is partly compositional. Overall, the models proposed in this study indicate that they are able to better learn features contributing to host specificity than k-mer classical models.

4.3 In-silico Mutagenesis

In silico mutagenesis is a computational technique used to determine the change in the total free energy of a protein that occurs because of a mutation in the amino acid sequence [35]. Here, it is used as a perturbation analysis of a learned mapping $f_\theta \circ \phi$. Let $s \in \mathcal{A}^L$ be a sequence and let $\mathcal{M}_{i\to b}(s)$ denote the sequence obtained by mutating position i to residue $b \in \mathcal{A}$. Define the predicted host distribution $\hat{\boldsymbol{p}}(s) = f_\theta(\phi(s)) \in \Delta^{|\mathcal{Y}|}$. A mutation sensitivity score at position i can be written as

$$\Delta_i(b) = D\big(\hat{\boldsymbol{p}}(s), \hat{\boldsymbol{p}}(\mathcal{M}_{i\to b}(s))\big), \tag{12}$$

where $D(\cdot,\cdot)$ is a divergence/distance between probability vectors (e.g., a coordinate-wise difference in a single host probability, a cosine distance, an ℓ_2 distance, or KL divergence). Intuitively, $\Delta_i(b)$ measures how much the learned host map changes under a localized perturbation (a single-residue mutation) at site i.
In this study, the Random Forest mapping was used to compute $\hat{\boldsymbol{p}}$ (highest overall accuracy). The change in probability was quantified using a simple delta, cosine similarity, KL divergence and ℓ_2 distance. The delta subtracts the probability that a sequence belongs to a particular host from the probability after mutation. Cosine similarity compares probability vectors based on their angle [36], the ℓ_2 score (Euclidean distance) measures their distance [37], and KL divergence compares distributions [38]. These are given by equations (13)–(16) respectively.

$$\Delta P = P_{new} - P_{old} \tag{13}$$

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{14}$$

$$D_{KL}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{15}$$

$$\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2} \tag{16}$$

In Equation (13), P_{old} and P_{new} denote the scalar predicted probability of a given host class before and after mutation. In Equation (14), $\mathbf{A}$ and $\mathbf{B}$ denote two host-probability vectors (e.g., $\hat{\boldsymbol{p}}(s)$ and $\hat{\boldsymbol{p}}(\mathcal{M}_{i\to b}(s))$), and $\cos(\theta)$ measures their angular similarity [36]. In Equation (15), $D_{KL}(P\|Q)$ quantifies the information-theoretic discrepancy between two distributions P and Q [38]. In Equation (16), $\|\mathbf{x}\|_2$ denotes the Euclidean norm of a vector, which induces an ℓ_2 distance between probability vectors [37].
In this study, an alanine scan has been used, which involves converting the amino acid at a given residue position to alanine, which is chemically stable and inert, to determine the importance of that amino acid in the context of host specificity [39]. For most residues a standard alanine-scan was performed but for serine was mutated to cysteine, to ensure that the mutation is conservative and the hydroxyl group is replaced by the thiol group with similar polarity [40].

For statistical testing, 4 random mutations were made in each of the four antigenic regions (the discontinuous antigenic site II was split into 2 antigenic regions), and the perturbation magnitude (e.g., ΔP) was computed for each mutation. This can be framed as testing

$$H_0 : \text{ median}(\Delta P) = 0 \qquad \text{vs.} \qquad H_1 : \text{ median}(\Delta P) \neq 0, \tag{17}$$

using a Wilcoxon signed-rank (single-sample) test [41]. Let d_i be the nonzero signed differences (e.g., $d_i = \Delta P_i$) and let R_i denote the rank of $|d_i|$ among $\{|d_j|\}$. The statistic

$$W = \sum_{i:d_i>0} R_i \tag{18}$$

sums the ranks of positive differences; under H_0 the signs are (approximately) symmetric, so unusually large or small values of W indicate a shift away from zero median.
To compare antigenic regions, a two-sample Mann–Whitney U test (rank-sum) can be used to test whether one region tends to yield larger perturbations than another [42]. These results support that perturbations within known antigenic sites induce statistically significant changes in $\hat{\boldsymbol{p}}(s)$, aligning the learned mapping with prior biological literature.

4.4 Analysis of Hidden States

A hidden state in an RNN represents the short-term memory of the network. Among the RNNs trained using the integer encoding pipeline, the LSTM had the highest accuracy metrics, and therefore, it was used to analyze interpretability. UMAP was used to reduce the dimensionality of the hidden feature vectors and visualize it. Clustering of hidden states representing the same class indicates that the machine can accurately discriminate between the classes. Figure shows the visualized hidden states of the LSTM, which indicates high predictive power in determining the Canis lupus familiaris class. Due to the imbalance in the classes, there is an overrepresentation of some classes, and certain classes are not represented as they get lost during dimensionality reduction. The machine's map also visibly shows signs of misclassifying certain undersampled classes with dominant classes, which concurs

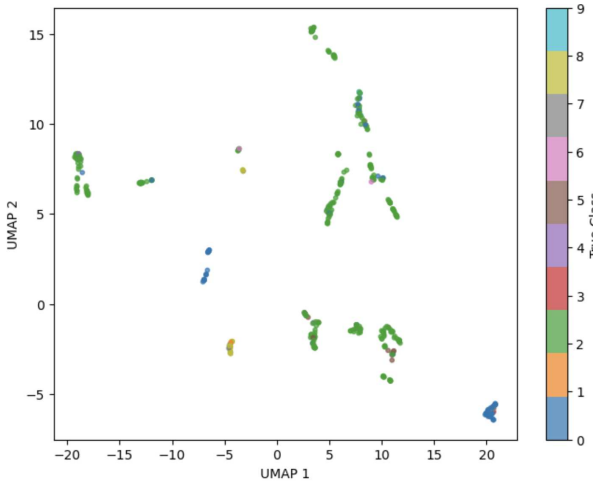


Figure 4: Hidden State Analysis of LSTM Hidden States. There is clustering of the dots attributed to a particular host class and the Canis class shows various separate clustering patterns.

with the findings from host-specific F1 scores. Another feature of the figure is that there are many separate areas of clustering of the Canis lupus familiaris class, indicating that the hidden states are also representing other differences in the proteins. These significant differences could be related to the properties of the proteins, such as hydrophobicity, solubility, or structure.

4.5 Saliency Map Analysis

Saliency maps have been used in this study to determine the importance of a single residue position in determining the host. This analysis was performed on the CNN even though it had lesser accuracy than the LSTM, because saliency maps in CNNs represent global patterns that are useful for determining residue importance, while in LSTMs they represent the feature importance for the hidden state, thus representing only local patterns [31]. Let $g \in \{1, \ldots, G\}$ index host groups and let $S_{g,j}$ denote a residue-wise saliency summary (e.g., average saliency at residue j for samples from host g). A Kruskal–Wallis test can be framed as

$$H_0 : S_{1,j}, \ldots, S_{G,j} \text{ have a common distribution} \quad \text{vs.} \quad H_1 : \text{ at least one differs}, \tag{19}$$

with test statistic (in terms of ranks) [43]

$$H = \frac{12}{N(N+1)} \sum_{g=1}^{G} n_g \big(\bar{R}_g - \bar{R}\big)^2, \tag{20}$$

where $N = \sum_g n_g$ is the total sample size, n_g is the group size, and $\bar{R}_g$ is the mean rank within group g.

To test whether saliency signals exceed what is expected under a null without residue structure, computed saliency values were compared to a randomly permuted baseline [44]. Let d_i denote paired differences between observed and permuted saliency summaries, e.g., $d_i = S_i - \tilde{S}_i$. The Wilcoxon signed-rank test evaluates

$$H_0 : \text{ median}(d) = 0 \qquad \text{vs.} \qquad H_1 : \text{ median}(d) \neq 0, \tag{21}$$

with rank-sum statistic $W = \sum_{i:d_i>0} R_i$ on ranks R_i of $|d_i|$ [41]. Across many residues, p-values were adjusted with the Benjamini–Hochberg procedure [45], which declares significance for ordered p-values

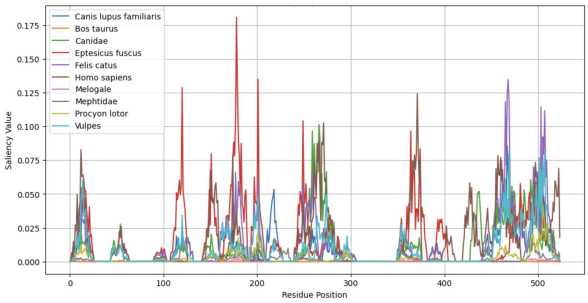


Figure 5: Average Saliency per species. There are distinct patterns in the regions of importance independent of host.

$p_{(1)} \leq \cdots \leq p_{(m)}$ when $p_{(k)} \leq (k/m)q$ for a chosen FDR level q. The combined results indicate that saliency patterns differ by host and depart significantly from randomized baselines.

Of the regular patterns that are seen in the saliency maps, there is a relatively strong saliency value for the residues in the range 460-520. This region, in addition to showing the strongest signals across hosts, also is after the ectodomain, which is typically truncated at 439 [46]. To validate the importance of this region a permutation test was conducted comparing the saliency in comparison to other randomly selected regions over 500 samples, which statistically validates the hypothesis with a p-value of 0.03. This region is the cytoplasmic domain of RABV G protein, which determines the virus's entry into the neurons and interaction with the M protein [9]. Therefore, there is reason to believe that the high saliency value of this region is due to RABV evolving to adapt the most optimal way to enter the host nervous cell which are different to each other. Overall, saliency map analysis identifies new regions of interest and patterns in G protein-host dynamics.

5 Discussion

This study creates a novel pipeline for successfully determining the host from the RABV G protein. This type of predictive modelling is useful for determining the phylogeny of new variants of RABV and effectively containing it. Therefore, this type of model is also successful in the context of the broader One Health, which takes a unified approach to human health by looking at the overall health of the environment [47]. While, human deaths due to RABV have decreased due to greater vaccination of common reservoirs, epizootic spreads are still common and such a model can help easily identify the cause of such a spread [48]. This will be especially effective as the current database is expanded to include a greater number of hosts and the number of protein samples of less common hosts increases. This will not only increase the possible use cases of such a model but also improves the overall accuracy of the model.

In addition to training a new mapping function, the use of XAI in computational virology is also a significant contribution. The results of the learned maps validate existing literature on the importance of antigenic sites in host specificity and add to the literature by identifying new regions. In particular, the region in the range 460–520, which shows a strong signal across all species, also provides preliminary evidence for factors other than antigen–antibody binding affecting host affinity, as it is not located in the ectodomain. The identification of this new region of interest from perturbation and saliency map analysis indicates new regions which contribute to host adaptions in ways that existing RABV literature does not fully emphasize. While these results provide promising early signs, it is still necessary to validate these findings through wet-lab studies, which might also help explain the empirical basis behind these statistical findings. Further, as part of One Health efforts, there is a need

to vaccinate wildlife reservoirs of RABV, and there are no vaccines currently available for bats, which are a common RABV host [49]. Understanding residue-wise contributions to pathogenicity and host specificity will help develop targeted vaccinations for particular hosts such as bats.

In terms of predictive modelling, while attempts at making models more predictable greatly reduce the opaque nature of these models, the first pipeline presented in this study is still not very explainable. The PROTBert embeddings coupled with the PCA lead to this effect. A way to counteract this is to use residue-level embeddings and train a deep RNN. This wasn't done in this study due to the high computational costs of generating residue-wise embeddings. To generalize this idea, in training more complex architectures, it is important to ensure that computational costs are not so high that they diminish the translation potential of the predictive model. The current models do manage to achieve this balance while also being interpretable, which can be considered the most important consideration for most ML models.

6 Conclusion

This study presents a highly interpretable and computationally efficient approach for learning a host-mapping function for a particular RABV strain given the G protein sequence. Sensitivity and interpretability analyses of the learned mappings validate existing literature while also pointing to new patterns that require wet-lab validation. The LSTM-based map achieved the highest accuracy of 90.0%, followed by the CNN and GRU, supporting the utility of deep sequence-to-label mappings in studying protein–host interactions. The results can be used for better containment and for developing immunization strategies against RABV. While this study focuses on RABV, the same mathematical pipeline (encoding ϕ plus learned map f_θ) can be applied to other zoonotic viruses affecting many hosts, supporting broader One Health efforts.

7 Acknowledgments

The author would like to thank the contribution of NCBI and those who contributed to insightful discussions on Protein-Host Interactions.

References

[1] F. Zandi, F. Goshadrou, A. Meyfour, and B. Vaziri, "Rabies infection: An overview of lyssavirus-host protein interactions," *Iranian Biomedical Journal*, vol. 25, no. 4, pp. 226–242, 2021.

[2] P. H. Wang *et al.*, "The roles of rabies virus structural proteins in immune escape," *Canadian Journal of Microbiology*, 2024.

[3] J. A. Conselheiro, G. T. Barone, S. A. T. Miyagi, S. O. de Souza Silva, W. C. Agostinho, J. Aguiar, and P. E. Brandão, "Evolution of rabies virus isolates: Virulence signatures and effects of modulation by neutralizing antibodies," *Pathogens*, vol. 11, no. 12, p. 1556, 2022.

[4] E. M. Bonnaud, C. Troupin, L. Dacheux, E. C. Holmes, E. Monchatre-Leroy, *et al.*, "Comparison of intra- and inter-host genetic diversity in rabies virus during experimental cross-species transmission," *PLOS Pathogens*, vol. 15, no. 6, p. e1007799, 2019.

[5] W. Wang, J. Ma, J. Nie, J. Li, S. Cao, L. Wang, C. Yu, W. Huang, Y. Li, Y. Yu, M. Liang, B. Zirkle, X. S. Chen, X. Li, W. Kong, and Y. Wang, "Antigenic variations of recent street rabies virus," *Emerging Microbes & Infections*, vol. 8, no. 1, pp. 1584–1592, 2019.

[6] C. Boutelle, N. Mollentze, C. Gigante, F. Rocha, M. A. N. Vigilato, D. G. Streicker, and R. Wallace, "A logistic regression model to predict the next rabies virus host-shift event," *Scientific Reports*, vol. 15, no. 1, p. 19306, 2025.

[7] J. Luo, B. Zhang, Y. Wu, and X. Guo, "Amino acid mutation in position 349 of glycoprotein affects the pathogenicity of rabies virus," *Frontiers in Microbiology*, vol. 11, 2020.

[8] A. A. Howell, C. J. Versoza, and S. P. Pfeifer, "Computational host range prediction—the good, the bad, and the ugly," *Virus Evolution*, vol. 10, no. 1, p. vead083, 2024.

[9] Y. You, "Rabies virus glycoprotein: Structure, function, and antivirals," *Viral Engineering*, vol. 10, no. 1, 2025.

[10] M. Takayama-Ito, K. Kawai, K. Yamada, Y. Tohya, and H. Kida, "Multiple amino acids in the glycoprotein of rabies virus are involved in lethality in adult mice," *Virus Research*, vol. 116, no. 1-2, pp. 1–7, 2006.

[11] J. Luo, B. Zhang, Z. Lyu, Y. Wu, Y. Zhang, and X. Guo, "Single amino acid change at position 255 in rabies virus glycoprotein decreases viral pathogenicity," *The FASEB Journal*, vol. 34, no. 7, pp. 9650–9663, 2020.

[12] D.-K. Yang, H.-H. Kim, K.-W. Lee, and J.-Y. Song, "The present and future of rabies vaccine in animals," *Clinical and Experimental Vaccine Research*, vol. 2, no. 1, pp. 19–25, 2013.

[13] H. ElAbd, Y. Bromberg, A. Hoarfrost, T. L. Lenz, *et al.*, "Amino acid encoding for deep learning applications," *BMC Bioinformatics*, vol. 21, no. 1, p. 235, 2020.

[14] K. K. Yang, Z. Wu, C. N. Bedbrook, and F. H. Arnold, "Learned protein embeddings for machine learning," *Bioinformatics*, vol. 34, no. 15, pp. 2642–2648, 2018.

[15] D. Ofer, "The language of proteins: Nlp, machine learning & beyond," *Computational Biology and Chemistry*, vol. 90, p. 107408, 2021.

[16] A. Villegas-Morcillo *et al.*, "Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function," *Bioinformatics*, vol. 37, no. 2, pp. 162–170, 2021.

[17] N. Brandes, D. Ofer, Y. Peleg, N. Rappoport, and M. Linial, "Proteinbert: a universal deep-learning model of protein sequence and function," *Bioinformatics*, vol. 38, no. 8, pp. 2102–2110, 2022.

[18] S. Gaffar *et al.*, "Tfprotbert: Detection of transcription factors binding to methylated dna," *International Journal of Molecular Sciences*, vol. 26, no. –, p. 4234, 2025.

[19] ScienceDirect, "Multinomial logistic regression - an overview," n.d. Accessed: 2025-10-20.

[20] IBM, "What is support vector machine?," n.d. Accessed: 2025-10-20.

[21] DataCamp, "The curse of dimensionality in machine learning: Challenges, impacts, and solutions," 2023. Accessed: 2025-10-20.

[22] ScienceDirect, "Principal component analysis - an overview," n.d. Accessed: 2025-10-20.

[23] Xomnia, "What is the curse of dimensionality, and why does deep learning overcome it?," 2023. Accessed: 2025-10-20.

[24] A. Bisht, "Comprehensive understanding of lstm architecture: A deep dive into long short-term memory networks," 2025. Accessed: 2025-10-20.

[25] Anishnama, "Understanding gated recurrent unit (gru) in deep learning," 2023. Accessed: 2025-10-20.

[26] P. Sharma and M. Maheshwari, "Nlp based protein sequence classification through convolutional neural network," *Educational Administration: Theory and Practice*, vol. 30, no. 1, pp. 2166–2170, 2024.

[27] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[28] M. T. Islam, Z. Zhou, H. Ren, M. B. Khuzani, D. Kapp, J. Zou, L. Tian, J. C. Liao, and L. Xing, "Revealing hidden patterns in deep neural network feature space continuum via manifold learning," *Nature Communications*, vol. 14, 2023.

[29] R. Jing *et al.*, "layerumap: A tool for visualizing and understanding deep neural representations," *Bioinformatics / (or the journal listed)*, 2022.

[30] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *Proceedings of COMPSTAT*, pp. 177–186, 2010.

[31] C. Molnar, *Interpretable Machine Learning*. Christoph Molnar, 2023.

[32] National Center for Biotechnology Information, "Ncbi virus," n.d. Accessed: 2025-10-20.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2015.

[34] T. Tigerschiold, "What is accuracy, precision, recall, and f1 score?," 2022. Accessed: 2025-10-09.

[35] M. Tiberti *et al.*, "Mutatex: an automated pipeline for in silico saturation mutagenesis of protein stability," *Bioinformatics*, vol. 23, no. 3, 2022.

[36] T. Krantz and A. Jonker, "What is cosine similarity?," 2022. Accessed: 2025-10-09.

[37] S. Documentation, "Vector_l2_distance," 2022. Accessed: 2025-10-09.

[38] J. Han, "Kullback-leibler divergence," 2008. Accessed: 2025-10-09.

[39] F. Lefèvre, "Alanine-stretch scanning mutagenesis: a simple and efficient method for identifying functionally important residues in proteins," *Nucleic Acids Research*, vol. 25, no. 2, pp. 447–448, 1997.

[40] J. N. Higaki, "Introduction of a cysteine protease active site into trypsin," *Biochemistry*, vol. 28, no. 23, pp. 9060–9065, 1989.

[41] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[42] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 1947.

[43] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[44] P. I. Good, *Permutation, Parametric, and Bootstrap Tests of Hypotheses*. Springer, 3 ed., 2005.

[45] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995.

[46] H. M. Callaway, D. Zyla, F. Larrous, G. D. de Melo, K. M. Hastie, R. D. Avalos, A. Agarwal, D. Corti, and H. Bourhy, "Structure of the rabies virus glycoprotein trimer bound to a prefusion-specific neutralizing antibody," *Science Advances*, vol. 8, no. 24, 2022.

[47] Centers for Disease Control and Prevention, "One health overview," 2025. Accessed: 2025-10-09.

[48] C. R. Fisher, D. G. Streicker, and M. J. Schnell, "The spread and evolution of rabies virus: conquering new frontiers," *Nature Reviews Microbiology*, vol. 16, no. 4, pp. 241–255, 2018.

[49] M. E. Griffiths *et al.*, "Inferring the disruption of rabies circulation in vampire bat populations using a betaherpesvirus-vectored transmissible vaccine," *Proceedings of the National Academy of Sciences*, vol. 120, no. 11, 2023.