Computational and Structural Biotechnology Journal 27 (2025) 692–704

Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Research Article

Machine and deep learning to predict viral fusion peptides

A.M. Sequeira [a,*], M. Rocha [a], Diana Lousa [b,*]

[a] Department of Informatics, School of Engineering, University of Minho, Braga, Portugal
[b] ITQB NOVA, Instituto de Tecnologia Química e Biológica António Xavier, Universidade Nova de Lisboa, Oeiras, Portugal

ARTICLE INFO

Keywords:
Viral fusion peptides
Protein classification
Machine learning
Protein language models

ABSTRACT

Viral fusion proteins, located on the surface of enveloped viruses like SARS-CoV-2, Influenza, and HIV, play a vital role in fusing the virus envelope with the host cell membrane. Fusion peptides, conserved segments within these proteins, are crucial for the fusion process and are potential targets for therapy. Experimental identification of fusion peptides is time-consuming and costly, which creates the need for bioinformatics tools that can predict the segment within the fusion protein sequence that corresponds to the FP. Although homology-based methods have been used towards this end, they fail to identify fusion peptides lacking overall sequence similarity to known counterparts. Therefore, alternative methods are needed to discover new putative fusion peptides, namely those based on machine learning. In this study, we explore various ML-based approaches to identify fusion peptides within a fusion protein sequence. We employ token classification methods and sliding window approaches coupled with machine and deep learning models. We evaluate different protein sequence representations, including one-hot encoding, physicochemical features, as well as representations from Natural Language Processing, such as word embeddings and transformers. Through the examination of over 50 combinations of models and features, we achieve promising results, particularly with models based on a state-of-the-art transformer for amino acid token classification. Furthermore, we utilize the best models to predict hypothetical fusion peptides for SARS-CoV-2, and critically analyse annotated peptides from existing research. Overall, our models effectively predict the location of fusion peptides, even in viruses for which limited experimental data is available.

1. Introduction

Fusion events between viral and host cell membranes are critical for enveloped viruses, including SARS-CoV-2, Dengue, Influenza, and HIV, as they enable these viruses to enter and infect cells. These viruses possess a lipid envelope on their surface, which harbors specialized membrane-fusion proteins known as viral fusion proteins (VFP). These glycoproteins play a crucial role in catalyzing the fusion process between viral and cellular membranes, thereby facilitating virus entry into the host cell and subsequent infection [1–3].

These proteins are classified based on key structural features observed in their pre- and post-fusion states. While these features may vary across distant virus lineages, they are highly conserved within the same virus family. Viral fusion processes generally follow a shared mechanism [1] that is common to all enveloped viruses.

According to the most widely accepted membrane fusion model, the fusion process begins with the binding of the VFP to the host receptor. An external stimulus then triggers a conformational change in the VFP, causing a nonpolar segment called the fusion peptide (FP) or fusion loop (FL) to project and insert into the host cell membrane. This extended form subsequently folds into a hairpin structure, bringing the fusion peptide in close proximity to the transmembrane domain of the VFP. This step allows the two membranes to come closer together, overcoming the dehydration force and enabling direct membrane apposition. Subsequently, the two membranes undergo an initial hemifusion, where the lipids of the outer leaflets merge. This is followed by the merging of the inner leaflets, resulting in the formation of an initial fusion pore. The pore then expands until full fusion occurs [1–4].

The current consensus suggests that VFPs can be classified into three main classes: I, II, and III, although some authors propose the existence of additional classes [1]. Each class is characterized by distinct structural differences in pre- and post-fusion states, orientation within the virus membrane, triggering mechanisms, and the location of their FP [2]. Fig. 1 provides a schematic overview of the fusion process and the role of VFPs from different classes in this process.

Class I fusion proteins are present in families such as retroviruses and

[*] Corresponding authors.
E-mail addresses: amsequeira@ceb.uminho.pt (A.M. Sequeira), dlousa@itqb.unl.pt (D. Lousa).

https://doi.org/10.1016/j.csbj.2025.02.011
Received 30 October 2024; Received in revised form 10 February 2025; Accepted 17 February 2025
Available online 18 February 2025

coronaviruses and have been extensively studied. The fusion peptide of this class is typically situated at the post-proteolytic N-terminus of the glycoprotein [1]. In the pre-fusion state, class I VFP form α-helix rich trimers of non-covalently associated heterodimers [1,4] (Fig. 1A – Class I). Fusion proteins belonging to class II are present in the Flaviviridae and Togaviridae families [1]. Contrary to class I, they insert hydrophobic FL, (also known as internal FPs), into the membrane. The FL is composed mostly of apolar and other conserved residues. The architecture of class II VFP contains three-domains and is primarily composed of β-strands with a tightly folded fusion loop in the central domain [1,5] (Fig. 1A – Class II). Class III fusion proteins are found in herpesviruses and rhabdoviruses and combine structural signatures found in classes I and II, i.e., they are trimers, with both α - helices and β-sheets, that dissociate into monomers. These proteins insert hydrophobic fusion loops into membranes and further oligomerize into post-fusion trimers [4,5] (Fig. 1A-Class III).

Fusion peptides, typically 20–30 residues long, exhibit moderate hydrophobicity, and are characterized by a high content of glycine (Gly) and alanine (Ala) residues. In most cases, they also contain aromatic residues, such as Tryptophan (Trp), and usually exhibit high flexibility, as the fusion process imparts a significant conformational rearrangement on the fusion proteins [1,2,4]. Within different virus families, FPs can display substantial sequence and structural diversity. However, conservation is observed within virus families, and mutations often result in a loss of function [4,6]. The locations of the FP within the viral fusion proteins vary depending on the VFP class (Fig. 1B). FPs from Class I are located at the post-proteolytic N-terminal tip of the fusion protein, while class II fusion proteins have an internal FL and class III fusion proteins usually have bipartite FLs [1,2].

The significance of VFP and their FP in membrane fusion underscores their potential as therapeutic targets. They play a crucial role in the development of protective vaccines and the design of novel fusion inhibitors for pathogenic enveloped viruses [1–3]. Moreover, understanding the fusion process can contribute to the development of strategies to efficiently deliver molecules and genes into target cells [1].

Laboratory-based efforts to define and characterize FP sequences are labor-intensive and costly, making bioinformatics tools indispensable in this endeavor. Given the observed conservation of these peptides within families, sequence alignment strategies, such as BLAST and Clustal, have been commonly used [2]. However, these tools are limited when it comes to identifying FP in VFP, lacking overall sequence homology with known templates or in viruses that have not been extensively studied. To address this challenge, methods not solely reliant on similarity are essential for the identification of new putative fusion peptides. In light of recent advancements in Artificial Intelligence (AI) applied to protein sequence analysis, Machine Learning (ML) methods may emerge as valuable tools for studying VFPs. Wu et al. developed a method that combined Hidden Markov Models (HMM) with similarity comparison to predict FP specifically from retroviruses, and this approach was further enhanced using a Support Vector Machine (SVM) model [7,8]. However, these approaches were limited in scope, focusing exclusively on retroviruses and not on the broader spectrum of enveloped viruses. Moreover, they relied heavily on similarity-based features, making them less effective for identifying FPs that lack significant identity or similarity to known sequences. In a significant development, Moreira et al. [2] presented the first ML models capable of predicting FP across all VFP classes and families. However, these initial models exhibited limited specificity, highlighting the need for the development of more effective computational methods for fusion peptide detection.

In this study, we employ various approaches to identify FP regions within VFPs. To identify the subsequence of the FP within the larger sequence VFP, one can apply sliding window approaches, where overlapping subsequences are generated and classified, or apply token classification methods, where each amino acid is classified as belonging or not to the fusion peptide. We utilize both approaches, in conjunction with shallow ML models, as well as deep learning (DL) techniques based

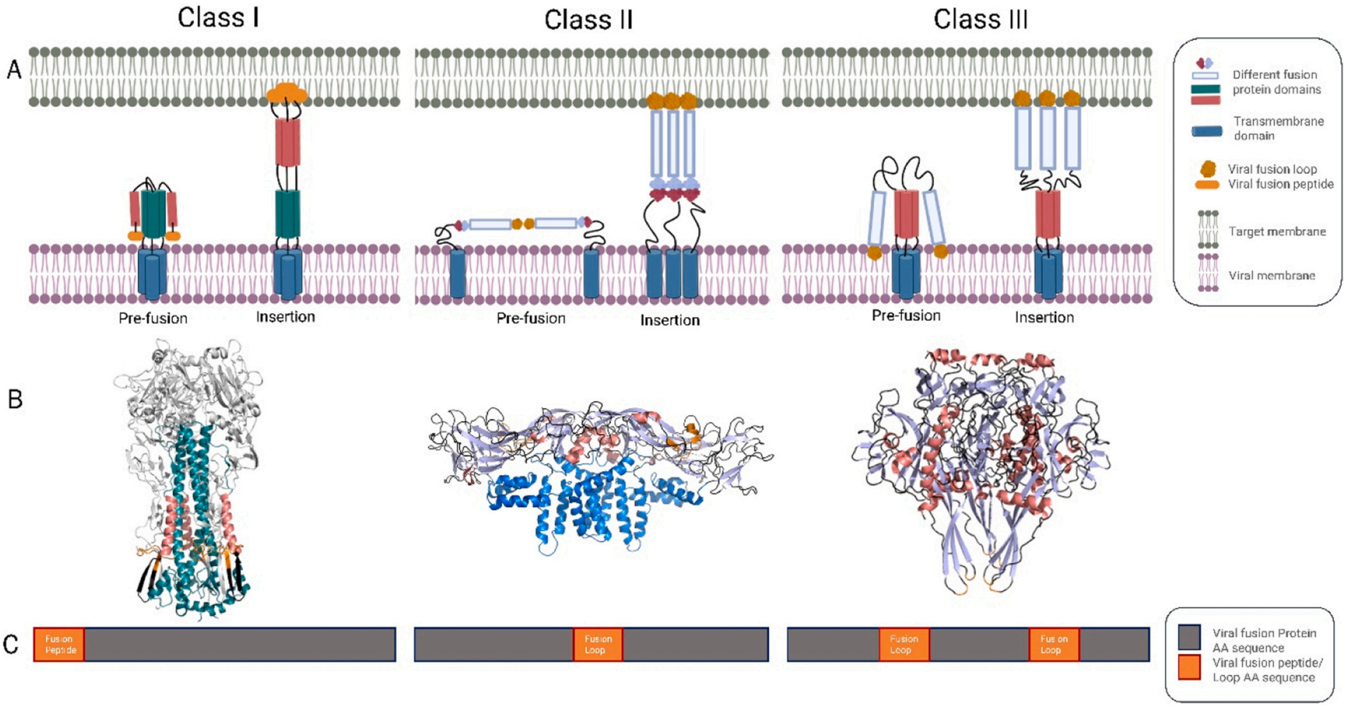

Fig. 1. – A) Major features of viral fusion protein structure across the three major classes of fusion proteins on native state and upon connection to the target membrane (scheme created with BioRender). B) Structures of representative proteins from Class I - Influenza A hemagglutinin (PDB code 2YPG), Class II - protein E from the Dengue virus (PDB code 3J27) and Class III - protein G from Vesicular Stomatitis Virus (PDB code 2J6J) in the pre-fusion state. The images were built with PyMOL and the protein secondary structures are shown using a cartoon representation with the colours defined according to the scheme shown in panel A. C) Depicts the general location of the fusion peptide and loops inside the Fusion protein sequence. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

on Recurrent Neural Networks (RNNs) and transformer architectures. To encode protein features, we explore different strategies including raw one-hot encoding, physicochemical features, and representations from the field of Natural Language Processing (NLP), such as word embeddings (WE) and transformer representations. Over 50 combinations of models and features are investigated, yielding promising results.

We leverage the best-performing models to predict hypothetical FPs from SARS-CoV-2, aligning with the most recent literature. Additionally, we critically analyze annotated peptides from the existing literature. Notably, the top-performing models are based on an ESM2b transformers for token classification. Overall, our models demonstrate the ability to effectively predict the location of FPs, even in the case of viruses that have not been extensively studied. The subsequent sections of this paper detail the generated datasets, the results obtained, and finally, an overview of the best models tested on the provided sequences. These results are discussed from both biological and computational perspectives.

2. Results

2.1. Viral fusion peptide dataset

The initial and crucial step involves constructing a reliable database containing both VFP and corresponding FP sequences. To accomplish this, FPs were sourced from ViralFP, an extensive database housing pertinent information on VFPs [2]. Subsequently, the dataset was curated to ensure its quality and accuracy. The final dataset comprises a total of 403 entries encompassing pairs of VFP and FP sequences, with 216 distinct FPs. The difference in numbers arises from the inclusion of different VFPs in the dataset, even if they possess identical FP sequences.

Fusion proteins typically consist of approximately 400–500 amino acids, while the length of most fusion peptides falls within the range of 20–25 amino acids. Regarding the distribution of viral fusion peptide classes, there are 261 peptides (142 excluding duplicates) belonging to Class I, 78 (48 excluding duplicates) to Class II, 27 (8 excluding duplicates) to Class III, and 37 with no attributed class (18 excluding duplicates). The most represented families within the dataset are Retroviridae, Flaviviridae, and Paramyxoviridae. In terms of fusion peptide similarity, out of the 403 sequences, 216 are found to be unique. To determine sequence similarity we utilized CDHIT [9]. Using a similarity cutoff of 90 % (with a word size of 5), 120 sequences clustered with less than 90 % similarity. Furthermore, there are 81 sequences with less than 80 % similarity and 40 sequences with 50 % similarity (with a word size of 3). The curated dataset is available in the study's repository, along with a notebook containing statistical information and graphical representations pertaining to the dataset.

To assess various models for identifying FP, we created a hold-out dataset consisting of 10 sequences comprising paired VFP and FP sequences. From the curated dataset, we determined sequence similarity at 80 %. Only sequences that are unique within their respective clusters were included, ensuring that no FP sequence with more than 80 % similarity appeared in both the training set and the hold-out dataset. We randomly selected 9 sequences and added the VFP sequence of SARS-CoV-2, which is still a matter of discussion with different segments having been proposed in the literature. A simplified overview of this dataset can be found in Table 1.

The limited size of the test dataset reflects the scarcity of experimentally validated FPs across all classes of enveloped viruses. While we curated the largest dataset available, future studies incorporating newly annotated FPs or experimentally validating novel predictions will be crucial to further enhance model robustness and generalizability.

With the training dataset and the holdout set, we took two approaches. The first one was using a sliding window approach, where subsequences are derived from VFP and classified as being FP or not in a regular ML protein classification problem. We used as features several strategies, such as one hot encoding, physicochemical features, word embeddings and transformers encodings and we coupled these features with traditional ML and DL based on RNN models. The second approach was based on the use of token classification, where each individual amino acid is classified as belonging or not to the FP sequence. Fig. 2 depicts the overall scheme of the project. The following sections of this article go through each of these approaches and analyse their results and take aways.

2.2. Classification of segments with a sliding window approach

One approach for predicting subsequences within a larger sequence is the sliding window. This method offers the advantage of transforming

Table 1
Overview of the hold-out test dataset, stating the UniprotID, VFP name, class, family and species of the virus, number of AAs in the VFP, the FP sequence and its location within the VFP sequence.

UniProtID	Name	Class	Family/Species	Nºaa VFP	Sequence of the FP	Sequence position of the VFep
P21443	Envelope glycoprotein	I	Retroviridae *Feline leukemia virus*	179	PISLTVALMLGGITVGGMARN	2 – 22
Q81487	Genome polyprotein	II	Flaviviridae *Hepatitis C virus*	193	LVAPPTLCSALYVEDAFGAVSL VGQAFTFRPR	75–107
Q68801	Genome polyprotein	II	Flaviviridae *Hepatitis C virus*	192	MVGAATLCSALYVGDLCG ALFLVGQGFSWRHR	74–106
P36334	Spike glycoprotein S	I	Coronaviridae *Human Betacoronavirus 1*	595	LAATSASLFPPWTAAAGVPFY	205–226
Q9QBZ4	Envelope glycoprotein gp160	I	Retroviridae *Human immunodeficiency virus 1*	346	AVGMGAVLFGFLGAAGSTMGA	1 – 21
Q9QSQ7	Envelope glycoprotein gp160	I	Retroviridae *Human immunodeficiency virus 1*	345	AAGLGALFLGFLGDSREHMGA	1 – 21
P07399	Pre-glycoprotein polyprotein GP complex	I	Arenaviridae *Lymphocytic choriomeningitis mammarenavirus*	233	GTFTWTLSDSSGVENPGGYCLTKWMILAAELKCFGNTAV	1–39
P35949	fusion glycoprotein F0	I	Pneumoviridae *Murine pneumonia virus*	436	FLGLILGLGAAVTAGVALAKT	1–21
Q8B0I1	G glycoprotein	III	Rhabdoviridae *Vesicular stomatitis virus*	511	[FRWYGPKY, CGYATVT]	86–94 130–137
P0DTC2	Spike glycoprotein	I	Coronaviridae *Human Coronavirus SARS-CoV−2*	588	-	-

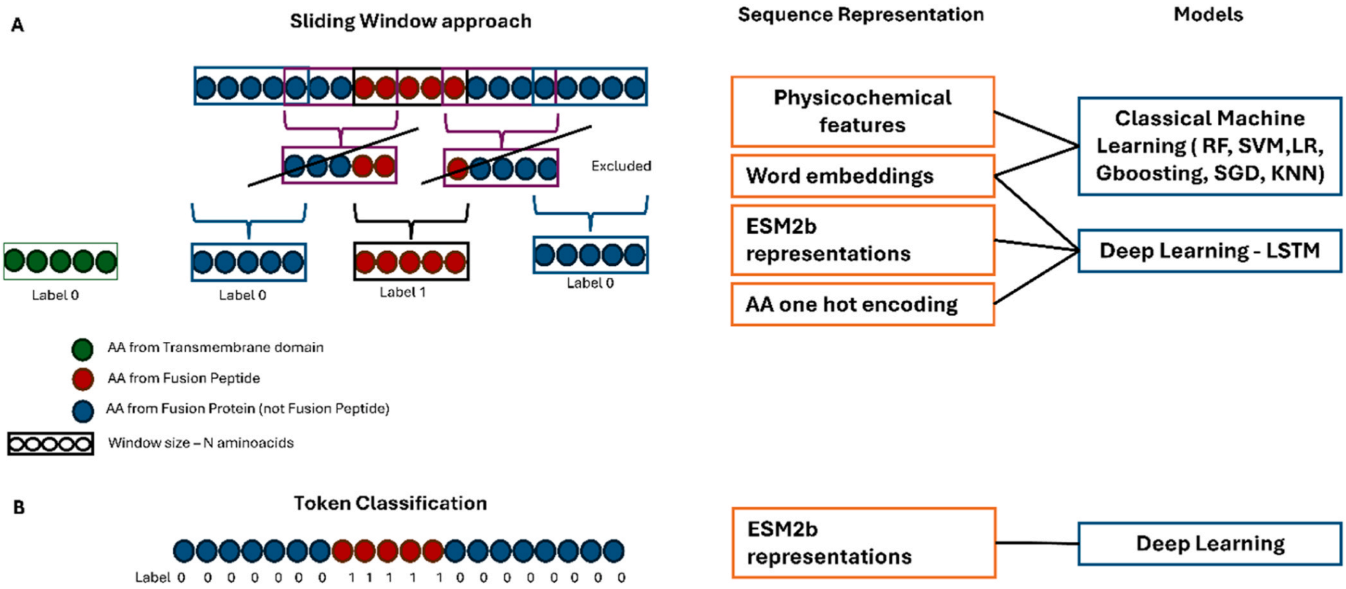

Fig. 2. Overview of the project approach. A) On the left: Schematic overview of the application of the sliding window approach. A sliding window of size **w** goes through the sequence. Subsequences that do not have an aa from the viral fusion peptide are considered negative. Subsequences that incorporate only aa’s from FP are considered positive and subsequences that contain a mixture of positive/negative are excluded. Subsequences annotated as transmembrane domains are also added, due to their similarity with FP; On the right: Feature extraction methods used and their coupling with ML and DL approaches. Physicochemical features are only employed with classical ML; word embeddings are tested using both classical ML and DL based on RNNs and transformers; ESM2b representations and one hot encoding are coupled with DL based on RNNs. B) On the left: overview on token classification, where each amino acid is classified as belonging or not to the FP; On the right: Feature representation and models used; for token classification, transformers based on ESM2b coupled with DL are used.

the problem into a conventional supervised learning problem. A window classifier, denoted as h_w, maps input windows of size w to individual output values, denoted as y. The window classifier h_w is trained by converting each sequential training example $(X_{i,}\ Y_i)$ into windows and applying a standard supervised learning algorithm. To classify a new sequence X, it is converted into windows, h_w is applied to predict each Y_t, and the predicted sequence Y is formed by concatenating the predicted Y_t values.

However, this method has some limitations. It fails to exploit correlations among nearby Y_t values. Only relationships that can be predicted from nearby X_t values are captured, while correlations among Y_t values that are independent of the X_t values are not captured [10].

In this approach, several important parameters need to be defined, including the window length w, the step size between each subsequence, and the strategy for building the classifier. This strategy encompasses other aspects such as the ratios of the generated datasets, the protein representation, and the classification model used.

In the specific context of predicting FP, the VFPs are sliced into subsequences using the sliding window approach. Segments that contain FP are labelled as 1, while subsequences that do not contain them are labelled as 0. Subsequences that include parts of FP and parts of non- FP are excluded from the dataset. Additionally, a border tolerance of 3 amino acids is included at each end of the FP due to the ill-defined nature of the fusion peptide borders. A window length of 21 is commonly used, as it captures the majority of FP in the dataset and their characteristic length. A step size of 1 is employed to ensure that all possible FP subsequences are included. A schematic overview of the sliding window approach can be seen in Fig. 2 - A.

The next step is to construct the datasets for applying supervised classification strategies. These datasets consist of the subsequences generated by applying the above-defined sliding window approach. Since this approach generates a large number of subsequences, including a subset of closely related ones, it is necessary to apply a filter based on similarity. In addition to FP subsequences, Transmembrane domains (TMDs) were also included in the dataset due to their physicochemical similarities to FP. The objective was to ensure that these datasets are appropriate to train robust models that can predict the FP location within a given VFP and are able to distinguish FPs from TMDs despite the common properties shared by these segments. To ensure the dataset’s quality, negative subsequences, and TMDs were filtered out using a 70 % similarity threshold. This filtering step aimed to avoid the inclusion of overly similar subsequences in the dataset. The 70 % similarity threshold was chosen empirically to balance sequence diversity and dataset size. This threshold ensures that sequences differ by at least six residues, minimizing redundancy while preserving a sufficient number of sequences for model training. As a result, the dataset consisted of 5819 negative subsequences, 784 TMDs, and 207 positive subsequences, which represents a highly unbalanced dataset towards negative subsequences and TMDs (complete dataset).

To explore the influence of the positive-to-negative ratio, two additional datasets were created. First, a dataset with a ratio of 1:2 was generated by randomly subsampling negative subsequences and TMDs from the original dataset (1 positive:1 negative subsequence: 1TMD) – Subsampled 1:2. Subsequently, a 1:1 ratio was created by randomly subsampling negative subsequences and TMDs from the second dataset (2 positive:1 negative subsequence: 1TMD) – Subsampled 1:1. By varying the positive-to-negative ratio, the impact of class imbalance on model performance could be examined. All models were then evaluated using the holdout dataset described earlier in the dataset section. Table 2

Table 2
Description of the datasets created for the sliding window approach. Ratios of positive sequences, negative subsequences, and TMDs and ratios of positive vs negative sequences (negative subsequences and TMDS) are depicted.

Dataset name	Ratio Pos: NegSubseq: NegTMD	Ratio Pos: Neg	Positive subseq	Negative subseq	Negative TMD
Complete	1:28:4	1:32	207	5819	784
Subsampled 1:2	1:1:1	1:2	207	231	207
Subsampled 1:1	2:1:1	1:1	207	103	103

describes these datasets, including the number of samples in each class and the positive-to-negative ratio.

To utilize peptide sequences in ML models, they need to be transformed into mathematical vectors. Protein representation plays a crucial role not only in developing effective predictive models, but also in gaining insights into the distinct characteristics of different proteins [11,12]. Various sequence representation schemes are commonly employed for this purpose, including physicochemical features [12–14], one-hot encoding [11,15,16], and embedding representations inspired by NLP [12,17–19].

Physicochemical features consider that the function or activity of a protein or peptide can be predicted from its physicochemical properties. These features can be computed solely based on the peptide sequence and encompass approaches, such as amino acid and pseudo amino acid composition. Physicochemical properties represent each amino acid using a set of physical properties (e.g., charge, hydrophobicity), and the entire protein is represented as a combination of these properties. Autocorrelation descriptors also capture physicochemical properties of amino acids at specific positions in higher dimensional protein space [12–14].

Another commonly used encoding method is sparse one-hot encoding, where each amino acid is represented as a one-hot vector of length 20, with all positions except one set to 0. However, this method typically requires peptides or proteins to be of the same length for compatibility [11,15,16,20]. One-hot encoding does not require prior knowledge of amino acids or protein sequences, as it establishes relationships between amino acids solely during the model training process.

In recent years, feature representations inspired by NLP have shown promising results when applied to biological problems [17–19]. One such representation is based on WE applied to proteins, introduced by Asgari et al. [17]. ProtVec, based on the Word2Vec algorithm, splits protein sequences into k-mers to capture the context of these word representations, containing crucial information about the protein sequence. Asgari et al. trained the algorithm using 3-mers from all proteins in UniprotKB and publicly shared the corresponding weights for all trigrams. More recently, Evolutionary Scale Modelling (ESM) models, which adapt the complex transformer architecture, have demonstrated outstanding results in protein analysis [19,21]. These models leverage evolutionary information to capture protein sequence patterns and relationships.

Finally, ML and DL can be applied to the subsequences represented as features calculated by one of the methods above. Classical ML models have demonstrated good results in protein classification problems and include well-established models such as Support Vector Machines (SVM), Random Forest (RF), Gradient Boosting (GB), Logistic Regression (LR), Stochastic Gradient Descent (SGD), Gaussian Naïve Bayes (GNB), and K-nearest neighbours (KNN). These models are suitable for the small dataset size and the nature of FP, which suggest that physicochemical and handcrafted features are effective representations for classical ML [12,14].

DL models also offer a powerful approach. RNNs are particularly suited for modelling sequence data, as they process input sequences one element at a time, utilizing a loop to iterate over sequence elements, while maintaining an internal state [22]. Long Short-Term Memory (LSTM) networks, a type of RNN, can retain past information even when dealing with long sequences and have demonstrated superior performance compared to other RNN architectures. In the context of biological sequences, a model can benefit from processing the sequence both forward and backward, capturing the context surrounding each item, rather than just the preceding items. Bidirectional LSTMs have proven to be effective in capturing non-local and long-range relationships within protein sequences and are widely used in bioinformatics [20,23,24]. However, the limited size of the dataset poses a constraint on the use of DL models, and this limitation also impacts the use of one-hot encoding.

Considering the different protein representations and ML models, not all combinations were tested. Handcrafted physicochemical features were employed with traditional ML models. WE and the latest ESM model, ESM2b, were used with both classical ML models and LSTM networks. One-hot encoding was exclusively used with LSTM networks (Fig −2 A). The results of these strategies are further described in the following subsections. Given the significant volume of models generated, we focus our attention on combinations that surpass a threshold of 5 accurately corrected test sequences (holdout dataset) and models that have the highest 10-fold Cross-validation (CV) scores (accuracy, precision, recall, MCC, F1, and ROC-AUC above 0.8). These noteworthy combinations are then examined in greater detail. Comprehensive information, including scores and predicted sequences from all models, can be accessed on the GitHub page.

The 10-fold CV scores for these models are depicted in Table 3. An overview of the test performance of the best models obtained by CV is shown in Fig. 3 and described in Supplementary Table 1. The performance metrics displayed describe the number of test entries for which a prediction: is generated; are correctly predicted and correspond to a unique prediction; are correctly predicted but do not correspond to a unique prediction; are incorrectly predicted.

2.2.1. *Classification of subsequences with physicochemical features and machine learning*

FP exhibit distinctive and well-defined physicochemical characteristics. Consequently, the initial approach involved calculating the physicochemical features of subsequences and utilizing them as input for various traditional ML models. Feature selection was performed by considering the top 50 % percentile of features, determined using the mutual information classification function. Both the selected feature set and the complete feature set were evaluated.

To optimize the ML models, hyperparameter tuning was conducted using grid search in combination with a 10-grouped k-fold strategy. The grouping was based on an 80 % similarity threshold among the subsequences, ensuring that sequences sharing more than 80 % similarity were consistently placed in the same fold.

Multiple ML algorithms, including SVM, RF, GB, LR, KNN, GNB, and SGD, were tested. The three distinct datasets described above, each with different positive-negative ratios, were employed for evaluation purposes. Table 3 displays the 10-fold CV scores, as well as the accuracy, precision, recall, MCC, F1, and ROC-AUC of the models for which all these metrics are above 0.80. In general, the models achieved very good performances. Models trained on the complete dataset tend to have low recall (0.55–0.70), and, thus, are not shown on the table. This is most likely due to the very low positive ratio, which results in a poor ability to recall positive instances. The datasets Subsampled with a ratio of 1:2 and 1:1, on the other hand, achieve good results in all metrics, with the majority of the models achieving both precision and recall above 0.80.

The models trained with CV were then applied to predict the sequences of the test subset (9 entries with annotations and the SARS-CoV-2 sequence), which had not been included in the training set. In general, models trained on the complete dataset tend to be unable to generate any predictions for the majority of the test sequences. However, these models consistently make correct predictions for test sequences Q9QBZ4 and P35949. This aligns with the CV scores that already showed low recalls.

On the other hand, models trained on the dataset Subsampled 1:1 ratio tend to generate predictions for the majority of cases presented in the test set. SVM-based models, in this case, make multiple predictions per sequence. While they do predict the correct FP sequences for some test cases, they produce incorrect predictions for other test cases, rendering them unsuitable for the problem at hand. GB models with mutual feature selection, as well as RF models with and without feature selection, achieve 5 or more correctly predicted test sequences and are discussed in more detail below.

Models trained with the dataset Subsampled 1:2 ratio predict more test sequences compared to the complete dataset, but fewer than the Subsampled 1:1 dataset. Additionally, these models do not tend to

Table 3
Mean and standard deviation 10-fold CV scores for the models with accuracy, precision, recall, MCC, F1, and ROC-AUC above 0.80 and best performance test set models ML combinations that surpass a threshold of 5 accurately corrected test sequences (holdout dataset) for all the combinations using a sliding window strategy.

Sequence representation/ feature selection		Dataset	model	CV scores (10 stratified group kfold) mean and std					
				accuracy	precision	recall	MCC	F1	ROC AUC
Physicochemical	Mutual	Subsampled 1:2	RF	0.93 ± 0.04	0.95 ± 0.04	0.81 ± 0.11	0.83 ± 0.08	0.87 ± 0.06	0.89 ± 0.06
	Mutual	Subsampled 1:2	SVM	0.94 ± 0.04	0.96 ± 0.05	0.85 ± 0.11	0.86 ± 0.09	0.89 ± 0.07	0.92 ± 0.06
	None	Subsampled 1:2	GB	0.94 ± 0.03	0.92 ± 0.03	0.87 ± 0.11	0.86 ± 0.08	0.89 ± 0.07	0.92 ± 0.06
	None	Subsampled 1:2	Linear SVM	0.93 ± 0.02	0.88 ± 0.06	0.89 ± 0.07	0.84 ± 0.06	0.88 ± 0.05	0.92 ± 0.03
	None	Subsampled 1:2	LR	0.94 ± 0.03	0.99 ± 0.02	0.81 ± 0.11	0.86 ± 0.07	0.89 ± 0.07	0.90 ± 0.05
	None	Subsampled 1:2	SVM	0.95 ± 0.03	0.98 ± 0.04	0.83 ± 0.10	0.86 ± 0.07	0.89 ± 0.06	0.91 ± 0.05
	Mutual	Subsampled 1:1	GB	0.92 ± 0.05	0.95 ± 0.05	0.89 ± 0.09	0.85 ± 0.10	0.91 ± 0.05	0.92 ± 0.05
	Mutual	Subsampled 1:1	RF	0.91 ± 0.06	0.94 ± 0.06	0.85 ± 0.11	0.82 ± 0.12	0.89 ± 0.06	0.90 ± 0.07
	Mutual	Subsampled 1:1	SVM	0.93 ± 0.03	0.96 ± 0.05	0.88 ± 0.10	0.86 ± 0.07	0.92 ± 0.06	0.92 ± 0.04
	None	Subsampled 1:1	RF	0.91 ± 0.06	0.95 ± 0.07	0.84 ± 0.11	0.81 ± 0.12	0.89 ± 0.09	0.90 ± 0.06
	None	Subsampled 1:1	SVM	0.92 ± 0.04	0.97 ± 0.05	0.85 ± 0.10	0.84 ± 0.09	0.90 ± 0.07	0.91 ± 0.05
Word embedding	method3	Complete	SVM	0.99 ± 0.00	0.95 ± 0.04	0.64 ± 0.17	0.77 ± 0.11	0.75 ± 0.13	0.82 ± 0.08
	method3	Complete	KNN	0.99 ± 0.01	0.88 ± 0.12	0.81 + ± 0.14	0.84 ± 0.12	0.84 ± 0.12	0.90 ± 0.07
	method3	Subsampled 1:2	RF	0.88 ± 0.06	0.91 ± 0.08	0.66 ± 0.18	0.70 ± 0.15	0.75 ± 0.14	0.82 ± 0.09
	method3	Subsampled 1:2	SVM	0.92 ± 0.03	0.90 ± 0.09	0.82 ± 0.12	0.81 ± 0.07	0.85 ± 0.07	0.89 ± 0.05
	method3	Subsampled 1:1	SVM	0.92 ± 0.06	0.93 ± 0.06	0.86 ± 0.13	0.82 ± 0.13	0.89 ± 0.09	0.90 ± 0.07
	method1	Subsampled 1:1	RF	0.83 ± 0.12	0.96 ± 0.10	0.64 ± 0.24	0.67 ± 0.22	0.75 ± 0.21	0.81 ± 0.12
	method1	Complete	LSTM2	0.99 ± 0.00	0.95 ± 0.06	0.64 ± 0.18	0.77 ± 0.10	0.75 ± 0.12	0.82 ± 0.09
	method1	Complete	LSTM5	0.99 ± 0.00	0.83 ± 0.13	0.68 ± 0.22	0.74 ± 0.16	0.73 ± 0.17	0.84 ± 0.11
ESM2b	T68M	Subsampled 1:2	RF	0.91 ± 0.05	0.93 ± 0.07	0.75 ± 0.14	0.77 ± 0.12	0.82 ± 0.10	0.86 ± 0.07
	T68M	Subsampled 1:1	SVM	0.89 ± 0.06	0.91 ± 0.07	0.83 ± 0.12	0.76 ± 0.13	0.86 ± 0.10	0.88 ± 0.07

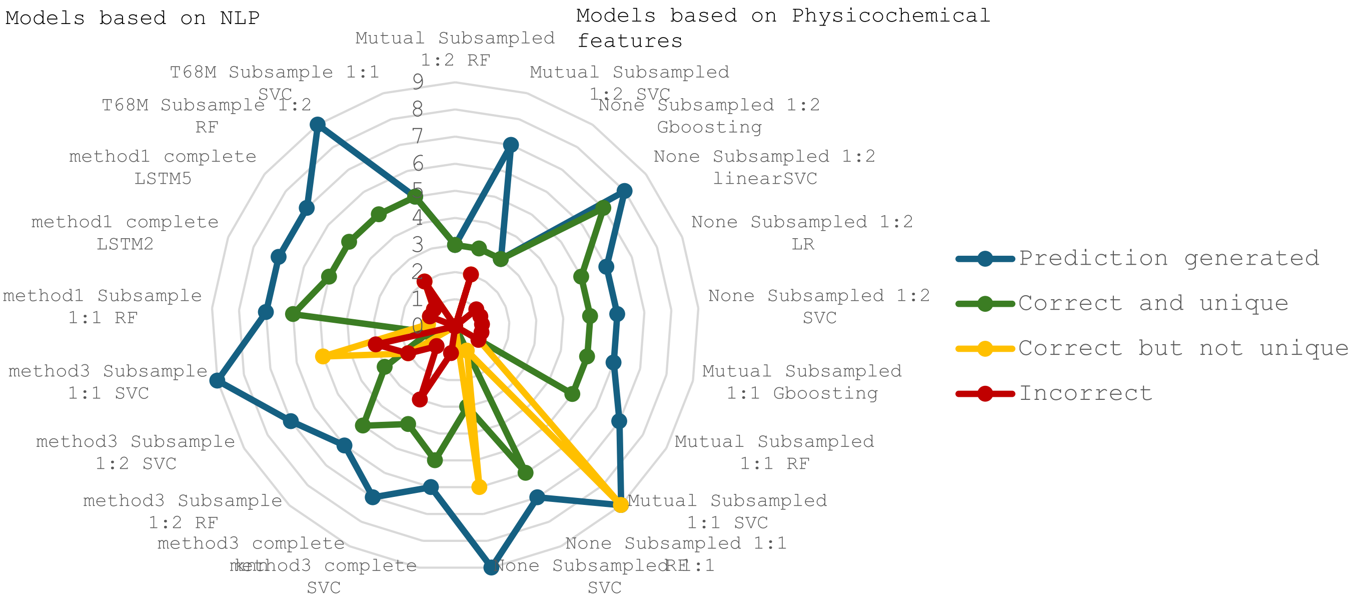

Fig. 3. Analysis of the predictions in the test set of the best models described in Table 3 applied to test sequences (holdout dataset). For the set of 9 sequences that were used for testing, the metrics presented for each model represent the number of sequences for which: a prediction was generated (column name "Prediction generated"), only one prediction was generated and was correct (column name "Correct and unique"), several predictions were generated including the correct one (column name "Correct but not unique"), all the generated predictions were incorrect (column name "Incorrect"). The information is represented as table in Supplementary Table 1.

generate more than one predicted subsequence per protein test sequence. The SVM and LR models with no feature selection demonstrate the best results, correctly predicting 5 test sequences. The best model was obtained with a dataset with a ratio 1:1, with no feature selection, and an RF model. This model correctly predicts 6 sequences (P21443, Q68801, P36334, Q9QBZ4, Q9QSQ7, and P07399), one sequence (P35949) contains two subsequences predicted, one correct and one incorrect. This model did not predict any FP sequence for two sequences (Q81487, Q8B0I1). The SARS-CoV-2 peptide (P0DTC2) has 3 predictions, however, with a more conservative score (above 0.6 probability) just one subsequence is predicted.

The model identifies several key features that are highly influential in making accurate predictions. The top features that contribute significantly to the model's performance include hydrophobicity, and the amino acid composition of Glycine (Gly), Alanine (Ala), Serine (Ser), Threonine (Thr), Proline (Pro), Histidine (His), and Thyrosine (Tyr). Additionally, the composition of Gly, polarizability associated with the composition of Gly, Ala, Ser, Aspartic acid (Asp), and Threonine (Thr), molecular weight, and the Gly-Ala dipeptide composition are also relevant features. These findings underscore the model's ability to capture meaningful biological representations to some extent, as the identified features align with some of the known important differentiating factors for FP.

An additional discussion of the best models is made in the Overview section.

2.2.2. Classification of subsequences with one-hot encoding and deep learning

Subsequently, we explored the approach of encoding subsequences as one-hot vectors and feeding them into LSTM-based networks. We conducted extensive tests, varying the architecture by experimenting with different numbers of layers, units, and bidirectional configurations, and incorporating attention mechanisms. Unfortunately, both the training and test scores obtained from this approach were consistently low, indicating that it was not fruitful to pursue further. These models tend to have high accuracies (above 0.8) and low recalls and MCCs (0.6 – 0.7); on the test set, they also tend to predict a high number of subsequence as being FP. A potential explanation for these results lies in the extremely limited size of the dataset. It appears that the dataset's small scale does not provide sufficient information for DL models to learn meaningful representations of the amino acid sequences when using one-hot encoding alone. Consequently, the DL models struggle to generate substantial and reliable predictions due to the lack of additional information in the encoding process.

2.2.3. Classification of subsequences with NLP features using machine and deep learning

Considering the successful application of NLP to protein classification, we aimed to evaluate the effectiveness of WE and ESM, pretrained language models for proteins in conjunction with classical ML and DL methods.

Starting with WE and given the limited size of our dataset, we employed the pre-trained protein WE model Protvec [17]. This model is trained on amino acid trigrams and yields a 100-dimensional vector representation. There are various approaches to applying WE, and for our study, we adopted the most used techniques from the literature. The first approach involves substituting each trigram in the sequence with its corresponding 100-dimensional vector, resulting in a final vector of dimension 100 * (sequence length - 2). This approach is suitable for both ML and DL models as it preserves token sequence and positional information. The second approach entails counting the occurrences of all trigrams, multiplying each trigram by its occurrence count, and summing them to generate a final protein vector of dimension 100. This approach is more suitable for ML models [25].

Initially, WE using the second approach was applied to obtain a 100-dimensional feature representation for each sequence. Subsequently, these features were utilized as input for ML models following the same strategy described earlier.

Overall, the performance of these models was inferior to that of models utilizing physicochemical features. Considering the CV scores, these models consistently obtain lower metrics when compared to models based on physicochemical features, especially considering recall, MCC, and the F1 score. Table 3 depicts the scores of models with scores above 0.80 in all metrics, scores obtained with complete dataset and SVM and dataset subsample 1:2 with RF were included due to their better performance in the test set.

Considering the test set, the majority of these models tended to predict multiple subsequences per protein. Similarly to the physicochemical-based models, the ones trained on the dataset Subsampled 1:1 predict multiple subsequences, and, in contrast, the ones trained on the complete dataset predicted fewer subsequences overall. The bad performance in the test set accompanies the tendency of the weaker results in the CV models. Considering the test set, two models should be highlighted, which demonstrated 5 correctly predicted test sequences. The first model was trained on the complete dataset with SVM. The second model was trained on the dataset subsampled 1:2 ratio and employed RF. These models are probably not consistent, as the standard deviation is above 0.1, which could explain why metrics relatively lower in CV scores could produce better results in the test set.

Furthermore, we decided to apply the first WE method, which involves representing each amino acid trigram with a 100-dimensional vector, to shallow ML models. The CV scores were low overall, with just one model (dataset Subsampled 1:1 and SVC) achieving more than 0.80 in all metrics. In general, the models show standard deviations in the metrics above 0.1, which may represent some instability. The rest of the models, although with high precisions have recalls ranging from 0.5 to 0.7. Considering the test set, as anticipated, these models exhibited overall lower performance and predicted multiple subsequences for proteins. Among these models, the best-performing one was based on the dataset Subsampled 1:1, utilizing RF, with 6 correct predictions (Q81487, Q68801, Q9QBZ4, Q9QSQ7, P07399, and P35949), and one correct with a second wrong prediction (P36334). This model did not make any predictions for 2 sequences (P21443, Q8B0I1). Additionally, this model made three subsequence predictions for the SARS-CoV-2. 10-fold CV scores for these models are depicted in Table 3. The model achieving best CV scores, identified a significant part of all of the test sequences as corresponding to the FP and was discarded.

Finally, we explored the application of WE using the first method, which retains positional information, in combination with RNNs. Specifically, we tested various configurations utilizing LSTM models with a varying number of layers, units, bidirectionality, and attention mechanisms (refer to the methods section for a detailed explanation). These models produced good results with precisions ranging from 0.83 to 0.95 and recall significantly lower from 0.60 to 0.68. Similarly to the previous WE-based models, the standard deviation is high showing some fluctuations in the models. Considering the test set, two models should be highlighted, both using the Complete dataset that achieved 5 correctly predicted sequences from the test set: LSTM network with 2 layers of LSTM (64 and 32 units), followed by a dense layer with 32 units and a final classification layer and a network composed of 2 bidirectional LSTM (128 and 64 units), followed by an attention mechanism and a dense layer with 64 units and a final classification layer. These models, with slight variations in the positions of the start and end of the predicted peptides, achieved correct predictions for 5 sequences (P21443, Q68801, Q9QBZ4,Q9QSQ7, and P07399), did not predict any subsequence on 2 sequences (Q81487 and Q8B0I1), and wrongly predicted one sequence (P36334). Both propose 1 FP for SARS-CoV-2.

Deriving from the exceptional results reported in the literature, we also explored the utilization of the Transformer protein language models from the Meta Fundamental AI Research Protein Team, namely the state-of-the-art ESM-2, which is shown to outperform all tested single-sequence protein language models across a range of prediction tasks [19,21] for feature representation. We experimented with feeding the sequence representations from two different models: the smaller model with 6 layers and 8 million parameters (T68 M) and the larger model with 33 layers and 650 million parameters (T33650M). These representations were fed into ML (RF, SVM) and DL models based on LSTM. Using ML models achieved better scores both in CV scores and the test set, this may be explained by the already complex features and with a very small dataset. Considering the ML models, the precision models ranged from 0.85 to 1 with recall from 0.66 to 0.83.

Considering the hold-out set, these models tended to predict multiple subsequences per sequence and did not perform well. Just two models achieved at least 5 correct predictions. The dataset with 1:1 ratio, representations from T68M and SVC, and the dataset with 1:2 ratio, representations from T68M and RF.

Next, we attempted to employ LSTM-based models using the representations from T68M and T33650M. To accommodate for the recurrent nature of these models, we utilized a representation size of 23 * 1280, rather than taking the mean representation per sequence. We tested both sequence and contact representations. However, these models did not surpass the performance of previous strategies and tended to predict multiple subsequences per protein. Consequently, we did not explore this approach further.

Furthermore, we conducted finetuning experiments on both T68M and T33650M for 3 epochs using all three datasets. Unfortunately, none of these models achieved a minimum of 5 correctly predicted sequences. Given that the models were already displaying signs of overfitting, extending the fine-tuning duration was not considered. The poor performance observed when utilizing these complex representations in conjunction with deep learning can be attributed to the limited size of the dataset and the high similarity among many positive sequences. These factors further reduce the diversity of the dataset, making it challenging for these models to learn effectively.

2.3. Classification of fusion peptides with token classification

Deriving from the success of ESM we applied it to token classification. Instead of classifying an entire sequence, we categorized each amino acid token as belonging or not to a FP. To perform this task, and similarly to the classification of segments, we used the ESM2 T6 8 M UR50D model and the ESM2 T12 35 M [19,21] as previously.

We fine-tuned the ESM2 models with the fusion peptide dataset for different epochs, ranging from 3 to 30 for ESM2 T6 8 M and 3–20 for ESM2 T12 35 M UR50D. Subsequently, we evaluated the models on the holdout dataset with 10 test sequences. Notably, these models outperformed the sliding window approach models by a significant margin. It is worth noting that increasing the complexity of the models did not seem to improve the classification performance and considering the computational requirements of the remaining ESM models, we focused on testing these two variants.

In token classification tasks, masked accuracy is a common evaluation metric. It measures how accurately the model predicts the correct class labels for the masked tokens in the sequence. It focuses on the performance of the model when dealing with partially observed or hidden information. All the models achieve similar scores in training, with epochs and model size not having significant differences. However, it is worth noting that model T12 considerably takes more time to run. In the same way, the improvement of results is not significative with increasing the number of epochs of training. Supplementary Table 2 shows the masked accuracy scores and loss of the training process.

In the test set, the models achieved outstanding performance, with the majority getting 7 out of 9 right predictions. Table 4 further explores the Jaccard index scores individually for each test sequence).

Considering the holdout test, all the models performed very similarly with some variations on the beginning and ending of the fusion peptides. However, as discussed above, the borders of these peptides are not fully established, and one should take that into consideration when looking at the results. In terms of model, using the smallest ESM2b model T6 8 M yielded overall better results with significantly less time running and computer resources needed. The model with the best trade-off between scores and time of computing. was the ESM2b T68M fine-tuned for 20 epochs. To represent the predictions of this model on the test sequences, Fig. 4 depicts the 10 VFP with the annotated FP in Bold and underscored; each amino acid residue in the sequence is coloured accordingly to the score of the model as belonging to the FP (red high probability). In the figure, it is possible to note how this model, with exception of sequences Q8B0I1 and P36334, can identify FP inside the VFP and without predicting other sequences.

The majority of the models got right the sequences P21443, Q81487, Q68801, Q9QBZ4, Q9QSQ7 and P35949 and a part of the peptide of sequence P07399. The peptide Q8B0I1, is not only not correctly predicted in any of the models, but no model makes a single prediction on this peptide. Regarding the sequence P36334 from Spike glycoprotein S of the Human Coronavirus, the majority of the token classification transformers (except the T68M finetuned for 3 epochs) predicted this sequence (with some minor variations) to be ’SKASSRSAIEDLLFDK’, position 140–156, instead of the annotated ’LAATSASLFPPWTAAAGVPFY’, position 206–226. Interestingly, we note that both homologous regions in SARS-CoV-2 have been postulated as corresponding to the fusion peptide.

The last sequence to be tested was the Spike glycoprotein from SARS-CoV-2. Scores will be discussed in the next section, given that there is no consensus regarding the region that corresponds to the FP in this virus. All the transformer models predicted a FP for this fusion protein. Although with some variation on the start and end amino acid residues, the sequence that was most often predicted is ’QIYKTPPIKDFGGFNFSQI’. A more detailed discussion of the predictions of all models can be seen in the next section.

2.4. A biological perspective on the predictions on test sequences

Finally, we conclude and examine patterns from the predictions of the test sequences made by the models that achieved more than 5 correct predictions. In total, we analysed 21 models for both sliding window approach (5 based on physicochemical features, 5 based on WE and 2 on ESM2b), and token classification approach (9 transformers models). Through this analysis, we aim to identify commonalities and patterns in the successful predictions across these diverse approaches.

Sequence Q9QBZ4, a glycoprotein class I from Retroviridae is correctly predicted in all 22 models. However, models based on a sliding window approach with WE tend to extend the end of this peptide. Sequence Q9QSQ7, also a glycoprotein class I from Retroviridae, is predicted correctly on 21 of the models with one model (sliding window with dataset ratio 1:1, f. selection andGB) not predicting any subsequence. Sequence P21443, an Envelope glycoprotein class I from a virus from the same family (Retroviridae) was correctly identified by 20 models and led to no predictions by 2 models (sliding window approach with WE for both datasets all, WE method3 andSVM and dataset half, WE method1 and RF).

Sequence P35949, a fusion glycoprotein from class I and family Pneumoviridae was predicted correctly in 18 of the models. 4 of the models (sliding window, physicochemical dataset half with and without feature selection and RF and WE method 1, dataset all with LSTM) predict 2 subsequences for this protein, the correct annotated peptide, and a second fusion peptide, with differences in the terminals of the

Table 4
Jaccard index for the 9 test sequences obtained with token classification with ESM2b T68 M and T12 models finetuned with 3,5,10,15 and 20 epochs. The model with the best trade-off between scores and time of computing. was the ESM2b T68 M fine-tuned for 20 epochs (bold).

		Jaccard index Test Sequences								
Model	Epoch	P21443	Q81487	Q68801	P36334	Q9QBZ4	Q9QSQ7	P07399	P35949	Q8B0I1
T68 M	3	0.86	0.66	0.91	0.05	0.95	0.76	0.85	0.95	0.00
T68M	5	0.86	0.91	0.94	0.00	0.95	0.76	0.64	0.95	0.00
T68M	10	0.86	0.91	0.94	0.00	0.95	0.76	0.77	0.95	0.00
T68M	15	0.90	0.75	0.94	0.00	0.95	0.76	0.56	0.95	0.00
T68M	**20**	**0.90**	**0.72**	**0.94**	**0.00**	**0.95**	**0.86**	**0.69**	**0.95**	**0.00**
T68M	30	0.90	0.63	0.91	0.00	0.95	0.81	0.74	0.95	0.00
T12	3	0.77	0.78	0.88	0.24	0.95	0.76	0.62	0.95	0.00
T12	5	0.77	0.75	0.94	0.00	0.95	0.81	0.49	1.00	0.00
T12	10	0.91	0.84	0.94	0.00	0.95	0.95	0.46	0.95	0.00
T12	15	0.90	0.13	0.94	0.00	0.95	0.67	0.49	0.95	0.00
T12	20	0.90	0.00	0.94	0.00	0.95	0.81	0.49	0.95	0.00

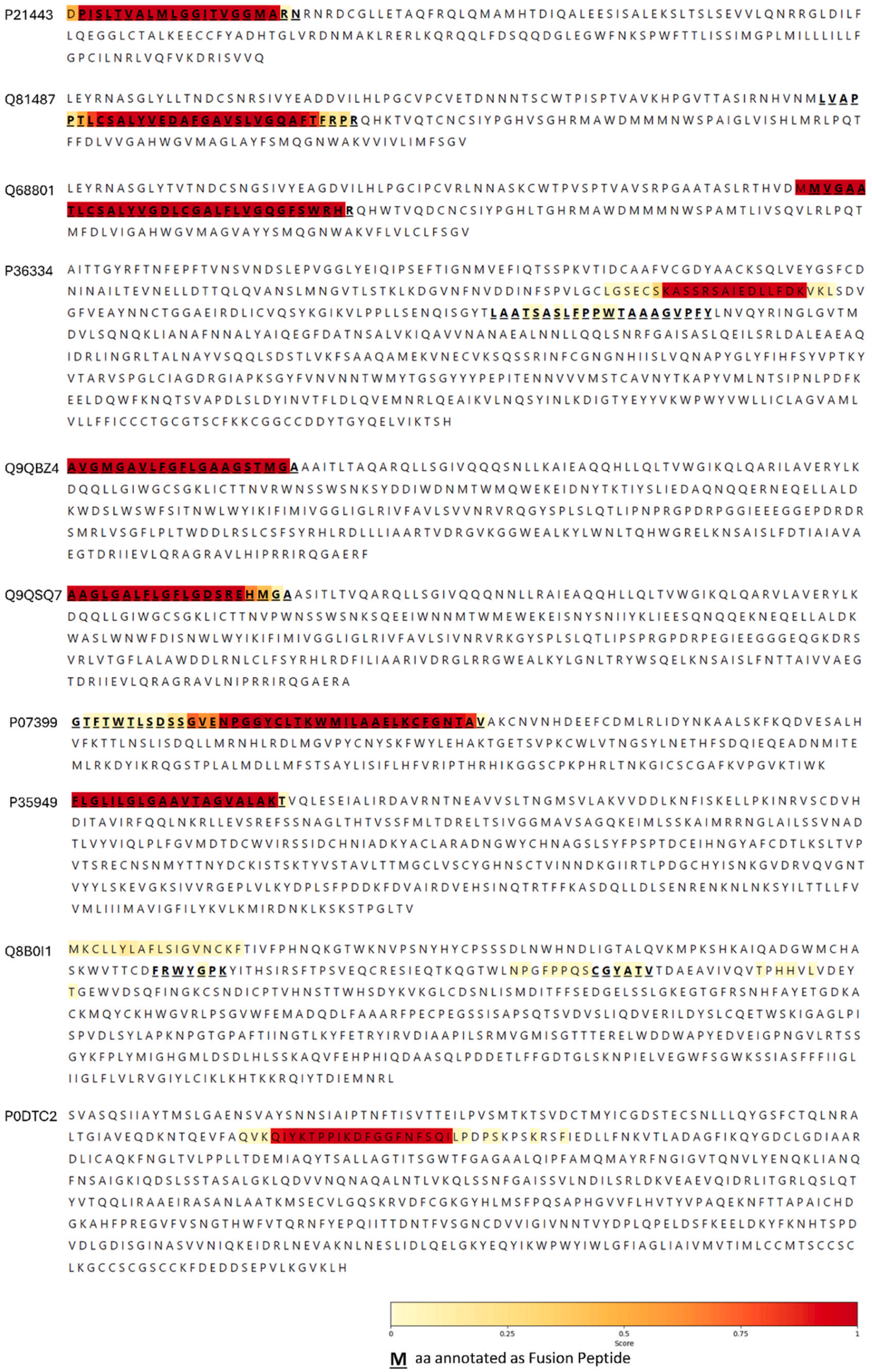
P21443
DPISLTVALMLGGITVGGMARNRNRDCGLLETAQFRQLQMAMHTDIQALEESISALEKSLTSLSEVVLQNRRGLDILF
LQEGGLCTALKEECCFYADHTGLVRDNMAKLRERLKQRQQLFDSQQDGLEGWFNKSPWFTTLISSIMGPLMILLLILLF
GPCILNRLVQFVKDRISVVQ
Q81487
LEYRNASGLYLLTNDCSNRSIVYEADDVILHLPGCVPCVETDNNNTSCWTPISPTVAVKHPGVTTASIRNHVNMLVAP
PTLCSALYVEDAFGAVSLVGQAFTFRPRQHKTVQTCNCSIYPGHVSGHRMAWDMMMNWSPAIGLVISHLMRLPQT
FFDLVVGAHWGVMAGLAYFSMQGNWAKVVIVLIMFSGV
Q68801
LEYRNASGLYTVTNDCSNGSIVYEAGDVILHLPGCIPCVRLNNASKCWTPVSPTVAVSRPGAATASLRTHVDMMVGAA
TLCSALYVGDLCGALFLVGQGFSWRHRQHWTVQDCNCSIYPGHLTGHRMAWDMMMNWSPAMTLIVSQVLRLPQT
MFDLVIGAHWGVMAGVAYYSMQGNWAKVFLVLCLFSGV
P36334
AITTGYRFTNFEPFTVNSVNDSLEPVGGLYEIQIPSEFTIGNMVEFIQTSSPKVTIDCAAFVCGDYAACKSQLVEYGSFCD
NINAILTEVNELLDTTQLQVANSLMNGVTLSTKLKDGVNFNVDDINFSPVLGCLGSECSKASSRSAIEDLLFDKVKLSDV
GFVEAYNNCTGGAEIRDLICVQSYKGIKVLPPLLSENQISGYTLAATSASLFPPWTAAAGVPFYLNVQYRINGLGVTM
DVLSQNQKLIANAFNNALYAIQEGFDATNSALVKIQAVVNANAEALNNLLQQLSNRFGAISASLQEILSRLDALEAEAQ
IDRLINGRLTALNAYVSQQLSDSTLVKFSAAQAMEKVNECVKSQSSRINFCGNGNHIISLVQNAPYGLYFIHFSYVPTKY
VTARVSPGLCIAGDRGIAPKSGYFVNVNNTWMYTGSGYYYPEPITENNVVVMSTCAVNYTKAPYVMLNTSIPNLPDFK
EELDQWFKNQTSVAPDLSLDYINVTFLDLQVEMNRLQEAIKVLNQSYINLKDIGTYEYYVKWPWYVWLLICLAGVAML
VLLFFICCCTGCGTSCFKKCGGCCDDYTGYQELVIKTSH
Q9QBZ4
AVGMGAVLFGFLGAAGSTMGAAAITLTAQARQLLSGIVQQQSNLLKAIEAQQHLLQLTVWGIKQLQARILAVERYLK
DQQLLGIWGCSGKLICTTNVRWNSSWSNKSYDDIWDNMTWMQWEKEIDNYTKTIYSLIEDAQNQQERNEQELLALD
KWDSLWSWFSITNWLWYIKIFIMIVGGLIGLRIVFAVLSVVNRVRQGYSPLSLQTLIPNPRGPDRPGGIEEEGGEPDRDR
SMRLVSGFLPLTWDDLRSLCSFSYRHLRDLLLIAARTVDRGVKGGWEALKYLWNLTQHWGRELKNSAISLFDTIAIAVA
EGTDRIIEVLQRAGRAVLHIPRRIRQGAERF
Q9QSQ7
AAGLGALFLGFLGDSREHMGAASITLTVQARQLLSGIVQQQNNLLRAIEAQQHLLQLTVWGIKQLQARVLAVERYLK
DQQLLGIWGCSGKLICTTNVPWNSSWSNKSQEEIWNNMTWMEWEKEISNYSNIIYKLIEESQNQQEKNEQELLALDK
WASLWNWFDISNWLWYIKIFIMIVGGLIGLRIVFAVLSIVNRVRKGYSPLSLQTLIPSPRGPDRPEGIEEGGGEQGKDRS
VRLVTGFLALAWDDLRNLCLFSYRHLRDFILIAARIVDRGLRRGWEALKYLGNLTRYWSQELKNSAISLFNTTAIVVAEG
TDRIIEVLQRAGRAVLNIPRRIRQGAERA
P07399
GTFTWTLSDSSGVENPGGYCLTKWMILAAELKCFGNTAVAKCNVNHDEEFCDMLRLIDYNKAALSKFKQDVESALH
VFKTTLNSLISDQLLMRNHLRDLMGVPYCNYSKFWYLEHAKTGETSVPKCWLVTNGSYLNETHFSDQIEQEADNMITE
MLRKDYIKRQGSTPLALMDLLMFSTSAYLISIFLHFVRIPTHRHIKGGSCPKPHRLTNKGICSCGAFKVPGVKTIWK
P35949
FLGLILGLGAAVTAGVALAKTVQLESEIALIRDAVRNTNEAVVSLTNGMSVLAKVVDDLKNFISKELLPKINRVSCDVH
DITAVIRFQQLNKRLLEVSREFSSNAGLTHTVSSFMLTDRELTSIVGGMAVSAGQKEIMLSSKAIMRRNGLAILSSVNAD
TLVYVIQLPLFGVMDTDCWVIRSSIDCHNIADKYACLARADNGWYCHNAGSLSYFPSPTDCEIHNGYAFCDTLKSLTVP
VTSRECNSNMYTTNYDCKISTSKTYVSTAVLTTMGCLVSCYGHNSCTVINNDKGIIRTLPDGCHYISNKGVDRVQVGNT
VYYLSKEVGKSIVVRGEPLVLKYDPLSFPDDKFDVAIRDVEHSINQTRTFFKASDQLLDLSENRENKNLNKSYILTTLLFV
VMLIIIMAVIGFILYKVLKMIRDNKLKSKSTPGLTV
Q8B0I1
MKCLLYLAFLSIGVNCKFTIVFPHNQKGTWKNVPSNYHYCPSSSDLNWHNDLIGTALQVKMPKSHKAIQADGWMCHA
SKWVTTCDFRWYGPKYITHSIRSFTPSVEQCRESIEQTKQGTWLNPGFPPQSCGYATVTDAEAVIVQVTPHHVLVDEY
TGEWVDSQFINGKCSNDICPTVHNSTTWHSDYKVKGLCDSNLISMDITFFSEDGELSSLGKEGTGFRSNHFAYETGDKA
CKMQYCKHWGVRLPSGVWFEMADQDLFAAARFPECPEGSSISAPSQTSVDVSLIQDVERILDYSLCQETWSKIGAGLPI
SPVDLSYLAPKNPGTGPAFTIINGTLKYFETRYIRVDIAAPILSRMVGMISGTTTERELWDDWAPYEDVEIGPNGVLRTSS
GYKFPLYMIGHGMLDSDLHLSSKAQVFEHPHIQDAASQLPDDETLFFGDTGLSKNPIELVEGWFSGWKSSIASFFFIIGL
IIGLFLVLRVGIYLCIKLKHTKKRQIYTDIEMNRL
P0DTC2
SVASQSIIAYTMSLGAENSVAYSNNSIAIPTNFTISVTTEILPVSMTKTSVDCTMYICGDSTECSNLLLQYGSFCTQLNRA
LTGIAVEQDKNTQEVFAQVKQIYKTPPIKDFGGFNFSQILPDPSKPSKRSFIEDLLFNKVTLADAGFIKQYGDCLGDIAAR
DLICAQKFNGLTVLPPLLTDEMIAQYTSALLAGTITSGWTFGAGAALQIPFAMQMAYRFNGIGVTQNVLYENQKLIANQ
FNSAIGKIQDSLSSTASALGKLQDVVNQNAQALNTLVKQLSSNFGAISSVLNDILSRLDKVEAEVQIDRLITGRLQSLQT
YVTQQLIRAAEIRASANLAATKMSECVLGQSKRVDFCGKGYHLMSFPQSAPHGVVFLHVTYVPAQEKNFTTAPAICHD
GKAHFPREGVFVSNGTHWFVTQRNFYEPQIITTDNTFVSGNCDVVIGIVNNTVYDPLQPELDSFKEELDKYFKNHTSPD
VDLGDISGINASVVNIQKEIDRLNEVAKNLNESLIDLQELGKYEQYIKWPWYIWLGFIAGLIAIVMVTIMLCCMTSCCSC
LKGCCSCGSCCKFDEDDSEPVLKGVKLH
0
0.25
0.5
0.75
1
Score
M aa annotated as Fusion Peptide

Fig. 4. Predictions of the transformer ESM2b T68M fine-tuned for 20 epochs on the test sequences. Annotated FP are underlined. Predictions on each amino acid residue are depicted from white to red (high probability of belonging to FP). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

peptide, ’LTSIVGGMAVSAGQKEIMLSSKA’.

Sequence P07399, a glycoprotein, from class I belonging to the family Aeroviridae, had 18 models correctly predicting the sequence and 4 models that did not predict any sequence. It is worth highlighting that the majority of these models, although correctly predicting the sequence, do not predict the entire sequence. Sliding window approaches miss several amino acid residues in the C-terminus of the peptide whereas, token classification models predict that the sequence starts 10–15 amino acid residues after (except the T68M with 3 epochs) when compared to the annotated one. One possible explanation is related to the biology of the FP itself. Considering the closeness to the VFP from Lassa virus, also from the arenavirus family, it is possible that the FP may contain two distinct components: an N-terminal fusion peptide (GTFTWTLSDSSGVENP), and an internal fusion loop (GGYCLTKWMILAAELKCFGNTAV) [26]. This would indicate that transformer token classification models may be depicting the internal fusion loop to the detriment of the N-terminal fusion peptide while the sliding window approaches favour the N-terminal fusion peptide.

Sequence Q68801 is a class II envelope glycoprotein from a virus belonging to the Flaviridae family. The fusion peptide from this protein was correctly predicted by 17 models, 1 model predicted the corrected sequence but with other subsequences and 4 models did not generate any prediction for this test case. Sequence Q8148, which is also a class II fusion protein from class II from a virus belonging to the same family, represented a more challenging case, being correctly predicted 12 times. 9 models did not achieve any prediction and 1 model predicted a wrong sequence. All the models from the token classification correctly predicted sequence Q68801. Sequence Q81487 was only depicted wrongly in the token classifier from T12 fine-tuned by 15 epochs predicting only 3 AAs (however corresponding to the correct FP). This may indicate that token classification is more effective in predicting peptides from class II, which is again consistent with their ability to predict internal fusion peptides.

Sequence P36334 is a spike glycoprotein from class I from Coronaviridae. Only 3 models, based on physicochemical and RF models, achieved correct prediction and 2 models did not generate any prediction. 16 models wrongly predicted the fusion peptide for this test case (5 of these with 2 subsequences predicted) and 1 model predicted 2 subsequences being one the correct one and the second the same predicted in the models that failed the prediction. The models tend to predict the sequence SKASSRSAIEDLLFDK (with some models extending the peptide to the subsequence SKASSRSAIEDLLFDKVKLSDVGFV), position 140–156, instead of the annotated ’LAATSASLFPPWTAAAGVPFY’, position 206–226. It is very interesting to note that the homologous sequence to "SKASSRSAIEDLLFDKVKLSDVGFV" in SARS-CoV-2 has been shown to be important for membrane fusion as is considered the fusion peptide by some authors.

Regarding the peptide Q8B0I1 that belongs to Vesicular stomatitis virus (VSV), it is not correctly predicted in any of the models and 3 models are able to generate predictions for this test case, which places it as a particularly difficult test case. The FP of VSV is not widely accepted and remains a subject of scientific debate. Here, we considered the existence of bipartite fusion loops as described in [27]. The location of this peptide has been difficult owing to the lack of an obvious stretch of suitable residues in its primary sequence [28] and was only validated with joint high-resolution protein structure and mutagenesis assays, with the structural differences of this peptide resulting in it being considered a novel “class III” fusion protein [27,28]. This arrangement of hydrophobic loops is highly reminiscent of the fusion loop of a class II fusion protein. However, the hydrophobic amino acid residues are shared over two non-contiguous loops, instead of in a contiguous stretch of primary sequence as class II [27]. Furthermore, evidence suggests that other elements of the VSV G protein, such as the transmembrane region are crucial for membrane fusion [29]. The authors envisioned that because of the bipartite nature of the peptide, the only one in the database, and the significant differences with other peptides pose significant challenges to our models. Furthermore, class III is sub-represented in the dataset. Cases like this would greatly benefit from training with a larger dataset and bring space for the improvement of our models.

The last sequence to be tested was the Spike glycoprotein from SARS-CoV-2. Scores are not presented, since there is still some debate regarding to which segment is the actual fusion peptide in this protein. All the transformer models predicted a fusion peptide for this protein. Although with some variation on the start and end amino acid residues, the peptide most predicted is "QIYKTPPIKDFGGFNFSQI" from positions 103–121 of the fusion protein. 3 of the models extend the prediction including more residues in the beginning and at the end of the sequence: IYKTPPIKDFGGFNFSQILPDPSKPSKRSFIEDLLFNK.

Furthermore, the sliding window approach models based on physicochemical features tend to overpredict this peptide, with several subsequence predictions.

We note that several regions have been proposed to act as the SARS-CoV-2 fusion peptide. Currently, the most accepted hypothesis is that the sequence “SFIEDLLFNKVTLADAGFIKQYGDCLGDIAARDLICAQKF” that spans from residues 816–855 in the full length protein (residues 1–40 in the post-cleavage S2 subunit) corresponds to the SARS-CoV-2 fusion peptide [30] and it is now apparent that this region is important for fusion (it is here designated as SCV2-FP1). Other studies consider than only a sub-segment within SCV2-FP1 (residues 816–840) –here designated SCV2-FP1_short - corresponds to the SARS-CoV-2 fusion peptide [31]. It has also been considered that the region spanning residues 816–855 contains two fusion peptides: one that encompasses residues 816–837 and another that comprises residues 835–855 and this is the annotation that is followed in some sequences deposited in UniProt (e.g. Uniprot ID P0DTC2).

On the other hand, findings from other studies indicate that the sequence “MIAQYTSALLAGTITSGWTFGAGAALQIPFAMQMAYRF”, corresponding to residues 867–905 (residues 52–90 in the post-cleavage S2 subunit) – here designated SCV2-FP2 - displays a higher propensity to interact with the membrane and promote membrane leakage and lipid mixing than SCV2-FP1_Short [32]. The same study also investigated an alternative fusion peptide upstream of the furin cleavage site, which also has fusogenic activity but is located in the S1 subunit, which makes it unlikely that it will act as the fusion peptide in the biological context. Cryo-EM studies show that SCV2-FP2 interacts with the membrane and also find that the SCV2-FP1 does not seem to be crucial for fusion, as introducing multiple substitutions in this region does not abolish membrane fusion [33].

Overall, these results indicate that there is still a high uncertainty regarding to which region is actually playing the role of the fusion peptide in the SARS-CoV-2 fusion protein. Some of our predictions are in line with the studies that claim that the sequence “SFIEDLLFNKVTLADAGFIKQYGDCLGDIAARDLICAQKF” that spans from residues 816–855 is the SARS-CoV-2 fusion peptide, which indicates that our methods detected patterns in this sequence that are compatible with such a role. Nonetheless, further experimental studies are needed to further clarify this matter. This test case showcases the uncertainties that exist when defining fusion peptides, posing difficult challenges to predict tools such as the one presented here.

Overall, the models are able to classify fusion peptides belonging to proteins from class I and class II and struggle with class III. Considering the predictions from models based on transformers, just one sequence (out of the 6) from class I got misclassified, while the only protein on the test set from class III was not correctly predicted by any model. This may be due to the particularities of the sequence itself and also because class III peptides are underrepresented in the dataset. Furthermore, sliding window approaches and token classification models, predict different peptides in some sequences, which highlights the different characteristics learned by the models. In general, token classification models outperform sliding window approaches.

3. Discussion

Prediction of viral fusion peptides is a challenging task, but crucial for understanding fusion processes and identifying potential therapeutic targets. Currently, these peptides are either identified through costly lab work or by comparing them to existing peptides. This limited availability of information creates a bias towards certain virus groups. Therefore, a sequence-based prediction method for fusion peptides would be highly valuable, enabling faster identification across multiple viruses.

In this study, we explored different approaches: sliding windows with various feature encodings and machine learning models, as well as token classification using state-of-the-art transformer models and using a holdout dataset to compare both approaches. We recognize that using a small holdout set of 10 test sequences, while not optimal for comparing a large number of models, was necessary given the limited data available. We opted for a more qualitative study, focusing on evaluating the models' behaviors and understanding their limitations in the context of sparse data. This approach allowed us to gain insights into where the models may struggle due to the incomplete or insufficient data, even if the results may not be fully generalizable.

Overall, the transformer ESM2b token classification outperformed the sliding window approach. However, certain sliding window strategies, particularly those based on physicochemical features, showed promising results that were comparable to token classification. This can be attributed to the distinct physicochemical properties exhibited by these peptides. DL approaches coupled with sliding window approach did not yield satisfactory results, likely due to the limited dataset size.

The best models performed well on the low-similarity test sequences, indicating their potential usefulness across diverse viral protein sequences. All predicted sequences were also tested for TMDs, as fusion peptides share similar properties. Even when incorrect sequences were predicted, our models did not overlap with TMD sequences, demonstrating their ability to distinguish between these categories.

The models are able to classify fusion peptides belonging to proteins from class I and class II, while struggling with class III proteins, likely due to unique sequence characteristics and underrepresentation in the dataset.

The best models developed in this work, one of the first in the literature to propose fusion peptide detection, exhibit outstanding performance in predicting fusion peptides within the context of fusion proteins. They can suggest initial sequences for experimental validation, reduce costs, aid in fusion protein research, and identify fusion peptides in dissimilar sequences. These predictions may also challenge existing assumptions in the literature, generating new hypothesis that can be tested in the wet lab. To further enhance these models, enlarging the dataset and enriching underrepresented viral families and fusion protein classes would be crucial. Incorporating evolving transformer models and exploring alternative encoding schemes or hidden Markov models could also improve fusion peptide detection. Additionally, leveraging structural features, considering the availability of protein structure prediction models like AlphaFold [34], could be of interest.

4. Conclusion

Identification of fusion peptides poses significant scientific challenges, relying on costly experimental work, while most current bioinformatics approaches are limited by homology methods that overlook fusion peptides lacking sequence similarity or those from unexplored viruses. In this study, we explore diverse techniques to detect fusion peptides within fusion protein sequences, employing strategic approaches and protein representations. Our models demonstrate high performance in accurately identifying fusion peptides within viral fusion proteins. Notably, physicochemical sliding window approaches and transformer-based token classification models exhibit remarkable performance, with the latter showing superior results. We conducted a rigorous evaluation of the models and provided biological insightful observations on annotated fusion peptides. These results emphasize the potential for improvement, particularly when a completer and more updated dataset is made available. The curated dataset and the models presented make a significant contribution to this field, offering researchers a valuable tool to identify fusion peptides, comprehend fusion mechanisms, and develop targeted therapeutic interventions.

5. Methods

5.1. Viral fusion peptide dataset

Viral fusion peptides were obtained from the ViralFP database, accessible at https://viralfp.bio.di.uminho.pt/ [2], on April 28, 2022. From the initial dataset comprising 743 rows, entries without FP were removed. Additionally, FP entries with parentheses and those with similar FP with minor variations in the beginning and end letters were excluded, keeping the entry most curated or the largest one. This was made as the terminals of FPs are not well defined. Entries with both similar VFP and FP were removed, while entries with equal FP but with different VFP were kept. Entries with very incomplete information were also removed. Modification was made to entry 179, replacing the glycoprotein G from Vesicular stomatitis Indiana virus with a double viral fusion peptide [FRWYGPKY CGYATVT], as demonstrated by Sun et al. [28], and updating the fusion protein to Uniprot entry Q8B0I1. The final curated dataset consists of 403 entries, including 257 unique fusion peptides. The curated dataset and a notebook providing statistics and characterization of the dataset are available in the repository.

5.2. Construction of the hold-out dataset

To evaluate the different methods, a holdout dataset of 10 entries with VFP and FP was created. To achieve this, entries without VFP were excluded from the curated dataset. Using CDHIT [9], sequence similarity at 80 % was calculated, and only sequences that were unique within their cluster were considered. This ensured that no fusion peptide with more than 80 % similarity appeared in both the training set and the hold-out dataset. From these sequences, 9 entries were randomly sampled. In addition to these nine sequences, the fusion protein of SARS-CoV-2, which remains a subject of scientific debate, was included, resulting in a final test dataset containing 10 sequences.

5.3. Construction of datasets for sliding window approach

This section includes a description of the methods for the strategy of the sliding window approach.

Construction of datasets

Initially, segments from VFP were generated. The size of the subsequences played a crucial role, and the authors opted to fix the length to 21 amino acids, effectively capturing fusion peptides. These subsequences were created with a one-amino-acid gap (all possible windows) for ease of comparison and model execution. To ensure consistency, subsequences containing partial FP were excluded, retaining only fully negative or fully positive subsequences. This decision was driven by the lack of well-defined borders for fusion peptides, resulting in potential discrepancies at the beginning and end. Additionally, a border tolerance of 3 amino acids on each end of the fusion peptide was introduced. Subsequences containing amino acids within the border tolerance were also excluded.

This bag of subsequences was combined with TMD sequences, as they share physicochemical similarities with FP. To avoid the presence of similar sequences, a filter based on similarity was made. First, negative subsequences and TMDs with over 70 % similarity to positive sequences were removed. Then, TMDs with more than 70 % similarity with negative subsequences and negative subsequences with more than 70 % similarity with itself were also excluded to avoid excessive similarity in negative samples. CDHIT [9] was used for similarity calculations.

The resultant dataset was composed of 207 positive subsequences, 5819 negative subsequences, and 784 TMDs. The next step involved creating datasets with varying ratios. The positive sequences were retained, one negative subsequence per entry, and 207 TMDs were randomly sampled generating a dataset with 231 negative subsequences and 207 TMD. Furthermore, to create a dataset with 1:1 ratio, 103 negative subsequences and 103 TMDs were randomly sampled from the previous dataset.

5.4. Protein representations

Physicochemical features were calculated with the package ProPythia [14]. All of the available features were calculated. Columns with only zeros, duplicate columns, and no variance columns were also removed. Features were Standard Scaled using scikit-learn [35]. Feature selection was made using the function *mutual_info_classify* also from Scikit learn and excluding features in the 50 % percentile.

Word embedding representations were derived from the Protvec, a Word2Vec model pre-trained on UniprotKB [17]. This model has learned 100 dimension representations for all possible trigrams of amino acids. Two strategies were employed as described in [23]. The first one consisted of the substitution of every trigram on the sequence for the 100-dimension vector, producing a vector for each subsequence of dimension 1900 (100 * (length(sequence) −2). The second approach is to count the frequencies of the trigrams, multiply the 100-dimension vectors for the frequencies and sum column-wise, resulting in each subsequence being represented with a vector of dimension 100.

Sparse one-hot encoding was also derived. Each amino acid is represented by a vector of dimension 21 with '1' in a single position and the remaining 20 positions with '0'. This representation generates vectors of dimension (21 *21).

ESM2b representations [21] were grabbed accordingly to the indications in the repository page https://github.com/facebookresearch/esm. Both T6 8 M (model with 6 layers and 8 M parameters) and T33 650 M (model with 33 layers and 650 M parameters) models were considered. Mean representations (1280 dimension) were considered when applied to ML. Matrix representation (23 * 1280 dimension) were fed to LSTM networks. Contact representations available were also tested with LSTM networks.

5.5. Machine learning

Machine learning strategies were run with the aforementioned protein representations. The models were run using code both from ProPythia and Scikit learn [14,35]. SVM, RF, GB, LR, KNN, GNB, and SGD models were tested. First, a hyperparameter optimization grid search with a 10-Kfold group strategy was run optimizing MCC. The groups were made based on the 80 % similarity to ensure that similar sequences were not on both train and test at the same time. The parameter grids were the default ones used in ProPythia. Next, the best model was evaluated with a 10-fold group cross-validation to ensure robust results. Accuracy, precision, recall, MCC, F1 score and ROC AUC were calculated as described in scikit-learn [35]. Models were retrained on all available data and tested on the holdout dataset.

5.6. Deep learning

Deep learning models were based on LSTM networks and were tested with one hot encoding, Word embeddings (with method 1), and transformers representations. LSTM networks were constructed using Tensorflow [36]. Different architectures were tested and optimized by changing the number of recurrent layers, units, bidirectionality, attention mechanisms, and dense layers and units.

Networks were constituted by one or 2 LSTM or BiLSTM layers with units from 128 and 32 with dropout of 0.2, followed or not by an attention mechanism (as defined in [23], a dense layer with 64 or 32 units, and a Dropout rate of 0.2 followed by a final classification layer with sigmoid activation. The models were trained with Adam optimizer and with binary crossentropy loss. Regularizers l1 [1e–5] and l2 [1e–4] were also added. The number of epochs was set to 500, with reduce learning rate with patience of 30 epochs and factor of 0.2 and early stopping with the patience of 50 epochs.

Similarly to the ML approaches, these models were evaluated using a 10-fold grouped cross-validation. Retrained all available data and test on the holdout set.

5.7. Fine-tuned ESM2b models

The different subsequence datasets were fed to the ESM2b T68M and T33 and finetuned for 3 epochs. As the models were already overfitting, increasing the number of epochs was not considered. The fine-tuned models were then used to predict the test sequences from the holdout dataset.

5.8. Token classification

In token classification, unlike sliding window, the label for each sample will be one integer per token in the input, i.e., the label is a sequence with the length of the VFP with the positions corresponding to the FP labelled as 1. Sequences are padded to 1500. The dataset and labels were then fed to the ESM2b T68M and T1235M transformer models and fine-tuned. ESM2b T68M was finetuned for 3,5,10,15,20 and 30 epochs and ESM2b T12 was finetuned for 3,5,10,15 and 20 epochs. During training, training and validation loss and masked accuracy were assessed. The fine-tuned models were then used to predict test sequences of the holdout dataset.

5.9. Hold-out dataset evaluation

According to the strategy defined, the test sequences were transformed and predicted. In the case of the sliding window approach, each amino acid residue gets the maximum value of all the subsequences where it appears.

For both sliding window and token classification strategies, both results and annotations were binarized and the Jaccard index was calculated. Scores, however, need to be taken with caution as the terminals of the FP are not strictly defined. For an intuitive overview, a Jupyter notebook projecting the predictions of all models was made. The notebook produces images similar to image Fig. 4 with a colour bar indicating the probability of belonging to FP in each position.

We predicted the TMDs from these sequences using DeepTMHMM software [37] to assess if the predictions made by our models would coincide with TMD regions.

CRediT authorship contribution statement

Lousa Diana: Writing – review & editing, Supervision, Funding acquisition, Data curation, Conceptualization. **Rocha Miguel:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Sequeira Ana Marta:** Writing – review & editing, Writing – original draft, Validation, Methodology, Formal analysis, Data curation, Conceptualization.

Declaration of Competing Interest

The authors declare no conflicts of interests.

Acknowledgements

This work was financially supported by "la Caixa" Foundation and

Fundação para a Ciência e a Tecnologia, IP under the grant agreement HR22–00722 – BioPlaTTAR, by the European Union under the Horizon Europe HORIZON-HLTH-2023-DISEASE-03-04, GA 101137419 – Eva-Mobs, and by Fundação para a Ciência e a Tecnologia, IP, through MOSTMICRO-ITQB R&D Unit (UIDB/04612/2020, UIDP/04612/2020), LS4FUTURE Associated Laboratory (LA/P/0087/2020) and iNOVA4-Health R&D Unit (UIDB/04462/2020, UIDP/04462/2020). FCT supported this study under the strategic funding of the UIDB/04469/2020 unit. A.M. Sequeira was supported by FCT and FSE through Ph.D. fellowship 2020.07867.BD.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.csbj.2025.02.011.

Data availability

The study utilized a publicly available dataset obtained from https://viralfp.bio.di.uminho.pt/. This original dataset of fusion proteins was analysed to derive the annotated final dataset. All datasets employed in this study, along with the code for replicating the research and files for result analysis, can be accessed at the following GitHub link: http://github.com/marta-seq/ViralFP.

References

[1] Lozada C, Barlow TMA, Gonzalez S, Lubin-Germain N, Ballet S. Identification and characteristics of fusion peptides derived from enveloped viruses. Front Chem 2021;9:689006.

[2] Moreira P, Sequeira AM, Pereira S, Rodrigues R, Rocha M, Lousa D. ViralFP: a web application of viral fusion proteins. Front Med Technol 2021;3:722392.

[3] Rey FA, Lok SM. Common features of enveloped viruses and implications for immunogen design for next-generation vaccines. Cell 2018;172(6):1319–34.

[4] Apellániz B, Huarte N, Largo E, Nieva JL. The three lives of viral fusion peptides. Chem Phys Lipids 2014;181:40–55.

[5] Podbilewicz B. Virus and cell fusion mechanisms. Annu Rev Cell Dev Biol 2014;30(1):111–39.

[6] Epand RM. Fusion peptides and the mechanism of viral fusion. Biochim Biophys Acta BBA - Biomembr 2003;1614(1):116–21.

[7] Wu S, Han J, Liu R, Liu J, Lv H. A computational model for predicting fusion peptide of retroviruses. Comput Biol Chem 2016;61:245–50.

[8] Wu S, Wu X, Tian J, Zhou X, Huang L. PredictFP2: a new computational model to predict fusion peptide domain in all retroviruses. IEEE/ACM Trans Comput Biol Bioinform 2020;17(5):1714–20.

[9] Fu L, Niu B, Zhu Z, Wu S, Li W. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics 2012;28(23):3150–2.

[10] Dietterich TG. Machine learning for sequential data: a review. In: Caelli T, Amin A, Duin RPW, De Ridder D, Kamel M, editors. Structural, syntactic, and statistical pattern recognition [Internet]. Berlin, Heidelberg: Springer Berlin Heidelberg; 2002. p. 15–30 (Goos G, Hartmanis J, Van Leeuwen J, editors. Lecture Notes in Computer Science; vol. 2396). Available from: ⟨http://link.springer.com/10.1007/3-540-70659-3_2⟩.

[11] Jurtz V.I., Johansen A.R., Nielsen M., Almagro Armenteros J.J., Nielsen H., Sønderby C.K., et al. An introduction to deep learning on biological sequence data: examples and solutions. Valencia A, editor. Bioinformatics; 2017 Nov 15: 33(22): 3685–90.

[12] Xu Y, Verma D, Sheridan RP, Liaw A, Ma J, Marshall NM, et al. Deep dive into machine learning models for protein engineering. J Chem Inf Model 2020;60(6):2773–90.

[13] Müller A.T., Gabernet G., Hiss J.A., Schneider G. modlAMP: Python for antimicrobial peptides. Valencia A, editor. Bioinformatics; 2017 Sep 1: 33(17): 2753–5.

[14] Sequeira AM, Lousa D, Rocha M. ProPythia: A Python package for protein classification based on machine and deep learning. Neurocomputing 2022;484:172–82.

[15] Abdin O, Nim S, Wen H, Kim PM. PepNN: a deep attention model for the identification of peptide binding sites. Commun Biol 2022;5(1):503.

[16] Bileschi ML, Belanger D, Bryant DH, Sanderson T, Carter B, Sculley D, et al. Using deep learning to annotate the protein universe. Nat Biotechnol 2022;40(6):932–7.

[17] Asgari E, Mofrad MRK. In: Kobeissy FH, editor. Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics, 10. PLoS One; 2015, e0141287.

[18] Brandes N, Ofer D, Peleg Y, Rappoport N, Linial M. ProteinBERT: a universal deep-learning model of protein sequence and function. In: Martelli PL, editor. Bioinformatics, 38; 2022. p. 2102–10.

[19] Rives A, Meier J, Sercu T, Goyal S, Lin Z, Liu J, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc Natl Acad Sci 2021;118(15):e2016239118.

[20] Strodthoff N, Wagner P, Wenzel M, Samek W. UDSMProt: universal deep sequence models for protein classification. In: Ponty Y, editor. Bioinformatics, 36; 2020. p. 2401–9.

[21] Lin Z, Akin H, Rao R, Hie B, Zhu Z, Lu W, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. Science 2023;379(6637):1123–30.

[22] Chollet F. Deep learning with Python. 2nd edition. Shelter Island: Manning Publications; 2021. p. 478.

[23] Sequeira AM, Rocha M. Recurrent deep neural networks for enzyme functional annotation. In: Rocha M, Fdez-Riverola F, Mohamad MS, Casado-Vara R, editors. n: Proceedings of the fifteenth international conference (PACBB 2021), practical applications of computational biology & bioinformatics [Internet]. Cham: Springer International Publishing; 2022 [Accessed 27 July 2024]. p. 62–73. (Lecture Notes in Networks and Systems; vol. 325). Available from: ⟨https://link.springer.com/10.1007/978-3-030-86258-9_7⟩.

[24] Shi Q, Chen W, Huang S, Wang Y, Xue Z. Deep learning for mining protein data. Brief Bioinform 2021;22(1):194–218.

[25] Sequeira A.M., Gomes I., Rocha M. Word embeddings for protein sequence analysis. In: : Proceedings of the 2023 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB) [Internet]. Eindhoven, Netherlands: IEEE; 2023 [Accessed 27 July 2024]. p. 1–8. Available from: https://ieeexplore.ieee.org/document/10264897/.

[26] Pennington HN, Lee J. Lassa virus glycoprotein complex review: insights into its unique fusion machinery. Biosci Rep 2022;42(2):BSR20211930.

[27] Sun X, Belouzard S, Whittaker GR. Molecular architecture of the bipartite fusion loops of vesicular stomatitis virus glycoprotein G, a Class III viral fusion protein. J Biol Chem 2008;283(10):6418–27.

[28] Sun X, Roth SL, Bialecki MA, Whittaker GR. Internalization and fusion mechanism of vesicular stomatitis virus and related rhabdoviruses. Future Virol 2010;5(1):85–96.

[29] Ci Y, Yang Y, Xu C, Shi L. Vesicular stomatitis virus G protein transmembrane region is crucial for the hemi-fusion to full fusion transition. Sci Rep 2018;8(1):10669.

[30] Lai AL, Freed JH. SARS-CoV-2 fusion peptide has a greater membrane perturbating effect than SARS-CoV with highly specific dependence on Ca2+. J Mol Biol 2021;433(10):166946.

[31] Gorgun D, Lihan M, Kapoor K, Tajkhorshid E. Binding mode of SARS-CoV-2 fusion peptide to human cellular membrane. Biophys J 2021;120(14):2914–26.

[32] Basso LGM, Zeraik AE, Felizatti AP, Costa-Filho AJ. Membranotropic and biological activities of the membrane fusion peptides from SARS-CoV spike glycoprotein: the importance of the complete internal fusion peptide domain. Biochim Biophys Acta BBA - Biomembr 2021;1863(11):183697.

[33] Shi W, Cai Y, Zhu H, Peng H, Voyer J, Rits-Volloch S, et al. Cryo-EM structure of SARS-CoV-2 postfusion spike in membrane. Nature 2023;619(7969):403–9.

[34] Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature 2021;596(7873):583–9.

[35] Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., et al. Scikit-learn: machine learning in Python; 2012 [Accessed 27 July 2024]. Available from: https://arxiv.org/abs/1201.0490.

[36] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems [Internet]. arXiv; 2016 [Accessed 27 July 2024]. Available from: https://arxiv.org/abs/1603.04467.

[37] Hallgren J., Tsirigos K.D., Pedersen M.D., Almagro Armenteros J.J., Marcatili P., Nielsen H., et al. DeepTMHMM predicts alpha and beta transmembrane proteins using deep neural networks [Internet]. 2022 [Accessed 27 July 2024]. Available from: http://biorxiv.org/lookup/doi/10.1101/2022.04.08.487609.