DeePSP-GIN: identification and classification of phage structural proteins using predicted protein structure, pretrained protein language model, and graph isomorphism network

Muhit Islam Emon, Badhan Das, Ashrith Reddy Thukkaraju, Liqing Zhang
Department of Computer Science
Virginia Tech
Blacksburg, Virginia, USA
lqzhang@cs.vt.edu

ABSTRACT

Phages are vital components of the microbial ecosystem, and their functions and roles are largely determined by their structural proteins. Accurately annotating phage structural proteins (PSPs) is essential for understanding phage biology and their interactions with bacterial hosts, which can pave the way for innovative strategies to combat bacterial infections and develop phage-based therapies. However, the sequence diversity of PSPs makes their identification and annotation challenging. While various computational methods are available for predicting PSPs, they currently lack the integration of protein structural information, an important aspect for understanding protein function. With the advent of deep learning models, protein structures can be predicted accurately and quickly from protein sequences, creating new opportunities for PSP prediction and analysis. We developed DeePSP-GIN, a graph isomorphism network (GIN) - based deep learning model leveraging predicted protein structures and protein language model for PSP identification and classification. To the best of our knowledge, DeePSP-GIN is the first method utilizing predicted protein structural information for PSP prediction tasks. It offers dual functionality of identifying PSP and non-PSP sequences and classifying PSPs into seven major classes. DeePSP-GIN converts predicted protein structures from PDB 3D coordinates into graphs and extracts node features from protein language model-generated embeddings. The GIN is then applied to the constructed graphs to learn the discriminating features. The experimental results show that DeePSP-GIN outperforms the state-of-the-art methods in both PSP identification and classification tasks in terms of F1-score. DeePSP-GIN achieves a 1.04% higher F1-score than the nearest competing method in the PSP identification task. Additionally, its overall F1-score in the PSP classification task is approximately 34.38% higher than that of the second-best method. The source code of DeePSP-GIN is available at https://github.com/muhit-emon/DeePSP-GIN under the MIT license.

BCB '24, November 22–25, 2024, Shenzhen, China

ACM ISBN 979-8-4007-1302-6/24/11
https://doi.org/10.1145/3698587.3701371

CCS CONCEPTS

• **Applied computing** → **Bioinformatics**; • **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

phage structural/virion proteins, protein 3D structures, graph neural network, protein language model, deep learning

ACM Reference Format:
Muhit Islam Emon, Badhan Das, Ashrith Reddy Thukkaraju, Liqing Zhang. 2024. DeePSP-GIN: identification and classification of phage structural proteins using predicted protein structure, pretrained protein language model, and graph isomorphism network. In *15th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '24), November 22–25, 2024, Shenzhen, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3698587.3701371

1 INTRODUCTION

Bacteriophages, commonly known as phages, are viruses that primarily infect bacteria. They are the most abundant and widespread biological entities on the Earth [10]. They exert a profound impact on microbial ecosystems, regulating their dynamics through bacterial lysis and facilitating horizontal gene transfer [15]. Beyond ecological impacts, phages play critical roles in applications ranging from food safety [6] and disease diagnostics [34] to genetic engineering of bacterial genomes [2] and phage therapy [3].

Phage structural proteins (PSPs), also referred to as phage virion proteins (PVPs), form the protective outer shell of a phage, safeguarding its genetic material [21]. PSPs play a vital role in how phages recognize and interact with their hosts [7]. Accurate annotation of PSPs is an important step for understanding the biological properties of phages and has numerous applications, including predicting phage hosts [5], designing phage-based antibacterial drugs as an alternative to antibiotics [18, 23], and harnessing phages to develop therapies against bacterial infections [27]. Experimental methods such as mass spectrometry and crystallography for identifying PSPs are time-intensive and costly. Additionally, the high sequence diversity of PSPs makes their detection challenging [31].

To address these challenges, several computational tools utilizing machine learning and deep learning algorithms have been developed for the identification and classification of PSPs. Most of these tools have been reviewed by Kabir et al. [21]. PVP-SVM [26], PVPred [11], PhagePred [28], and PVPred-SCM [8] employ traditional machine learning algorithms, such as support vector machine

(SVM), naive bayes (NB), and the scoring card method (SCM). Additionally, Meta-iPVP [9] and iPVP-MCV [17] are ensemble methods designed for identifying PSPs. More recently, deep learning-based methods have been developed for PSP prediction, including PhaVIP [32], DeePVP [13], PhANNs [7], and VirionFinder [14]. PhaVIP uses the Vision Transformer, an image classification model, to predict structural proteins by encoding protein sequences into unique images through chaos game representation. DeePVP and VirionFinder both utilize convolutional neural networks (CNNs) as classifiers. PhANNs is a feedforward neural network-based method that leverages k-mer frequency and physicochemical properties of protein sequences to predict PSPs. All available tools, except PhaVIP, DeePVP, and PhANNs, are limited to binary PSP identification, classifying input protein sequences as PSP or non-PSP. In contrast, PhaVIP, DeePVP, and PhANNs extend beyond binary identification by performing multi-class classification, assigning identified PSPs to categories such as major capsid, major tail, or portal. However, their performance on the multi-class classification task is unsatisfactory on the benchmark dataset constructed by Shang et al. [32].

Numerous research studies have demonstrated that predicted protein structures are valuable for accurately predicting protein functions [16, 29, 36, 37]. Advancements in deep learning techniques now enable accurate and rapid prediction of protein structures solely from primary sequences [20, 25]. Concurrently, protein language models have proven effective in learning representations of protein sequences, showing promising results across various bioinformatics prediction tasks [4, 12, 38]. Inspired by these advancements, we designed a PSP predictor leveraging predicted protein structures and pretrained protein language model.

In this work, we introduce DeePSP-GIN, a graph-based model for predicting phage structural proteins. DeePSP-GIN employs ESM-2 [24], a pretrained protein language model, to generate sequence representations and uses ESMFold [25] to predict the corresponding protein structures from input phage protein sequences. It then utilizes the graph isomorphism network (GIN) [35] to capture the structural information and the spatial relationships among residues. DeePSP-GIN serves two functions. First, it performs binary classification to determine whether a protein is a PSP or non-PSP. Second, for proteins identified as PSPs, it provides detailed annotations through multi-class classification, categorizing them into one of seven classes: “portal”, “major capsid”, “minor capsid”, “major tail”, “minor tail”, “baseplate”, and “tail fiber”. We utilized the dataset developed by Shang et al. [32], which incorporates the latest phage protein annotations from the RefSeq database (December 2022), for training and evaluating DeePSP-GIN. Testing on this benchmark dataset demonstrated that DeePSP-GIN outperforms state-of-the-art deep learning methods, including PhaVIP, DeePVP, PhANNs, and VirionFinder, in both binary and multi-class PSP classification tasks in terms of F1-score.

2 MATERIALS AND METHODS

2.1 Dataset

The PhaVIP [32] dataset was employed for training and evaluating our models. This dataset, obtained from the RefSeq viral protein database in December 2022, underwent rigorous selection criteria applied by the PhaVIP authors to ensure high data quality.

The authors of PhaVIP divided the dataset into training and test sets using two distinct strategies. The first approach, termed "splitting by time," segregated the data into a training set comprising sequences released before December 2020 and a test set consisting of sequences released thereafter. The second approach, referred to as "splitting by similarity," employed the similarity thresholds of 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9 to partition the dataset into six pairs of training and test sets. This partitioning was performed using the clustering tool GraphPart [33], ensuring that the test set sequences had similarity below the specified threshold to those in the corresponding training set. Similarity was computed as the product of pair-wise sequence identity and alignment coverage.

The PhaVIP authors provided the time-split sets for the binary classification task and the similarity-split sets for the multi-class classification task. Consequently, we utilized the time-split sets to train and test our model for the binary classification task, and the similarity-split sets for the multi-class classification task. The time-split dataset mirrors the scenario of using known PSPs to identify new ones, while the similarity-split sets aid in evaluating a model's ability to predict divergent PSPs.

We randomly selected 10% of the PSP sequences and 10% of the non-PSP sequences from the time-split training set to create a validation set. The validation set was used to select the hyper-parameters. To balance the time-split training set, we randomly sampled non-PSP sequences to match the number of PSP sequences. Finally, for the binary PSP classification task, our training dataset comprises 23,987 PSP sequences and 23,987 non-PSP sequences, while the test dataset contains 7,177 PSP sequences and 10,090 non-PSP sequences. For the multi-class classification task, we followed the similar data distribution as PhaVIP. Supplementary Tables S7 and S8 show the number of sequences in the time-split and similarity-split datasets employed for binary and multi-class classification tasks respectively.

2.2 Overview of DeePSP-GIN

The DeePSP-GIN workflow is illustrated in Fig. 1. DeePSP-GIN takes a phage protein sequence as input and extracts spatial relationships among residues from the predicted 3D structural information. Moreover, it harnesses a pretrained protein language model to generate per-residue embeddings that encapsulate semantic relationships among residues. Subsequently, a residue-level graph is constructed from the predicted protein 3D structure, with edges representing structural information and residue embeddings serving as initial node feature vectors. Then, the GIN is employed to learn the graph representation from the graph data. This representation is fed into the multi-layer perceptron binary classifier to classify the input sequence as PSP or non-PSP. Furthermore, for identified PSPs, the graph representation is processed by a multi-class classifier, another multi-layer perceptron, to yield detailed annotations. The multi-class classifier can categorize seven types of PSPs: “portal”, “major capsid”, “minor capsid”, “major tail”, “minor tail”, “baseplate”, and “tail fiber”.

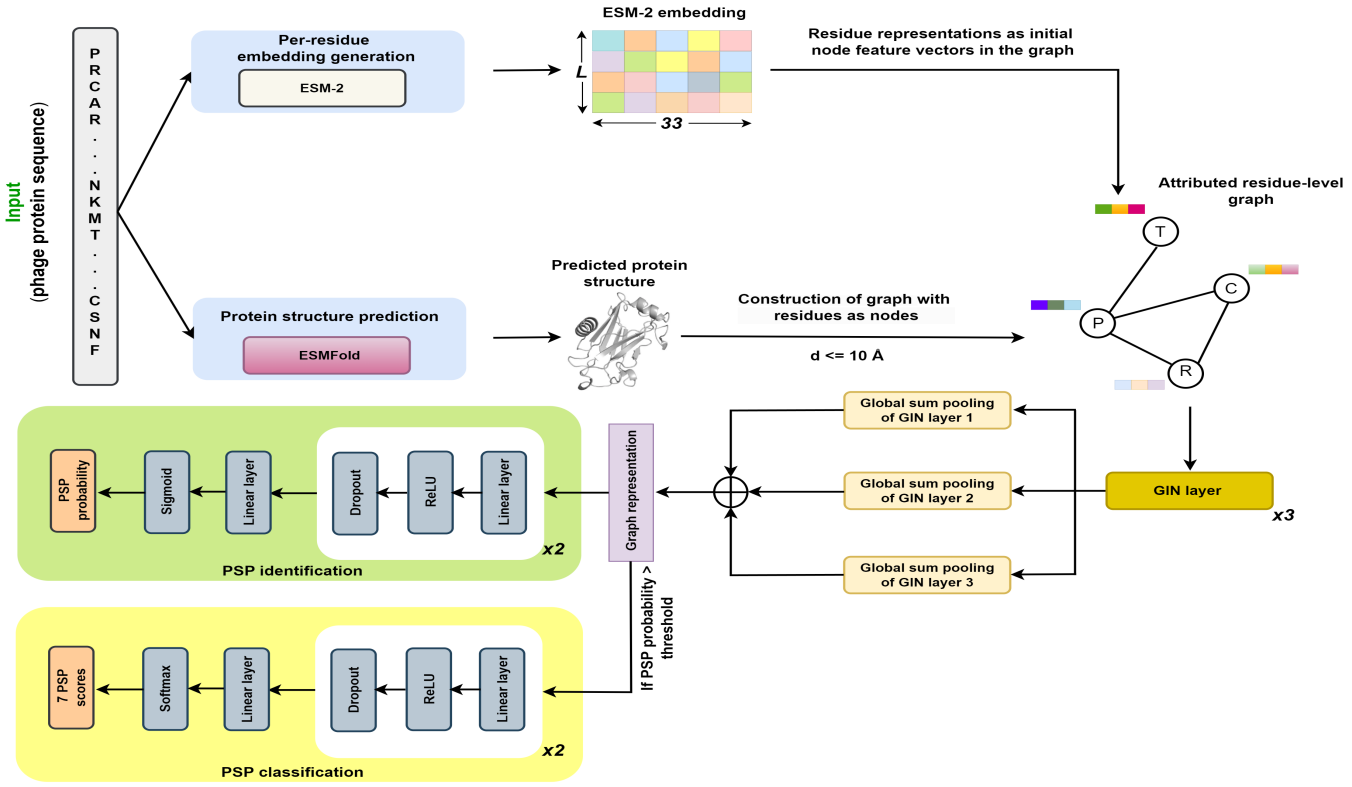

Figure 1: The workflow of DeePSP-GIN. DeePSP-GIN takes a phage protein sequence as input and feeds it into ESMFold and ESM-2 pretrained protein language model to obtain predicted structure and residue embeddings. A graph is built from the predicted structure, with residues as nodes and edges connecting spatially close residues. Node features come from the protein language model embeddings. Three GIN layers are employed to capture the structure information and the spatial relationships among the residues. The graph is then transformed into a fixed-length vector through sum pooling. An MLP processes this vector to calculate a PSP score, indicating the probability that the input protein is a PSP, and another MLP calculates likelihood scores for each of the 7 PSP classes to determine the most likely class.

2.3 Prediction of Protein Structures

To capture spatial information of each residue, we applied the deep learning folding algorithm ESMFold (esmfold_v1) [25] to predict protein structure. It employs the ESM-2 large-scale protein language model to achieve atomic-resolution structure prediction without relying on multiple sequence alignment. Notably, ESMFold accomplishes this feat with remarkable efficiency, yielding a significant 60-fold speedup compared to state-of-the-art methods while maintaining comparable accuracy. We used the pretrained ESMFold model to predict the structures of the protein sequences in our dataset. Since experimentally defined structures are unavailable for most proteins in our dataset, we utilized ESMFold to predict them computationally.

2.4 Protein Language Model Representations

We employed the pretrained protein language model named ESM-2 (esm2_t33_650M_UR50D) [24] to extract sequence representation for each residue. This model, comprising 33 transformer encoder layers and 650M parameters, was pretrained on millions of protein sequences from the UniRef50 database using a BERT-based architecture and a self-supervised masked language modeling objective. Following the implementation of [30], we used the pretrained ESM-2 model to generate 33 dimensional embeddings for each residue, which were subsequently normalized using a sigmoidal function.

2.5 Graph Construction

We represented each input protein sequence as a graph, built from its predicted structure. We utilized Graphein [19] to facilitate the processing of graph construction. For each sequence, ESMFold generates a PDB file containing the 3D coordinates of atoms in individual amino acid residues. We computed the coordinate of each residue by averaging the coordinates of all atoms that compose the residue. Specifically, for a given residue R comprising N atoms, we calculated the x-coordinate of R by averaging the x-coordinates of all constituent atoms. Similarly, we computed the y- and z-coordinates of R by averaging the corresponding coordinates of the N atoms. Then, we constructed a graph $G = (V, E)$ where each node $v \in V$ corresponds to a residue in the protein, associated with a 33 dimensional initial feature vector $x_v = h_v^{(0)}$ derived

from ESM-2 embeddings. Each edge $e \in E$ connects two residues that are spatially close to one another. In this study, we used the distance information between residues to represent their spatial relationships. Specifically, residues with a euclidean distance of no more than a threshold d were connected by an edge in the graph. This threshold d was set to 10Å, as it yielded the highest F1-score on the validation set.

2.6 Graph Isomorphism Network Layers

In this study, PSP prediction was formulated as a graph classification task. We employed the GIN [35] to extract the structural information from the graphs constructed using the predicted protein structures. The GIN model learns node representations, which are then aggregated to represent the entire graph.

Node Level Representations: The first step in the GIN learning process is creating node-level representations. The representation of a node v at the kth layer is computed using the following equation:

$$h_v^{(k)} = \text{MLP}^k\left(\left(1+\epsilon^k\right)\cdot h_v^{(k-1)} + \sum_{u\in N(v)} h_u^{(k-1)}\right) \quad (1)$$

Here, GIN employs Multi-Layer Perceptrons (MLPs) as the nonlinear function to learn node representations, $h_v^{(k)}$ is the representation or feature vector of node v at the kth layer, $N(v)$ is the set of neighbouring nodes of v, and ϵ is a learnable parameter by gradient descent. We used two-layer perceptrons with an output dimension of 32, resulting in node representations of a fixed size of 32 dimensions. To simplify the model, we set ϵ to 0 following [35].

Graph Level Representation: Once node-level representations are obtained, a graph-level readout function is applied to produce the graph-level representation. The graph representation for GIN is computed as follows:

$$h_G = \text{CONCAT}\left(\text{READOUT}\left(\left\{h_v^{(k)} | v \in G\right\}\right) | k = 1, ..., K\right) \quad (2)$$

Where K is the number of GIN layers. In our setting, we employed three GIN layers. Unlike other graph neural networks (GNNs) that consider only the final layer's representation, GIN aggregates node-level representations from all K layers, enhancing its representational power. We used summation pooling as the readout function, which has more expressive power than mean and max aggregators in capturing graph structures [35]. With three GIN layers and each node representation of size 32 dimensions, our concatenated graph-level representation has a size of 96 dimensions.

2.7 Multi-Layer Perceptrons

2.7.1 PSP Identification MLP. The graph representation from the GIN module is fed into an MLP for PSP identification. This MLP includes two hidden layers with ReLU [1] activation and an output layer. Dropout regularization (10%) is applied after each hidden layer to prevent overfitting. The Adam [22] optimizer updates the weights, and binary cross-entropy is used as the loss function. The output layer uses sigmoid activation to produce a PSP score between 0 and 1, classifying proteins as PSPs if the score exceeds 0.5 (default).

2.7.2 PSP Classification MLP. For proteins classified as PSPs, their graph representation is passed to another MLP for PSP class prediction. This MLP also has two hidden layers with ReLU activation and an output layer. Dropout regularization (10%) is applied after each hidden layer, and the Adam optimizer updates the weights. Negative log-likelihood is used as the loss function, and a softmax layer generates likelihood scores for seven PSP classes. The class with the highest score is chosen as the final prediction.

2.8 Experimental Setup

We employed a batch size of 128. In the training process, the number of epochs was set to 50, and the learning rate was 1e-3. Our experiments were conducted on an Ubuntu 18.04 machine equipped with 128 GB memory, Intel Core i9-9820X processor, and an NVIDIA Titan RTX GPU (24 GB memory) using PyTorch 1.13.

3 RESULTS

We compared the performance of our tool against four state-of-the-art deep learning methods, including PhaVIP [32], DeePVP [13], PhANNs [7], and VirionFinder [14] on the time-split test data for binary classification and the similarity-split data for multi-class classification. Additionally, We validated our tool and these competing methods for annotating PSPs in the *mycobacteriophage* PDRPxv genome and the Salmonella phage ZK22 genome (supplementary information sections B and C). We also compared the running time of our tool with the competing methods for PSP annotation on these genomes (supplementary information section D). Finally, we conducted ablation studies to examine the impact of three key aspects on our tool's performance: (1) using other GNN variants in place of GIN, (2) employing one-hot encoding (OHE) for residues instead of protein language model representations, and (3) excluding structural features from the model. Details of these ablation experiments can be found in supplementary information sections E.1-E.3.

3.1 Evaluation Metrics

To evaluate our tool, we employed a range of standard metrics, including precision, recall, F1-score, and area under the receiver operating characteristic curve (AUROC). These are the commonly used metrics for assessing PSP prediction performance, as noted in [21]. Details of the evaluation metrics can be found in supplementary information section A.

3.2 PSP Identification Performance Comparison on Time-Split Test Dataset

We evaluated the performance of DeePSP-GIN and other tools for PSP identification (binary PSP classification) on the test dataset split based on time. To further demonstrate the robustness of DeePSP-GIN, we conducted 5-fold cross-validation on the time-split training data, with the results presented in supplementary Table S9. We retrained PhaVIP, PhANNs, and DeePVP using our time-split training data with the suggested hyper-parameters since their source codes allow for retraining. However, VirionFinder does not offer a retraining functionality, so we applied it directly to the test data. The macro-average F1-scores of the tools for the binary PSP classification task are reported in Table 1. Moreover, detailed class-wise performance is presented in supplementary Table S10. DeePSP-GIN yielded the highest precision, recall, and F1-score for both PSP and non-PSP classes compared to the competing methods. The ROC

Table 1: Performance comparison of DeePSP-GIN with other methods for the binary classification task, evaluated using macro-average F1-score on the test data split by time

Method	Macro-average F1-score
DeePSP-GIN	**0.97**
PhaVIP	0.94
PhANNs	0.82
VirionFinder	0.80
DeePVP	0.96

curves of all the methods on the test dataset split by time are shown in supplementary Fig. S1. The results demonstrate that DeePSP-GIN achieved the highest AUROC score on the time-split test dataset.

3.3 PSP Classification Performance Comparison on Similarity-Split Test Dataset

Among the tools PhaVIP, DeePVP, PhANNs, and VirionFinder, VirionFinder does not provide detailed annotations of PSPs. Therefore, for the multi-class classification task, we compared DeePSP-GIN with PhaVIP, DeePVP, and PhANNs on the PhaVIP test sets split based on similarity. The PhaVIP authors provided six pairs of training and test sets with decreasing similarity, where the product of sequence identity and coverage was used to control the maximum similarity between the training and test sets. We performed 5-fold cross-validation of DeePSP-GIN on these six training sets, and the results are presented in supplementary Tables S11-S16. These results highlight the robustness of DeePSP-GIN in the multi-class classification task. Since the authors of PhaVIP retrained both DeePVP and PhANNs on these training sets and reported their performances on the corresponding test sets, we directly obtained the performance metrics of the methods from the PhaVIP authors' reports. The macro-average F1-scores of DeePSP-GIN, PhaVIP, DeePVP, and PhANNs are shown in Fig. 2. The detailed classification performances of the tools can be found in supplementary Tables S17-S22. For PhaVIP, PhANNs, and DeePVP, we observe an increasing trend in performance as train-test similarity increases. This improvement is expected since these methods rely exclusively on sequence-based features, leading to better outcomes when the training and test data are more similar. In contrast, DeePSP-GIN shows minimal variation in performance across different similarity thresholds. This suggests that DeePSP-GIN's integration of structural data with language model embeddings reduces its reliance on sequence similarity. The results show a clear performance gap between DeePSP-GIN and other state-of-the-art methods, showcasing its enhanced ability to classify PSPs across a wide range of similarities. While other methods' performances on the small class (minor capsid) are unsatisfactory, DeePSP-GIN achieved better performance across all classes, including the smaller ones, as depicted in supplementary Fig. S2.

4 CONCLUSION

In this study, we proposed DeePSP-GIN, a GIN-based deep learning model designed to identify and classify PSPs. DeePSP-GIN performs two functions: it classifies an input phage protein sequence as either

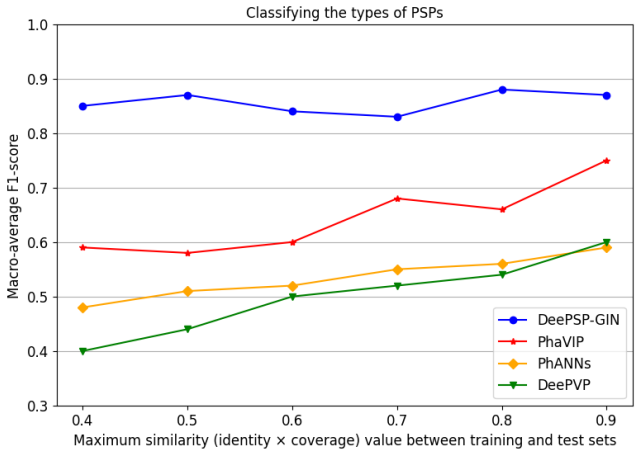

Figure 2: Performance comparison of DeePSP-GIN with other methods for the multi-class classification task on the test data split by similarity at different thresholds.

PSP or non-PSP and predicts the specific type of the identified PSP. DeePSP-GIN leverages the pretrained protein language model ESM-2 to generate per-residue representations and employs the protein language model-powered folding algorithm ESMFold to predict protein structures. Graphs are then constructed from the protein structures with residues as nodes. Two nodes are connected by an edge if the corresponding residues are spatially close in the 3D structure, based on a predefined distance threshold. The per-residue embeddings generated by the protein language model serve as initial features for the nodes. By applying the GIN to the graphs, DeePSP-GIN effectively captures structural features and spatial relationships among residues, resulting in enhanced performance compared to existing deep learning methods for both binary and multi-class classification tasks.

We evaluated DeePSP-GIN using benchmark datasets from PhaVIP, including time-split data to test its capability in identifying novel PSPs, and similarity-split data with increasing difficulty to assess its ability to classify divergent PSPs. Extensive testing on these datasets demonstrated that DeePSP-GIN outperforms state-of-the-art deep learning methods in terms of F1-score, highlighting the significant contributions of structural features and protein language model embeddings for improved prediction performance. It achieves a 1.04% higher F1-score in PSP identification and a substantial 34.38% improvement in overall F1-score for PSP classification compared to the second-best method. Our experiments demonstrate that existing methods struggle with small classes in the multi-class classification task, whereas DeePSP-GIN shows considerable improvement (see supplementary Tables S17-S22). For instance, while PhaVIP, PhANNs, and DeePVP achieve F1-scores of 0.37, 0.31, and 0.23, respectively, for the small class "minor capsid", DeePSP-GIN attains an F1-score of 0.70. Additionally, when evaluated on independent PSP sets from the *mycobacteriophage* PDRPxv and Salmonella phage ZK22 genomes, DeePSP-GIN achieves the highest F1-score compared to other tools (supplementary Tables S1 and S2). While DeePSP-GIN demonstrates better performance over

existing methods in PSP identification and classification, it requires additional computational time (supplementary Table S3) because of the time-intensive protein 3D structure prediction step.

Our ablation studies (supplementary information sections E.1-E.3) revealed that GIN outperformed other GNN variants such as GCN, GAT, and GraphSAGE. Additionally, using OHE for residues instead of protein language model embeddings resulted in a performance decline, and excluding structural features led to a drop in DeePSP-GIN's performance, reaffirming the importance of these components.

Future enhancements to DeePSP-GIN will focus on expanding its capabilities and improving performance. To broaden its applicability, we plan to incorporate additional types of PSPs, such as head-tail joining and collar proteins. Furthermore, exploring new features like protein domain information may provide additional performance gains. Another promising direction is fine-tuning the pretrained protein language model.

REFERENCES

[1] AF Agarap. 2018. Deep Learning Using Rectified Linear Units (ReLU). *arXiv preprint arXiv:1803.08375* (2018).

[2] Mireille Ansaldi. 2015. Bacterial genome remodeling through bacteriophage recombination. *FEMS microbiology letters* 362, 1 (2015), 1–10.

[3] Taher Azimi, Mehrdad Mosadegh, Mohammad Javad Nasiri, Sahar Sabour, Samira Karimaei, and Ahmad Nasser. 2019. Phage therapy as a renewed therapeutic approach to mycobacterial infections: a comprehensive review. *Infection and Drug Resistance* (2019), 2943–2959.

[4] Michael Bernhofer and Burkhard Rost. 2022. TMbed: transmembrane proteins predicted through language model embeddings. *BMC bioinformatics* 23, 1 (2022), 326.

[5] Dimitri Boeckaerts, Michiel Stock, Bjorn Criel, Hans Gerstmans, Bernard De Baets, and Yves Briers. 2021. Predicting bacteriophage hosts based on sequences of annotated receptor-binding proteins. *Scientific reports* 11, 1 (2021), 1467.

[6] Harald Brüssow and Frank Desiere. 2001. Comparative phage genomics and the evolution of Siphoviridae: insights from dairy phages. *Molecular microbiology* 39, 2 (2001), 213–223.

[7] Vito Adrian Cantu, Peter Salamon, Victor Seguritan, Jackson Redfield, David Salamon, Robert A Edwards, and Anca M Segall. 2020. PhANNs, a fast and accurate tool and web server to classify phage structural proteins. *PLoS computational biology* 16, 11 (2020), e1007845.

[8] Phasit Charoenkwan, Sakawrat Kanthawong, Nalini Schaduangrat, Janchai Yana, and Watshara Shoombuatong. 2020. PVPred-SCM: improved prediction and analysis of phage virion proteins using a scoring card method. *Cells* 9, 2 (2020), 353.

[9] Phasit Charoenkwan, Chanin Nantasenamat, Md Mehedi Hasan, and Watshara Shoombuatong. 2020. Meta-iPVP: a sequence-based meta-predictor for improving the prediction of phage virion proteins using effective feature representation. *Journal of Computer-Aided Molecular Design* 34, 10 (2020), 1105–1116.

[10] Ana Georgina Cobián Güemes, Merry Youle, Vito Adrian Cantú, Ben Felts, James Nulton, and Forest Rohwer. 2016. Viruses as winners in the game of life. *Annual review of virology* 3, 1 (2016), 197–214.

[11] Hui Ding, Peng-Mian Feng, Wei Chen, and Hao Lin. 2014. Identification of bacteriophage virion proteins by the ANOVA feature selection and analysis. *Molecular BioSystems* 10, 8 (2014), 2229–2235.

[12] Yitian Fang, Yi Jiang, Leyi Wei, Qin Ma, Zhixiang Ren, Qianmu Yuan, and Dong-Qing Wei. 2023. DeepProSite: structure-aware protein binding site prediction using ESMFold and pretrained language model. *Bioinformatics* 39, 12 (2023), btad718.

[13] Zhencheng Fang, Tao Feng, Hongwei Zhou, and Muxuan Chen. 2022. DeePVP: Identification and classification of phage virion proteins using deep learning. *Gigascience* 11 (2022), giac076.

[14] Zhencheng Fang and Hongwei Zhou. 2021. VirionFinder: identification of complete and partial prokaryote virus virion protein from virome data using the sequence and biochemical properties of amino acids. *Frontiers in microbiology* 12 (2021), 615711.

[15] Lucía Fernández, Ana Rodríguez, and Pilar García. 2018. Phage or foe: an insight into the impact of viral predation on microbial communities. *The ISME journal* 12, 5 (2018), 1171–1179.

[16] Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. 2021. Structure-based protein function prediction using graph convolutional networks. *Nature communications* 12, 1 (2021), 3168.

[17] Haitao Han, Wenhong Zhu, Chenchen Ding, and Taigang Liu. 2021. iPVP-MCV: A multi-classifier voting model for the accurate identification of phage virion proteins. *Symmetry* 13, 8 (2021), 1506.

[18] Shayla Hesse and Sankar Adhya. 2019. Phage therapy in the twenty-first century: facing the decline of the antibiotic era; is it finally time for the age of the phage? *Annual review of microbiology* 73, 1 (2019), 155–174.

[19] Arian Jamasb, Ramon Viñas Torné, Eric Ma, Yuanqi Du, Charles Harris, Kexin Huang, Dominic Hall, Pietro Lió, and Tom Blundell. 2022. Graphein-a python library for geometric deep learning and network analysis on biomolecular structures and interaction networks. *Advances in Neural Information Processing Systems* 35 (2022), 27153–27167.

[20] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. 2021. Highly accurate protein structure prediction with AlphaFold. *nature* 596, 7873 (2021), 583–589.

[21] Muhammad Kabir, Chanin Nantasenamat, Sakawrat Kanthawong, Phasit Charoenkwan, and Watshara Shoombuatong. 2022. Large-scale comparative review and assessment of computational methods for phage virion proteins identification. *EXCLI journal* 21 (2022), 11.

[22] Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[23] Itziar Lekunberri, Jessica Subirats, Carles M Borrego, and José Luis Balcázar. 2017. Exploring the contribution of bacteriophages to antibiotic resistance. *Environmental Pollution* 220 (2017), 981–984.

[24] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. 2022. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv* 2022 (2022), 500902.

[25] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. 2023. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* 379, 6637 (2023), 1123–1130.

[26] Balachandran Manavalan, Tae H Shin, and Gwang Lee. 2018. PVP-SVM: sequence-based prediction of phage virion proteins using a support vector machine. *Frontiers in microbiology* 9 (2018), 476.

[27] Shawna McCallin, Jessica C Sacher, Jan Zheng, and Benjamin K Chan. 2019. Current state of compassionate phage therapy. *Viruses* 11, 4 (2019), 343.

[28] Yanyuan Pan, Hui Gao, Hao Lin, Zhen Liu, Lixia Tang, and Songtao Li. 2018. Identification of bacteriophage virion proteins using multinomial naive Bayes with g-gap feature tree. *International Journal of Molecular Sciences* 19, 6 (2018), 1779.

[29] Aymen Qabel, Sofiane Ennadir, Giannis Nikolentzos, Johannes F Lutzeyer, Michail Chatzianastasis, Henrik Boström, and Michalis Vazirgiannis. 2022. Structure-Aware Antibiotic Resistance Classification Using Graph Neural Networks. In *NeurIPS 2022 AI for Science: Progress and Promises*.

[30] Rahmatullah Roche, Bernard Moussad, Md Hossain Shuvo, and Debswapna Bhattacharya. 2023. E (3) equivariant graph neural networks for robust and accurate protein-protein interaction site prediction. *PLoS Computational Biology* 19, 8 (2023), e1011435.

[31] Victor Seguritan, Nelson Alves Jr, Michael Arnoult, Amy Raymond, Don Lorimer, Alex B Burgin Jr, Peter Salamon, and Anca M Segall. 2012. Artificial neural networks trained to detect viral and phage structural proteins. *PLoS computational biology* 8, 8 (2012), e1002657.

[32] Jiayu Shang, Cheng Peng, Xubo Tang, and Yanni Sun. 2023. PhaVIP: Phage VIrion Protein classification based on chaos game representation and Vision Transformer. *Bioinformatics* 39, Supplement_1 (2023), i30–i39.

[33] Felix Teufel, Magnús Halldór Gíslason, José Juan Almagro Armenteros, Alexander Rosenberg Johansen, Ole Winther, and Henrik Nielsen. 2023. GraphPart: homology partitioning for biological sequence analysis. *NAR genomics and bioinformatics* 5, 4 (2023), lqad088.

[34] Lin-Fa Wang and Meng Yu. 2004. Epitope identification and discovery using phage display libraries: applications in vaccine development and diagnostics. *Current drug targets* 5, 1 (2004), 1–15.

[35] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[36] Ke Yan, Hongwu Lv, Yichen Guo, Wei Peng, and Bin Liu. 2023. sAMPpred-GAT: prediction of antimicrobial peptide by graph attention network and predicted peptide structure. *Bioinformatics* 39, 1 (2023), btac715.

[37] Qianmu Yuan, Chong Tian, Yidong Song, Peihua Ou, Mingming Zhu, Huiying Zhao, and Yuedong Yang. 2024. GPSFun: geometry-aware protein sequence function predictions with language models. *Nucleic Acids Research* (2024), gkae381.

[38] Yumeng Zhang, Yangming Zhang, Yi Xiong, Hui Wang, Zixin Deng, Jiangning Song, and Hong-Yu Ou. 2022. T4SEfinder: a bioinformatics tool for genome-scale prediction of bacterial type IV secreted effectors using pre-trained protein language model. *Briefings in Bioinformatics* 23, 1 (2022), bbab420.