molecules

MDPI

Article

Hybrid BiLSTM–Graph Neural Networks Learn Efficient Secondary Structure Representations for Protein–Protein Interface Prediction

Niccolò Pancino [1], Fiamma Romagnoli [1], Elia Giuseppe Ceroni [2], Caterina Gallegati [1] and Monica Bianchini [1,*]

[1] Department of Information Engineering and Mathematics (DIISM), University of Siena, 53100 Siena, Italy; niccolo.pancino@unisi.it (N.P.); fiamma.romagnoli@unisi.it (F.R.); c.gallegati@student.unisi.it (C.G.)
[2] National Research Council of Italy (CNR), Institute of Informatics and Telematics (IIT), 56127 Pisa, Italy; elia.ceroni@iit.cnr.it
* Correspondence: monica.bianchini@unisi.it

Abstract

Protein–protein interfaces (PPIs) are essential for numerous biological processes and represent attractive targets for drug discovery and protein engineering. Although recent protein language models have enabled powerful sequence representations for downstream structural prediction tasks, they require substantial computational resources and produce general-purpose embeddings that are not optimized for specific applications. In this work, we investigate whether task-specific representation learning can provide a computationally efficient alternative for PPI prediction. Proteins are represented as graphs of secondary structure elements (SSEs), where each node corresponds to an SSE described by its ordered amino acid sequence and edges encode spatial relationships between neighboring SSEs. We propose an approach based on a hybrid Bidirectional Long Short-Term Memory–Graph Neural Network (BiLSTM-GNN) architecture that learns SSE embeddings directly from residue-level physicochemical descriptors and optimizes them jointly with a downstream graph neural network. The proposed approach is compared with an alternative pipeline based on pretrained ESM2 embeddings while keeping the graph representation and GNN architecture unchanged. Experimental results show that the task-specific BiLSTM encoder achieves competitive predictive performance with respect to the pretrained protein language model, in almost all the considered metrics, in the most challenging evaluation setting while requiring a lower computational effort. These findings suggest that end-to-end task-specific representation learning constitutes an effective and efficient alternative to pretrained protein language models for PPI prediction.

Keywords: protein–protein interfaces; graph neural networks; bidirectional LSTM; protein language models; ESM2; machine learning; secondary structure elements; bioinformatics

Check for updates

Academic Editor: Tigran Chalikian

Received: 20 July 2026
Revised: 4 November 2026
Accepted: 10 November 2026
Published: 13 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Proteins are large biopolymeric molecules that play many functional and structural roles in the wide and diverse realm of living organisms. Protein function is intricately linked to structure, which can be studied and characterized at four distinct levels: primary, secondary, tertiary, and quaternary. The primary structure is the specific sequence of amino acids composing the protein. Local folding patterns of the peptide chain, such as helices,

sheets or coils, constitute the secondary structure. The tertiary structure is the overall three-dimensional conformation of a protein molecule, while the association of multiple protein chains gives rise to the quaternary structure [1]. It is the sequence of amino acids that dictates the ultimate protein structure, thus defining its biochemical functions [2]. Therefore, characterizing the fundamental structural features of proteins is key to understanding how they participate in molecular complexes, mediate biological processes, and interact with ligands [3]. While determining whether two proteins interact is important, identifying the specific residues and structural regions involved in the interaction is often more informative.

Protein–protein interfaces (PPIs; throughout this work, the acronym PPI refers to protein–protein interfaces rather than protein–protein interactions) are the regions of physical contact between interacting proteins. PPIs correspond to the set of amino acids that establish intermolecular interactions between binding partners, thereby determining the specificity, stability, and regulation of the interaction [4]. These regions frequently mediate biological processes such as signal transduction, enzymatic regulation, immune recognition, and macromolecular assembly. Moreover, interface residues are often enriched in disease-associated mutations and constitute attractive targets for therapeutic intervention [5]. Therefore, accurate identification of PPIs is essential not only for understanding the molecular mechanisms underlying protein function, but also for supporting drug discovery and protein engineering applications [6,7].

Since traditional experimental methods for interface identification are often costly and time-consuming, computational approaches have become increasingly attractive alternatives [8]. The advent of deep learning models for protein structure prediction, most notably AlphaFold, has substantially improved our ability to infer protein three-dimensional conformations from sequence information alone. However, accurately predicting protein interactions and identifying interaction interfaces remains a considerably more challenging task. While structural prediction methods provide highly reliable monomeric structures, they do not directly solve the problem of determining which regions of a protein participate in intermolecular interactions [9]. Consequently, dedicated computational approaches for interface prediction are still required. The increasing availability of experimentally resolved protein complexes and interface annotations has fueled the development of computational approaches for PPI prediction.

Protein language models trained on millions of amino acid sequences have emerged as powerful tools for extracting biologically meaningful representations directly from sequence data. Models such as ESM2 generate embeddings that capture evolutionary, structural and functional information without requiring explicit feature engineering. These representations have been successfully applied to a variety of downstream tasks, including structure prediction, function annotation and interaction prediction [10]. Proteins and molecules in general can also be naturally represented as graphs, either with resolution at the single amino acid or at the secondary structure element (SSE) level. Graph neural networks (GNNs) can leverage this data structure to predict PPIs, exploiting well-known approaches based on message-passing GNNs [3,11], graph convolutional networks [12] and graph attention networks [13]. GNNs have already shown promising results in drug discovery applications. Here, the in silico identification of interfaces could be crucial for identifying binding sites. Understanding how and where molecules bind can facilitate the design of more effective and selective therapeutic compounds while reducing off-target effects [14–17].

Accurate identification of interface regions within protein complexes is a fundamental step toward understanding the mechanisms governing protein–protein recognition and binding. In our previous work [11], PPI prediction was formulated as a node-level binary classification task, where each node represented an amino acid residue. Feature importance

analysis revealed that descriptors derived from secondary structure were among the most informative for interface prediction. From a structural perspective, interface residues rarely act in isolation. Instead, they are often organized within larger secondary structure motifs, such as α-helices and β-sheets, which collectively contribute to the formation and stabilization of the interaction surface. Residues belonging to the same α-helix or β-sheet often act cooperatively and share structural and physicochemical properties [18,19]. Consequently, representing proteins at the SSE level may provide a more compact and biologically meaningful description while preserving important information relevant to molecular recognition and binding. Moreover, since each SSE can be naturally described as an ordered sequence of residues, proteins can be represented as graphs of sequences, where sequence information and structural relationships are jointly modeled. For this reason, this work investigates a graph-of-sequences representation in which proteins are modeled as graphs of SSEs, with the objective of improving PPI prediction. In the proposed framework, each node corresponds to an SSE represented as an ordered sequence of residues, while edges describe spatial relationships between SSEs.

As previously mentioned, protein language models, such as ESM2, provide powerful residue representations that have become increasingly popular for downstream structural prediction tasks. However, these models require substantial computational resources for inference and generate general-purpose embeddings that are not optimized for any specific prediction objective. In contrast, task-specific representation learning offers the possibility of learning compact embeddings directly optimized for the downstream problem while significantly reducing computational requirements.

Building upon our previous observation that secondary structure information plays a central role in PPI prediction, we investigate whether explicitly modeling SSEs as residue sequences leads to more informative graph representations. To this end, we propose a hybrid Bidirectional Long Short-Term Memory (BiLSTM)-GNN architecture and compare its learned task-specific embeddings with embeddings obtained from a lightweight pre-trained protein language model (ESM2-T6-8M), while keeping the downstream graph classifier unchanged. The results show that the proposed task-specific encoder achieves competitive predictive performance compared with ESM2-T6-8M while requiring lower computational effort.

In the machine learning literature, supervised node-level classification tasks are conventionally referred to as *prediction*. However, from a structural perspective, a distinction can be drawn between interface annotation and genuine interface prediction. When multi-chain complexes are provided as input (*Whole* and *Interface* configurations), spatial contacts are geometrically present in the input topology, meaning the model performs structural interface annotation or labeling. Conversely, when isolated single chains are evaluated in the absence of partner information (*Chain* configuration), the task constitutes genuine interface location prediction. Throughout this manuscript, the term 'prediction' is used in its broader machine learning sense, with the *Chain* dataset serving as our primary benchmark for genuine interface prediction.

The rest of this paper is organized as follows. Section 2 reports and discusses the experimental results. Section 3 describes the dataset used, the representation learning strategies along with the models employed, and the experimental setup used in the study. Finally, conclusions are drawn along with a mention of future perspectives in Section 4.

2. Results and Discussion

In this section, we evaluate the hybrid BiLSTM-GNN framework for protein–protein interface prediction. In Section 2.1, performance is first assessed across three dataset configurations providing different levels of structural context: *Whole* (fully interconnected

structures), *Interface* (coupled interacting chain pairs), and *Chain* (isolated single chains, serving as the most challenging setting). In Section 2.2, the task-specific BiLSTM encoder is compared to the pretrained ESM2-T6-8M model on the *Chain* dataset to isolate the quality of the learned node representations. Finally, a computational efficiency assessment is presented in Section 2.3, highlighting the resource trade-offs between dynamic end-to-end training and offline feature extraction. Together, these results show that merging structural relationships with sequence-level features delivers a robust, scalable, and highly efficient solution for interface identification.

2.1. Performance of the BiLSTM-GNN Model

Before comparing task-specific and pretrained representation learning strategies, we first evaluate the proposed BiLSTM-GNN architecture under different levels of structural context. The performance of the best architecture on each dataset is reported in Table 1. These results demonstrate that effective PPI predictors can be developed by combining the relational information captured by the SSE-based graph representation with the sequential information encoded in the residue sequences of each SSE, while characterizing both SSEs and residues through their physicochemical properties.

Table 1. The performance of the best architecture is reported by means of the evaluation metrics defined in Section 3.7. For each dataset configuration, values refer to the evaluation on a test set of the best configuration resulting from the grid-search procedure. Acc and BAcc are accuracy and balanced accuracy.

Dataset	Acc	BAcc	Precision	Recall	F-Score	AUC
Whole	0.8546	0.8016	0.7637	0.6837	0.7215	0.8964
Interface	0.7988	0.5941	0.3994	0.2778	0.3277	0.6639
Chain	0.7963	0.7158	0.6704	0.5334	0.5941	0.7904

Overall, the best model configurations obtain good results on all datasets. The metrics show high values for the *Whole* dataset, suggesting that the hybrid model is capable of correctly identifying the class of more than 85% of the nodes. In detail, this dataset configuration achieves the highest F-score, balanced accuracy, and AUC values: 0.7215, 0.8016, and 0.8964, respectively. Indeed, good precision and recall values are obtained as well. Moving from the inner quaternary structure interactions described by the *Whole* dataset to the *Chain* dataset, composed only of single chains, a significant drop in model performance can be observed. In this case, the F-score, balanced accuracy, and AUC values are 0.5941, 0.7158, and 0.7904, respectively. This is consistent with the fact that the model is provided with much less context in terms of the surrounding environment, which proves to be essential for determining whether the SSE under examination is part of a PPI. However, the largest performance drop is observed for the *Interface* dataset, suggesting that pairwise interface information alone is insufficient for the model to reliably determine whether an SSE participates in a protein–protein interaction. In fact, for this dataset, the F-score, balanced accuracy and AUC values are 0.3277, 0.5941 and 0.6639, respectively. This may be due to the graph-of-sequence data domain, which prevents information from flowing between intra-sequence elements, i.e., when generating a sequence embedding, no residues of other SSEs, which may be connected with any kind of relation, are considered. The results also show that models equipped with self-attention mechanism performed better than their counterparts without attention. This is probably due to the improved contextualization capabilities of the attention-equipped models, which can capture the subtle physical and chemical interactions and dependencies between each secondary structure residue in a more effective way.

We also analyzed where the model succeeds and fails by breaking down the test predictions according to the type of SSE. In this context, SSE states are mapped into three broad structural classes: α-helices (H, G, I), β-sheets (E, B), and coils/irregular regions (T, S, -). The classes differ markedly in size and in the proportion of SSEs labeled as part of the interface: coil accounts for 69% of the SSEs, but only 25% of them are labeled as part of the interface, whereas this proportion is 39.5% for α-helices (14% of the SSEs) and 29.6% for β-sheets (17% of the SSEs), against 27.7% overall (Table 2). Overall, the proportion of SSEs predicted as part of the interface (28.0%) matches the labeled one, so the differences between classes reflect how positive predictions are distributed across SSE types rather than a global bias.

Table 2. Distribution of SSE classes and observed versus predicted interface prevalence.

SSE Class	Share of All SSEs (%)	Labeled as Interface Within Class (%)	Predicted as Interface Within Class (%)
α-Helix	14	39.5	47.6
β-Sheet	17	29.6	38.1
Coil/other	69	25.0	21.6
Overall	100	27.7	28.0

The model follows the class-dependent prevalence: it is liberal on α-helices (47.6% predicted as part of the interface, recall 0.71, precision 0.59) and, to a lesser extent, on β-sheets (38.1%, recall 0.72, precision 0.56), and conservative on coil (21.6%, recall 0.48, precision 0.56). The F-score is highest for α-helices (0.64), but this largely reflects their high proportion of positive SSEs and the liberal predictions. Balanced accuracy and AUC, which are less affected by class imbalance, identify β-sheets as the best-predicted class (balanced accuracy 0.74, AUC 0.82), while α-helices (0.69 and 0.76) and coil (0.68 and 0.77) obtain lower values. A plausible explanation is that the position of a strand within a sheet (edge or inner) is reflected in the number of neighboring SSEs, which is precisely the information encoded by the graph. Coil elements account for 62% of the SSEs labeled as part of the interface and, consequently, for 75% of the false negatives. Their recall is the lowest (0.48), while precision (0.56) is in line with that of the other classes. Turns, bends and irregular regions are short, exposed and flexible, lack a regular geometry, and their label often hinges on one or two residues; moreover, they may contribute to binding through induced fit, which cannot be anticipated without information on the partner. The issue seems to concern calibration rather than ranking: the AUC within coil (0.77) is slightly higher than that of α-helices (0.76), and 63% of the false negatives have a score between 0.30 and 0.50, just below the single threshold applied to all classes (Table 3).

Table 3. Interface-prediction performance by SSE class at the common classification threshold.

SSE Class	Precision	Recall	F-Score	BAcc	AUC
α-Helix	0.59	0.71	0.64	0.69	0.76
β-Sheet	0.56	0.72	0.63	0.74	0.82
Coil/other	0.56	0.48	0.52	0.68	0.77

As a post hoc diagnostic, lowering the threshold to about 0.37 for coil raises its recall to 0.68 and its F-score to 0.54, at the cost of a lower precision (0.45), suggesting that SSE-type-aware calibration could further improve recall, which is the most desirable property in the presence of class imbalance.

We also analyzed the prevalence of positive SSEs and the performance of the model as a function of segment length (Table 4). Since an SSE is labeled as part of the interface

as soon as one of its residues is (Section 3.4), the proportion of labeled SSEs increases monotonically with length, and the proportion predicted as part of the interface follows the same trend, so that recall and F-score increase with length. Accuracy, which depends on prevalence, is highest for the shortest SSEs, where most SSEs are negative, and lower for the longest ones. Balanced accuracy and AUC, which are less affected by prevalence, vary much less across length ranges, indicating that the ability of the model to rank SSEs does not depend strongly on their length; balanced accuracy decreases only for the longest SSEs, where positives are over-predicted. Length alone is a weak predictor of the label (AUC 0.60, against 0.78 for the model), and it accounts for part of the differences in prevalence between classes, which follow the order of their mean lengths (Table 2).

Table 4. Prevalence of positive SSEs and performance on the test set as a function of SSE length, in number of residues. SSEs (%) is the percentage of SSEs in each length range; Label (%) and Pred (%) are the percentages of SSEs in the range that are labeled and predicted as part of the interface, respectively, irrespective of whether the prediction is correct. Acc, BAcc, Prec and Rec are accuracy, balanced accuracy, precision and recall. Metrics are computed at the common classification threshold; the last row refers to all SSEs.

Length	SSEs (%)	Label (%)	Pred (%)	Acc	BAcc	Prec	Rec	F-Score	AUC
1	35.5	20.1	14.9	0.82	0.67	0.56	0.42	0.48	0.79
2	27.6	26.9	23.3	0.76	0.67	0.55	0.48	0.51	0.75
3	11.8	31.2	33.4	0.73	0.69	0.56	0.60	0.58	0.76
4–5	10.5	31.3	37.8	0.72	0.70	0.54	0.65	0.59	0.78
6–9	8.3	37.3	50.6	0.68	0.69	0.55	0.75	0.63	0.77
10–19	5.6	47.5	63.3	0.67	0.67	0.61	0.82	0.70	0.76
≥20	0.8	59.9	77.0	0.69	0.64	0.69	0.88	0.77	0.75
all SSEs	100.0	27.7	28	0.76	0.70	0.56	0.57	0.57	0.78

Finally, two aspects must be considered when interpreting the absolute values. First, because of the labeling rule, long SSEs are more likely to be positive than short ones, and part of the differences in prevalence between classes reflects segment length rather than an intrinsic propensity. Second, in the *Chain* dataset a negative label only means that the SSE does not contact any other chain of the resolved complex; since proteins can bind multiple partners, part of the apparent false positives (44% of the SSEs predicted as part of the interface) may correspond to genuine binding regions for partners not annotated in the structure, so precision should be regarded as a conservative estimate.

2.2. Comparison with Pretrained ESM2 Embeddings

Having established that the proposed graph-of-sequences representation effectively captures interface information, we next investigate whether task-specific sequence embeddings are competitive with pretrained protein language model representations. The objective of this study is not to replace pretrained protein language models, but to determine whether, for the specific task of PPI prediction, task-specific representation learning can provide equally informative SSE embeddings with a lower computational cost. The proposed BiLSTM-GNN architecture was compared with an alternative pipeline in which SSE representations were obtained from the pretrained protein language model ESM2. Since the objective of this comparison was to evaluate the quality of the learned node representations independently of the amount of structural context available, both models were evaluated on the *Chain* dataset, which represents the most challenging experimental setting. The results are reported in Table 5.

Table 5. Comparison in performance between our hybrid BiLSTM-GNN model and ESM2 + GNN with the same hyperparameter setting, on average. Average score and standard deviation over five runs are reported.

Metric	BiLSTM	ESM2
Accuracy	0.7628 ± 0.0420	0.7207 ± 0.0280
Balanced Accuracy	0.6904 ± 0.0199	0.6741 ± 0.0121
Precision	0.6029 ± 0.0834	0.5200 ± 0.0785
Recall	0.5265 ± 0.0395	0.5691 ± 0.1013
F-Score	0.5557 ± 0.0273	0.5283 ± 0.0257
AUC	0.7776 ± 0.0314	0.7550 ± 0.0123

The proposed hybrid BiLSTM-GNN architecture is competitive with the pretrained ESM2 baseline across almost all evaluation metrics, achieving higher accuracy (0.7628 vs. 0.7207), balanced accuracy (0.6904 vs. 0.6741), precision (0.6029 vs. 0.5200), F-score (0.5557 vs. 0.5283), and AUC (0.7776 vs. 0.7550). On the other hand, the ESM2-based model obtains a slightly higher recall (0.5691 vs. 0.5265).

The higher precision achieved by the BiLSTM encoder indicates that end-to-end task-specific representation learning produces more reliable positive assignments, resulting in fewer false positive interface predictions. Conversely, while ESM2 recovers a slightly larger fraction of interface SSEs, its lower precision implies a higher rate of false positives. Consequently, the task-specific embeddings lead to an overall better trade-off between precision and recall, as demonstrated by the higher F-score and AUC values. These observations suggest that task-specific sequence representations learned jointly with the downstream GNN are competitive with embeddings extracted from a pretrained protein language model. While ESM2 provides general-purpose representations learned from large-scale protein sequence collections, the BiLSTM encoder is optimized directly for the interface prediction task, allowing it to capture sequence patterns that are specifically relevant for discriminating interface from non-interface SSEs.

When considered together with the computational assessment presented in the following section, these results demonstrate that the proposed BiLSTM-GNN architecture offers an attractive alternative to pretrained protein language models. Despite relying on a considerably lighter sequence encoder, it achieves comparable predictive performance while requiring fewer computational resources, making it well suited for large-scale analyses and computationally demanding applications.

2.3. Computational Efficiency

To assess the practical viability of the proposed framework, the computational requirements of the end-to-end task-specific BiLSTM-GNN architecture were compared to the pretrained ESM2-based pipeline, as shown in Table 6. The computation time shows an operational advantage for the hybrid BiLSTM-GNN model, which was able to complete the training and test procedures in about 36 hours. It is therefore used as a relative baseline unit of measurement at $1\times$. In contrast, the alternative pipeline relying on the pretrained protein language model requires approximately $1.4\times$ the computational effort, for 50 h in total. This difference in resource consumption is particularly meaningful given that the experimental setup intentionally featured a lightweight language model variant. It is worth noting that the recorded relative computation times represent a best-case, minimum-use scenario for the compared frameworks. Because the ESM2 model was executed only once per sequence to generate static node features, it did not contribute any computational overhead to the active training epochs. Instead, the hybrid BiLSTM-GNN model functions entirely end-to-end, requiring the sequence encoder to be dynamically re-evaluated and

updated for every single batch and epoch throughout the entire optimization process. Thanks to the end-to-end training procedure, used to learn the secondary structure element representations directly from residue-level physicochemical descriptors, the hybrid BiLSTM-GNN model altogether avoids the demanding resource and memory requirements typically associated with large-scale Transformer forward passes.

Table 6. Comparison between the hybrid BiLSTM-GNN and ESM2 + GNN models.

Model	Embedding Generation	Computation Time (h)	Relative Run Time
BiLSTM-GNN	Learned end-to-end	36	1×
ESM2 + GNN	Pretrained ESM2	50	approx. 1.4×

3. Materials and Methods

This section presents the datasets and experimental methodology adopted in this study. An overview of the experimental design is provided in Section 3.1, followed by a detailed description of the datasets in Section 3.2. The adopted data representation strategy is introduced in Sections 3.3 and 3.4, whereas the implementation of the proposed model and the selected hyperparameters are described in Sections 3.5 and 3.6, respectively. Finally, Section 3.7 presents the evaluation metrics used to assess model performance.

3.1. Experimental Design

PPI prediction was formulated as a node-focused binary classification task in a graph-of-sequences domain. The BiLSTM-GNN hybrid model was evaluated on three dataset configurations, namely *Whole*, *Interface*, and *Chain*, in order to assess the effect of different levels of structural context on interface prediction. To maintain consistency across training, validation, and test sets in the three cases, a seed was employed to guarantee no overlap between the sets. In this way, the same proteins were associated with the same set in all three settings. The direct comparison with the ESM2-based strategy was performed on the *Chain* dataset only, which provided the least explicit information about the interacting partner and therefore represented the most stringent setting for evaluating the informativeness of SSE embeddings. In this setting, performance depended primarily on the quality of the node representations rather than on the availability of full inter-chain structural context. To ensure a fair comparison, both approaches operated on the same graph-of-sequences representation and used the same downstream GNN architecture, thus differing only in the procedure adopted to generate SSE embeddings.

3.2. Dataset Description

The datasets used are revised versions of those constructed in [11], which proposed three alternative residue-based datasets: Whole, Interface, and Chain. The *Whole* dataset consists of undirected graphs representing the whole three-dimensional, residue-based structures of proteins. In this case, all the chains are interconnected to let the information flow along them, possibly giving insights into their overall structure, function, and behavior within biological systems. In the *Interface* dataset, each protein is treated as a collection of coupled chains and each pair is considered a single sample, allowing the analysis to focus on such critical interaction sites. Consequently, multiple instances of the same chain can occur within a protein if it interacts with two or more other chains. In the *Chain* dataset, each sample is represented by a single chain, so for any given protein only one instance of each chain can occur in the produced graphs. This provides insight into the individual properties and behavior of each chain within the protein complex, allowing for a deeper understanding of how different protein components come together to form functional quaternary structures. The core idea is to use the available data on monomer interactions

within a quaternary structure to train a GNN to predict if a node, be it an amino acid residue or an SSE, is part of the interface region in a protein that the model has never seen. In the current study, a revised version of the three datasets is proposed. In particular, the residue-level representation is replaced by a coarser one, where each node represents an SSE associated with an ordered sequence of residues. As a result, the final number of nodes within each graph is reduced, while the total number of residues represented within the SSE sequences remains unchanged with respect to [11]. The three datasets represent three evaluation settings: how well can the model classify interface-associated SSEs with full complex context? How well can it classify them with the selected chain-pair context? How well can it infer interface-associated SSEs from a single chain without explicit partner-chain geometry? These questions are answered by the *Whole*, *Interface*, and *Chain* datasets, respectively. More details about the resulting datasets are reported in Table 7. Data are therefore represented in a non-Euclidean graph-of-sequences space, where each node corresponds to an SSE internally described by an ordered sequence of residues, while edges encode structural relationships between SSEs.

Table 7. Composition of the SSE-based datasets *Whole*, *Interface*, and *Chain*. Columns N_m, N_M, E_m, E_M, and % SSE_i refer to the minimum and maximum number of nodes and edges belonging to a single sample, and to the percentage of interacting SSEs, respectively. Multiple instances of the same protein chain may appear in the *Interface* dataset; consequently, the total number of nodes exceeds that of the *Whole* dataset.

Dataset	Samples	N_m	N_M	E_m	E_M	% SSE_i
Whole	31,979	4	3083	10	24,912	27.60
Interface	73,426	2	1880	2	12,082	17.85
Chain	87,375	2	944	2	6082	27.82

3.3. Secondary Structure Representations

In our graph representation, each node corresponds to an SSE containing an ordered sequence of amino acid residues. Secondary structure assignments are determined at the residue level using the DSSP (Define Secondary Structure of Proteins) algorithm, which uniquely assigns each individual amino acid residue within a protein structure to a specific secondary structure state. Contiguous stretches of amino acid residues sharing the exact same assigned state are subsequently grouped together to form individual SSE nodes, preserving the ordered linear sequence of residues within each element. Additional information for each SSE is reported in Table 8.

Grouping residues into individual segments defines the SSEs represented by the graph nodes. This structural description is distinct from the numerical encoding described below. SSEs contain different numbers of residues, whereas the downstream GNN requires each node to be described by the same number of numerical features within a given model. We therefore convert the residue information associated with each SSE into a fixed-length numerical vector, termed an SSE embedding. For example, a short segment and a long segment are represented by vectors with the same number of components, although the values of those components differ. This allows the GNN to process segments of different lengths using a common input format. The conversion from residue information to an embedding is referred to as encoding. An embedding provides a compact, learned representation of a segment; its individual components do not necessarily correspond to directly interpretable properties such as charge or hydrophobicity. The GNN uses these representations together with the spatial relationships between SSEs to classify each node as interfacial or non-interfacial.

Table 8. Length of the SSEs, expressed as number of residues, for each of the eight DSSP states, for the three classes obtained by grouping them (α-helices: H, G, I; β-sheets: E, B; random coil: T, S, -) and for all SSEs. *N* and % are the number of SSEs of each type and their percentage of SSEs over the total; min, max, mean, median, std and var are the minimum, maximum, mean, median, standard deviation and variance of their length. % is the percentage of SSEs of each type over the total number of SSEs in the dataset.

Class	ID	*N*	%	Min	Max	Mean	Median	Std	Var
α-helix	H	715,959	9.9	1	155	11.04	10	6.19	38.27
3_{10}-helix	G	274,650	3.8	1	14	3.36	3	0.88	0.77
π-helix	I	27,508	0.4	2	15	5.42	5	1.00	1.01
extended strand	E	920,374	12.7	1	94	5.30	5	2.64	6.95
β-bridge	B	280,443	3.9	1	3	1.02	1	0.15	0.02
H-bonded turn	T	1,273,310	17.6	1	15	2.07	2	0.83	0.69
bend	S	1,298,917	17.9	1	11	1.54	1	0.82	0.67
coil	-	2,449,680	33.8	1	35	1.90	1	1.34	1.79
α-helices	H, G, I	1,018,117	14.1	1	155	8.81	7	6.24	38.97
β-sheets	E, B	1,200,817	16.6	1	94	4.30	4	2.93	8.61
random coil	T, S, -	5,021,907	69.4	1	35	1.85	2	1.12	1.26
all SSEs		7,240,841	100.0	1	155	3.23	2	3.70	13.68

We compare two ways of obtaining these embeddings. In the first, a BiLSTM encoder learns to represent SSEs specifically for node classification and is trained jointly with the GNN. In the second, a pretrained protein language model, ESM2, supplies residue representations that are combined into SSE embeddings and remain fixed during GNN training. Thus, the comparison examines task-specific representation learning against the use of pretrained representations.

3.3.1. Task-Specific BiLSTM Encoding

In the task-specific approach, each residue is described by numerical features, including accessible surface area, torsion angles, and Meiler and ExPASy descriptors (the full list of residue-level features can be found in [11]). Within each SSE, these descriptors are arranged in residue sequence order and supplied to a bidirectional long short-term memory network (BiLSTM). A BiLSTM processes a sequence in both directions, allowing its representation to incorporate information from residues preceding and following a given position within the segment. The encoder converts each variable-length sequence of residue descriptors into a fixed-length SSE embedding. The embedding dimensionality is a model hyperparameter, with values of 8, 16 and 32 evaluated in this study (Table 9). The implementation uses a leaky ReLU activation function. The SSE embedding is combined with a numerical indicator of the segment's structural class and passed to the GNN, as illustrated in Figure 1. The GNN then uses the node representations and their graph connections to predict interface membership. The BiLSTM and GNN are trained jointly. Errors in the final interface predictions are used to update both the graph classifier and the sequence encoder. Consequently, the SSE embeddings change during training, allowing the model to learn representations that are useful for the specific classification task. Because segment lengths range from 1 to 155 residues, we also evaluated the inclusion of self-attention. This mechanism allows the encoder to combine information from different positions within an SSE and adjust their contributions to the representation. Its inclusion was treated as a hyperparameter during model selection. Attention weights describe how information is used by the model and should not, by themselves, be interpreted as evidence of the biological importance of individual residues.

Table 9. Hyperparameter search space of the hybrid BiLSTM-GNN model for each dataset configuration, with the corresponding best configuration. The parameter w_{bal} refers to the proportion of balancing weight assigned to the minority class at training time. Indeed, $w_{bal} = 0$ means no balancing policy was used, while $w_{bal} = 1$ means the classes were perfectly balanced. The parameter h refers to hidden layer units in the state and output networks, i.e., net_s and net_o. Note that a value of $h = 0$ means no hidden layers were used in the corresponding architecture.

Hyperparameter	Grid Search	Best Configuration		
		Whole	*Interface*	*Chain*
Sequence embedding	8, 16, 32	8	16	32
Attention mechanism	no, yes	yes	yes	yes
State dimension	16, 32, 64	32	32	16
Max iterations	3, 4, 5	5	5	5
w_{bal}	0, 0.2, 0.5, 1	1	0.5	0.5
net_s units h	0, 256, 512	0	256	0
net_o units h	0, 256	0	0	0

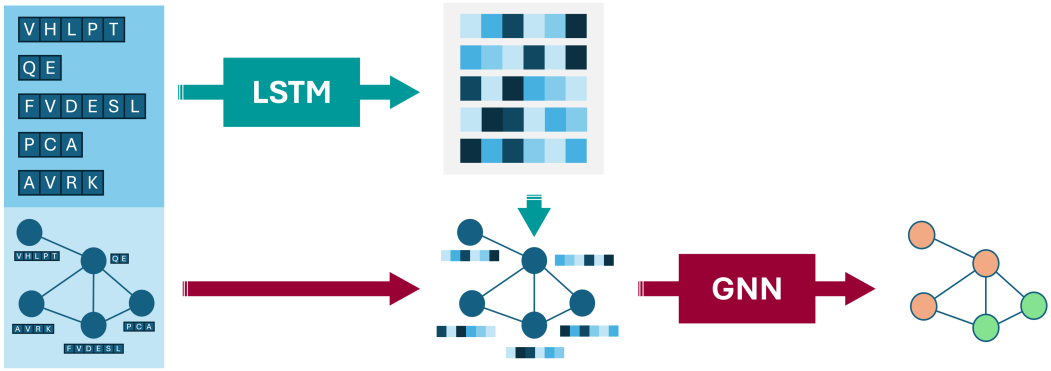

Figure 1. Visual representation of the hybrid model. The BiLSTM model creates a sequence embedding for each SSE node. The sequence embedding is then concatenated to the pre-existing node features, in the form of a one-hot representation of the SSE class. The resulting graph is processed by a GNN module, which predicts whether each SSE node within a protein is part of the protein–protein interface or not. On the right, an example of node-per-node prediction: negative prediction in red ("SSE is not part of the interface"), positive prediction in green ("SSE is part of the interface"). An SSE is interacting if at least one residue belonging to its sequence is part of the reference interface.

3.3.2. Pretrained ESM2 Encoding

The alternative approach uses ESM2, a protein language model pretrained on amino acid sequences, to generate residue representations before training the GNN. We selected ESM2-T6-8M (facebook/esm2_t6_8M_UR50D), which contains six Transformer layers and approximately eight million parameters, as a lightweight pretrained baseline consistent with the computational-efficiency objective of this study. The comparison therefore concerns this particular ESM2 configuration rather than protein language models in general. Each complete protein-chain sequence is supplied to ESM2, which produces a vector of 320 numerical features for every residue. These vectors are contextual: the representation of a residue depends on its surrounding sequence, rather than only on its amino acid identity. Residue vectors are extracted from the final hidden layer after removing the special beginning- and end-of-sequence tokens. To obtain an SSE embedding, we average

the vectors of the residues belonging to that segment. This operation, termed mean pooling, produces one 320-dimensional vector regardless of how many residues the SSE contains:

$$\mathbf{e}_{\text{SSE}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{e}_i, \qquad \mathbf{e}_i \in \mathbb{R}^{320}, \tag{1}$$

where N is the number of residues in the SSE and $\mathbf{e}_i$ is the ESM2 representation of its (i)-th residue. The resulting vector summarizes the segment using residue representations generated in the context of the complete chain sequence. ESM2 is used as a fixed feature extractor: its parameters are not updated for interface classification, and the SSE embeddings are computed once before GNN training. Only the downstream GNN is subsequently trained. The central distinction between the two pipelines is therefore how their node representations are obtained and adapted. The BiLSTM learns SSE embeddings through the interface-prediction objective, whereas ESM2 provides pretrained representations that remain unchanged during downstream learning. Both approaches supply a fixed-length vector for each SSE, although their vector dimensions differ, and use the same graph topology, reference labels and downstream GNN architecture.

3.4. Graph Construction

The embeddings generated according to the representation learning strategies described in the previous section are used to construct graph-of-sequences representations of proteins. In the proposed framework, each SSE is represented as a graph node, while edges encode the spatial relationships between pairs of SSEs. The three-dimensional coordinates of each SSE were obtained from the corresponding protein structure by considering the geometric center (centroid) of the residues belonging to the SSE. The graph construction procedure is identical for both experimental pipelines. Consequently, the graph topology, edge set, and node labels are shared across all experiments, while only the node representations differ according to the selected embedding strategy. This design enables a direct comparison between task-specific and pretrained SSE representations while keeping the downstream graph learning architecture constant.

The resulting graphs are therefore defined as

$$G = (V, E), \tag{2}$$

where each node $v \in V$ corresponds to an SSE described by its learned representation, and each edge $e \in E$ represents a spatial adjacency between two SSEs. Each node in the graph is associated with one of two classes, namely interacting and non-interacting. While in [11] individual residues were directly associated with the PPI, in the SSE-based representation an SSE is considered interacting if at least one residue belonging to its sequence is part of the interface; otherwise, it is labeled as non-interacting.

3.5. Shared Downstream GNN Architecture

The downstream GNN performs node-level binary classification by iteratively propagating information between neighboring SSEs through message passing. The learned node states are then mapped to interface/non-interface predictions. The implementation of all the models was conducted using GNNKeras, a Keras-based library for GNNs and homogeneous and heterogeneous graph processing [20]. Regarding the implementation details, the models were trained for up to 100 epochs using the cross-entropy loss function and the Adam optimizer with an initial learning rate of 0.001. To prevent overfitting, early stopping was applied using the validation F-score.

3.6. Hyperparameter Search

Determining the optimal architecture requires identifying the hyperparameter configuration that yields the best performance for the target problem. For this reason, a grid-search procedure was conducted on the GNN search space. In Table 9 the best architecture for each setting is reported in terms of F-score. The grid search was performed for the hybrid BiLSTM-GNN model only, and the best configuration was then used for ESM2-derived SSE classification.

3.7. Evaluation Metrics

Model performance should be evaluated in light of the class distribution within the datasets. In general, binary classification tasks are measured in terms of accuracy, which is, however, not precise when dealing with unbalanced situations such as the current one, where the negative examples (non-interface) outnumbered the positive ones (interface). Therefore, additional metrics were taken into account, namely balanced accuracy, recall, precision, F-score, and area under the ROC curve (AUC). When compared to the ground truth, the predictions proposed by a generic model can be described in terms of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) instances. In other words, a TP occurs when the model correctly identifies positive samples, while an FP is a negative sample incorrectly labelled as a positive one. TN and FN are defined similarly. This notation is the one employed in the following definition of the evaluation metrics:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

$$Balanced\ Accuracy = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F\text{-}score = \frac{2TP}{2TP + FP + FN} \tag{7}$$

All the evaluation metrics considered in this study take values in the interval $[0, 1]$, with higher values indicating better predictive performance. Accuracy measures the overall proportion of correct predictions, whereas balanced accuracy assigns equal importance to the two classes and is therefore more informative in the presence of class imbalance. Precision quantifies the proportion of predicted interface nodes that are true interface nodes, while recall quantifies the proportion of true interface nodes correctly identified. The F-score summarizes the trade-off between precision and recall through their harmonic mean, making it a suitable metric for problems with unbalanced datasets. For AUC, a value of 0.5 corresponds to random ranking performance, whereas a value of 1 indicates perfect discrimination between interface and non-interface nodes. Similarly, in a binary classification setting, a balanced accuracy of 0.5 generally represents chance-level performance.

Furthermore, given the high ratio of negative to positive examples in the datasets, a multiplier (w_{bal}) was used during training to assign more weight to positive examples, thereby balancing the learning process. When $w_{bal} = 1$, the training set is perfectly balanced, meaning that the total weight of positive examples equals that of negative examples, making each positive example more significant during training. Conversely, when $w_{bal} = 0$, no balancing strategy is applied, and a positive example is considered just as important as a negative one.

3.8. Use Case

To illustrate the practical use of the model, we considered the homodimeric citrate synthase of *Pyrococcus furiosus* (PDB code 1AJ8). It is an unambiguous biological dimer, with a single interface of about 4690 Å^2 and a PDBePISA complexation significance score of 1.0, i.e., an interface essential for the stability of the assembly rather than a crystal-packing contact. Moreover, the protein contains α-helices, β-strands and coil (41, 16 and 122 SSEs, respectively), all including SSEs labeled as part of the interface, and its two chains are identical, so they act as independent replicates for the model; the protein was never used to train the network.

The use case was performed on the *Chain* setting: the model receives the SSE graph of each chain taken in isolation, without any information about the partner, and predicts, for each SSE, whether or not it is part of the interface. Therefore, the task is a genuine prediction of the interface location rather than the annotation of an interface that is already geometrically visible. Figure 2 shows the SSE graph of 1AJ8 (179 SSEs, 94 of which are labeled as part of the interface) and the output of the model.

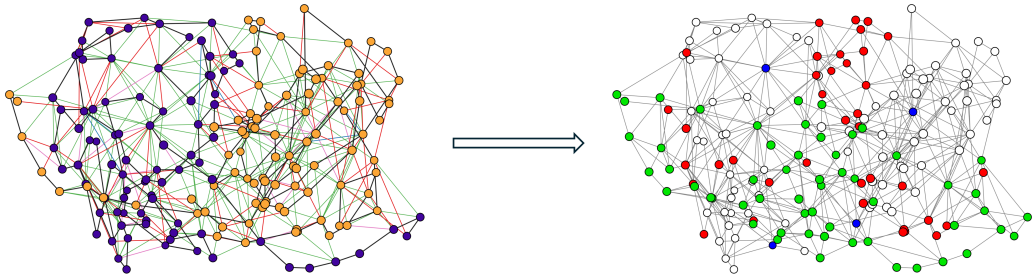

Figure 2. Use case on the citrate synthase of *Pyrococcus furiosus* (PDB code 1AJ8), a homodimer. On the left, the SSE graph of the complex, with nodes colored by chain (A in purple, B in orange) and edges colored by interaction type. On the right, the output of the model: true positive (TP), true negative (TN), false positive (FP) and false negative (FN) predictions are colored in green, white, blue and red, respectively.

The model correctly classifies 137 of the 179 SSEs (accuracy 0.77, balanced accuracy 0.77, AUC 0.91), with 56 true positives, 81 true negatives, only 4 false positives and 38 false negatives. Hence, the model is very reliable when it predicts an SSE as part of the interface (precision 0.93), while it recovers 60% of the SSEs labeled as part of the interface (recall 0.60, F-score 0.73). The behavior mirrors the class-wise analysis of Section 2: the recall is 0.74 for α-helices, 0.75 for β-strands and 0.52 for coil, all the predictions on β-strands and coil are correct, and the four false positives are all α-helices. The two chains obtain similar results (recall 0.62 and 0.57), indicating that the prediction is stable across the two replicates. The false negatives are mostly marginal contacts: the median buried surface area (computed by PDBePISA) of the missed SSEs is 38 Å^2, against 94 Å^2 for the correctly identified ones, and the 56 true positives account for 77% of the total area buried at the interface. Moreover, 82% of the false negatives have a score between 0.30 and 0.50, just below the decision threshold, and the score of the SSEs labeled as part of the interface increases with their buried area (Spearman correlation 0.57). Thus, the model captures the core of the interface and mostly misses peripheral SSEs, in most cases short coil elements (30 of the 38 false negatives), which are the hardest to anticipate without information on the partner. Being a single example, this use case is not meant to represent the average performance, which is reported in Table 1: the AUC on this protein (0.91) is higher than on the whole test set (0.76), probably because of the large and well-defined interface of a stable homodimer.

4. Conclusions

Building upon our previous observation that secondary structure information plays a central role in PPI prediction, this work introduced a graph-of-sequences framework in which proteins are represented as graphs of SSE described by ordered residue sequences. A hybrid BiLSTM-GNN architecture was proposed to learn task-specific SSE representations directly from residue-level physicochemical descriptors and was compared with an alternative pipeline based on embeddings extracted from the lightweight pretrained protein language model ESM2-T6-8M.

The proposed approach achieved its best results on the Whole dataset, where the contacts with the partner chains are present in the input, and retained most of this performance on the Chain dataset, the principal prediction setting.

Furthermore, the comparison with ESM2 showed that learning sequence representations directly for the target task yields comparable predictive performance at a lower computational cost. These findings highlight the potential of lightweight, end-to-end sequence encoders as a practical alternative to pretrained protein language models for specialized applications such as PPI prediction.

Overall, the results demonstrate that integrating task-driven SSE representations into a graph-based framework offers an efficient and scalable solution for PPI prediction.

Future work will investigate alternative sequence encoders and richer graph representations, as well as the application of the proposed framework to larger and more diverse protein interaction datasets. Additionally, while dataset extraction relied on stoichiometric, symmetry-based, and structural geometric criteria via PDBePISA, the extent of potential crystallographic-contact contamination has not been explicitly quantified.

Incorporating systematic assembly-level verification to eliminate potential crystal packing artifacts represents an important direction to further strengthen the biological interpretation and external applicability of the proposed framework.

Author Contributions: Conceptualization: N.P., F.R., E.G.C. and C.G.; data curation: N.P., F.R. and C.G.; formal analysis: F.R. and M.B.; funding acquisition: M.B.; investigation: F.R., E.G.C. and C.G.; methodology: N.P., F.R., E.G.C. and C.G.; project administration: M.B.; software: N.P. and F.R.; supervision: N.P. and M.B.; validation: F.R., E.G.C. and C.G.; visualization: N.P., F.R., E.G.C. and C.G.; writing—original draft: N.P., F.R., E.G.C., C.G. and M.B.; writing—review and editing: N.P., F.R., E.G.C., C.G. and M.B. All authors have read and agreed to the published version of the manuscript.

Funding: Thisstudy was co-funded by the European Union—*NextGenerationEU*, in the context of The National Recovery and Resilience Plan—Investment 1.5 Ecosystems of Innovation, Project Tuscany Health Ecosystem (THE), Spoke 3—Advanced technologies, methods and materials for human health and well-being. ECS00000017, CUP: B63C22000680007. C.G. acknowledges the support provided by the European Union—*NextGenerationEU*—Mission 4, Component 1, Investment 4.1 under the PNRR, as part of the PhD scholarship co-financed by Ministerial Decree no. 118/2023 (CUP: B63C23001110001). The research by N.P. was supported by the EU—*NextGenerationEU*, under the National Recovery and Resilience Plan (NRRP)—Extended Partnership SERICS, Spoke 3 Attacks and Defences, Mission 4, Component 2, Investment 1.3, AI-RESCUE. PE00000014, CUP: B63C24000490006.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PPI	Protein–Protein Interface
PDB	Protein Data Bank
PDBePISA	PDB Europe (Proteins, Interfaces, Structures and Assemblies)
GNN	Graph Neural Network
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional LSTM
DSSP	Define Secondary Structure of Proteins
SSE	Secondary Structure Element
ROC	Receiver Operating Characteristic
AUC	Area Under the ROC Curve
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
ESM	Evolutionary Scale Modeling

References

1. Soleymani, F.; Paquet, E.; Viktor, H.; Michalowski, W.; Spinello, D. Protein–protein interaction prediction with deep learning: A comprehensive review. *Comput. Struct. Biotechnol. J.* **2022**, *20*, 5316–5341.
2. Rehman, I.; Kerndt, C.C.; Botelho, S. Biochemistry, Tertiary Protein Structure. In *StatPearls [Internet]*; StatPearls Publishing: Treasure Island, FL, USA, 2022. Available online: https://www.ncbi.nlm.nih.gov/books/NBK470269/ (accessed on 20 May 2026).
3. Pancino, N.; Rossi, A.; Ciano, G.; Giacomini, G.; Bonechi, S.; Andreini, P.; Scarselli, F.; Bianchini, M.; Bongini, P. Graph Neural Networks for the Prediction of Protein–Protein Interfaces. In Proceedings of the 2020 European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning , Online, 2–4 October 2020; pp. 127–132.
4. Schreiber, G. Protein–protein interaction interfaces and their functional implications. In *Protein–Protein Interaction Regulators*; Royal Society of Chemistry: London, UK, 2020; Chapter 1, pp. 1–24.
5. Livesey, B.J.; Marsh, J.A. The properties of human disease mutations at protein interfaces. *PLoS Comput. Biol.* **2022**, *18*, e1009858.
6. Lee, M. Recent advances in deep learning for protein–protein interaction analysis: A comprehensive review. *Molecules* **2023**, *28*, 5169.
7. Pancino, N. Graph Neural Networks for Advanced Molecular Data Analysis. Ph.D. Thesis, University of Florence: Florence, Italy, 2023.
8. Hu, X.; Feng, C.; Ling, T.; Chen, M. Deep learning frameworks for protein–protein interaction prediction. *Comput. Struct. Biotechnol. J.* **2022**, *20*, 3223–3233.
9. Li, M.Q.C.; Wang, S.; Lin, S.R.; Ting, L.E.N.; Wan, Z.H.; Xie, G.; Zhang, J. Advantages and limitations of AlphaFold in structural biology: Insights from recent studies. *Protein J.* **2026**, *45*, 22–38.
10. Chen, J.Y.; Wang, J.F.; Hu, Y.; Li, X.H.; Qian, Y.R.; Song, C.L. Evaluating the advancements in protein language models for encoding strategies in protein function prediction: A comprehensive review. *Front. Bioeng. Biotechnol.* **2025**, *13*, 1506508.
11. Pancino, N.; Gallegati, C.; Romagnoli, F.; Bongini, P.; Bianchini, M. Protein–Protein Interfaces: A Graph Neural Network Approach. *Int. J. Mol. Sci.* **2024**, *25*, 5870.
12. Réau, M.; Renaud, N.; Xue, L.C.; Bonvin, A.M. DeepRank–GNN: A graph neural network framework to learn patterns in protein–protein interfaces. *Bioinformatics* **2023**, *39*, btac759.
13. Olechnovič, K.; Venclovas, Č. VoroIF–GNN: Voronoi tessellation–derived protein–protein interface assessment using a graph neural network. *Proteins Struct. Funct. Bioinform.* **2023**, *91*, 1879–1888.
14. Bongini, P.; Scarselli, F.; Bianchini, M.; Dimitri, G.M.; Pancino, N.; Liò, P. Modular multi–source prediction of drug side–effects with DruGNN. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2022**, *20*, 1211–1220.
15. Dimitri, G.M.; Lió, P. DrugClust: A machine learning approach for drugs side effects prediction. *Comput. Biol. Chem.* **2017**, *68*, 204–210.
16. Pancino, N.; Perron, Y.; Bongini, P.; Scarselli, F. Drug side effect prediction with deep learning molecular embedding in a graph–of–graphs domain. *Mathematics* **2022**, *10*, 4550.

17. Bongini, P.; Messori, E.; Pancino, N.; Bianchini, M. A Deep Learning Approach to the Prediction of Drug Side–Effects on Molecular Graphs. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2023**, *20*, 3681–3690.
18. Sheridan, R.P.; Lee, R.H.; Peters, N.; Allen, L.C. Hydrogen-bond cooperativity in protein secondary structure. *Biopolym. Orig. Res. Biomol.* **1979**, *18*, 2451–2458.
19. Salinas, V.H.; Ranganathan, R. Coevolution-based inference of amino acid interactions underlying protein function. *eLife* **2018**, *7*, e34300.
20. Pancino, N.; Bongini, P.; Scarselli, F.; Bianchini, M. GNNkeras: A Keras–based library for Graph Neural Networks and homogeneous and heterogeneous graph processing. *softwareX* **2022**, *18*, 101061.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.