Exploring Machine Learning Algorithms and Numerical Representations Strategies to Develop Sequence-Based Predictive Models for Protein Networks

David Medina-Ortiz[1(✉)], Pedro Salinas[2], Gabriel Cabas-Moras[1], Fabio Durán-Verdugo[2], Álvaro Olivera-Nappa[3], and Roberto Uribe-Paredes[1]

[1] Departamento de Ingeniería en Computación, Universidad de Magallanes, Av. Pdte. Manuel Bulnes, 01855 Punta Arenas, Chile
david.medina@umag.cl

[2] Escuela de Ingeniería en Bioinformática, Universidad de Talca, Av. Lircay S/N, Talca, Chile

[3] Departamento de Ingeniería Química, Biotecnología y Materiales, Universidad de Chile, Beauche 851, Santiago, Chile

Abstract. Predicting the affinity between two proteins is one of the most relevant challenges in bioinformatics and one of the most useful for biotechnological and pharmaceutical applications. Current prediction methods use the structural information of the interaction complexes. However, predicting the structure of proteins requires enormous computational costs. Machine learning methods emerge as an alternative to this bioinformatics challenge. There are predictive methods for protein affinity based on structural information. However, for linear information, there are no development guidelines for elaborating predictive models, being necessary to explore several alternatives for processing and developing predictive models. This work explores different options for building predictive protein interaction models via deep learning architectures and classical machine learning algorithms, evaluating numerical representation methods and transformation techniques to represent structural complexes using linear information. Six types of predictive tasks related to the affinity and mutational variant evaluations and their effect on the interaction complex were explored. We show that classical machine learning and convolutional network-based methods perform better than graph convolutional network methods for studying mutational variants. In contrast, graph-based methods perform better on affinity problems or association constants, using only the linear information of the protein sequences. Finally, we show an illustrative use case, expose how to use the developed models, discuss the limitations of the explored methods and comment on future development strategies for improving the studied processes.

Keywords: Protein networks · machine learning algorithms · protein language models · protein discovery · deep learning architectures

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13956, pp. 231–244, 2023.
https://doi.org/10.1007/978-3-031-36805-9_16

1 Introduction

Protein interaction networks allow understanding and identifying relationship patterns between proteins used in biotechnology, bioinformatics, drug discovery, and drug repurposing [8,21,31]. The affinity between proteins and ligands could be improved by combining bioinformatic approaches with protein engineering techniques, like directed evolution or rational design, benefiting pharmaceutical and industrial applications [13,30]. Furthermore, measuring changes in the stability of interaction complexes, evaluating biochemical constants such as dissociation, and recording rates of relatedness between two proteins foster information enrichment and facilitate the development of more complex methodologies benefiting the protein engineering applications [7,30].

Protein interaction network databases provide information on proteins and their interactors. Intact [14] and STRING [25] are two databases that present information on protein-protein interactions based on likelihood estimated using the number of reports in the literature. The use of a likelihood value is an ambiguous method when evaluating the level of affinity between two proteins. In addition, Intact and STRING do not have negative information, i.e., it cannot be assumed that if two proteins are not registered as interacting, they do not present interactions. As alternatives, databases such as PDBbind [27] and Pip [2] provide quantitative values that allow evaluating the interaction affinity between two proteins. Databases such as PROXiMATE [12] and SKEMPI [11] give knowledge on the changes in stability produced by mutations on the interaction complexes, evaluating different biochemical constants and the effects of association and dissociation.

Computational methods have been developed for constructing interaction predictive systems between two proteins using the information reported in biological databases. Classical approaches are based on molecular dynamics to study the affinity between proteins, employing tools like Autodock vina [6], InterEvDock [29], and ZDock [23]. The application of molecular dynamics methods implies a high computational cost and having the three-dimensional structures of the proteins to be evaluated, not applying to a study of latent space navigation and landscape reconstruction. Thanks to the constant growth of data reported in the literature, predictive systems via machine learning techniques have acquired a more significant role in recent years [5].

Commonly, machine learning methods use structural information to train predictive models, applying both interaction complexes [28], 3D-voxel box strategies, and using data from techniques such as molecular docking or dynamics obtained from molecular simulations [1,9]. In general, the methods apply deep learning architectures based on convolutional neural networks (CNN) [10,32], recurrent neural networks (RNN) [20,33], or long short-term memory (LSTM) [15,26].

Constructing interaction affinity predictive systems using only linear information is a current problem, and there are no established solutions or development guides with straightforward methods for elaborating these predictive systems. One of the significant challenges during the training of predictive models via linear information of protein sequences consists of the numerical representation of

protein sequence. Methods such as amino acid encodings based on physicochemical properties have been proposed [17]. In addition, the use of Fourier transforms for spatial transformations has been explored, allowing elaborate representations that emulate the structural behaviors of proteins [3,17,24]. Recently, representation learning methods based on Natural Language Processing (NLP) techniques have been studied, giving birth to different pre-trained models and computational tools such as bio-embedding [4], TAPE [22], or ECNet [16]. However, there is yet to be a clear alternative for employing these methods. On other hands, to represent the complex structure in protein interactions problems is commonly to apply concatenation strategies not being explored transformations and combinations strategies through techniques such as Principal Component Analysis.

This work explores different machine learning strategies for developing predictive models for protein-protein interaction using linear sequences. First, different coding methods are implemented to represent protein sequences numerically. Then, CNN and graph convolutional networks (GCN) architectures are designed for training predictive models, and classical supervised learning algorithms such as Random Forest (RF) and support vector machine (SVM) are trained. Alternatively, post-processing strategies for the linear representation of structural complexes via applying principal component analysis (PCA) or nonlinear combinations are analyzed. Each model is evaluated using metrics such as mean squared error (MSE) and R^2 score. The evaluated strategies are compared to determine common strategies to develop guidelines to build predictive models.

The evaluation of the methodologies was developed by studying six datasets collected from databases such as PROXiMATE or SKEMPI, demonstrating that classical machine learning methods and models via CNN outperform GCN models in problems related to changes in kinetic constants in mutational change evaluation. At the same time, predictive systems based on GCN architectures perform better in affinity problems or association constants between proteins, acquiring better learning of hidden patterns than CNN or classical machine learning methods. Finally, we show a use case of predictive model development via CNN and how the model is used to obtain predictions. Finally, we discuss the presented methods' limitations, future work, and possible development applications.

2 Methods

This section will describe the different methodologies developed to meet the objectives of this work, considering the stages of data collection, data processing, numerical representation strategies, training, and validation of predictive models.

2.1 Collecting and Processing Datasets

The PROXiMATE and SKEMPI databases were used to search affinity values in protein-protein interactions, protein complex with point mutations or mutational variants and the effects caused by mutational changes in terms of kinetic

constants, affinity, and other properties commonly applied in biochemical studies for the study of interactions in protein complexes. All records were downloaded manually using the download options available on each database. Then, the datasets were built using the downloaded records, and processed by a classical cleanup of removal of unreported records and statistical methods to detect outliers in the reported responses and the lengths of the protein sequences, to facilitate the application of numerical representation strategies.

2.2 Numerical Representation Strategies of Protein Sequences

For the numerical representations of protein sequences, three strategies were employed. The first was based on amino acid encodings using physicochemical properties reported in [17]. The second consisted of applying spatial transformations employing FFT following the basis reported in [3,17,24], which indicates that any set represented by codings based on amino acidic properties can be transformed via FFT. As a requirement for the application of FFT, the sequences encoded via physicochemical properties were manipulated so that they all had the same length by applying *zero-padding*, resulting in numerical vectors of maximum size 1024 after the application of FFT. The third strategy was the application of pre-trained protein language models. For this purpose, the glove, one-hot, bepler, and fasttext models, enabled in the bio-embedding tool [4], were used. When applying these models, it is unnecessary to manipulate sequence sizes or use *zero-padding*. However, as a requirement, a maximum length of amino acids per sequence, which consists of 1024, is required, thus agreeing with the application of the methods based on amino acid coding. This work did not explore strategies such as one hot encoding or sequence description, since we wanted to analyze the effect of amino acid encodings v/s representations based on NLP learning models.

2.3 Building Datasets to Train Predictive Models

The following strategy was developed to generate input datasets for predictive model training. First, interaction sequences were concatenated according to literature reports before the numerical representation strategy was applied. Then, transformation methods were implemented to explore the effect of linear and nonlinear combinations of the initial inputs using PCA and kernel-PCA methods, resulting in three matrices for each representation strategy: i) simple concatenation, ii) application of PCA after concatenation, and iii) application of kernel-PCA post concatenation. These matrices were employed as input for training predictive models based on CNN architectures and classical supervised learning algorithms. In the case of datasets for the development of predictive models based on graph architectures, datasets were constructed to represent interactions and edge weights (this value being the model response and the desired value to predict). Matrices with the unique protein sequences existing in the dataset are numerically represented by all the numerical representation strategies explored in this work.

2.4 Implementation Strategies of Preprocessing and Predictive Models and Execution Process

The data preprocessing strategies were implemented using Python v3.8 programming language and employing classic dataset management modules such as Pandas or Numpy. In the case of predictive model training, the scikit-learn library was used for the application of classical machine learning algorithms. The predictive models built via convolutional neural network architectures were implemented using the TensorFlow framework. For the predictive models based on graph convolutional network architectures, the Torch framework was used. The optimization of the training and data processing runs was done using GPUs and configuring the libraries for their use. A machine processed all model training runs with Linux operating system, Arch distribution, with the following characteristics: i) Processor: AMD Ryzen 7 6800H, 3.2 GHz up to 4.7 GHz, ii) Core numbers: Octa Core, iii) Memory: 16 GB DDR5-SDRAM 4800 MHz (2×8 GB) 2 Slot, iv) Storage: 512 GB SSD NVMe, PCI Express 4.0 M.2, Graphic card: NVIDIA GeForce RTX 3060 (6 GB GDDR6). A conda environment with all the requirements is enabled to facilitate the replication of the experiments and the evaluation of the generated implementation. The source code is available for non-commercial use, licensed under the GPL-3.0 license at the following URL https://github.com/ProteinEngineering-PESB2/protein-interactions-models to facilitate the replication of the experiments and the evaluation of the generated implementation.

3 Results and Discussions

This section describes the main results obtained and discusses the methodologies evaluated, contemplating both the evaluation of the machine learning algorithms and the strategies for the numerical representation of protein sequences assessed in this work. Finally, a use case is presented to demonstrate how to apply the methodologies for developing predictive models of interaction affinity in protein complexes.

3.1 Exploring Datasets and Numerical Representation Strategies

A total of six protein interaction datasets were built in this work (See Table 1). Three of the datasets contain protein-protein affinity information, and three have information on mutational changes and their effect on interaction complexes estimated in responses such as free energy difference (PROXiMATE-dg dataset) and kinetic constants (PROXiMATE-kd and SKEMPI-koff, PROXiMATE-kon and SKEMPI-kon datasets). Datasets from databases such as intact or STRING were discarded because they present only positive interactions, not contemplating knowledge of negative interactions, implying that the construction of classification models in the lack of negative responses would not make sense. In addition, it was desired to explore the existence of generalities in applying numerical

representation methods and predictive model development strategies, so it was preferred to explore varied datasets with reported and experimentally validated responses. All sequences were coded independently using the selected representation strategies, and data sets were generated by concatenating both proteins according to the interaction reported in the data sets used. Once the sequences were represented numerically, two additional strategies were applied to process or describe interactions from a linear point of view (only using the information of the protein sequences that interact with each other). In the first strategy, dimensionality reduction methods were applied via Principal component analysis (PCA) as a transformation strategy and development of descriptors via linear combinations of the initial points of each vector. Similarly, the second strategy applied kernel-PCA methods to ensure spatial transformations in nonlinear combinations. Thus, for each data set, three ways of representing the interaction were devised (concatenation, linear combinations, and nonlinear combinations), generating 60 alternatives to train for each dataset.

Table 1. Description of the datasets employed in this work

#	Dataset	Number of examples	Task	Reference
1	PROXiMATE-dg	2496	Evaluation of free energy differences	[12]
2	PROXiMATE-kd	2495	Estimation of kinetic constants	[12]
3	PROXiMATE-kon	561	Estimation of kinetic constants	[12]
4	SKEMPI-kon	855	Estimation of kinetic constants	[11]
5	SKEMPI-koff	860	Estimation of kinetic constants	[11]
6	SKEMPI-affinity	3180	Prediction of affinity between protein sequences	[11]

3.2 Training Predictive Models via Classic Machine Learning Algorithms and Deep Learning Architectures

Each dataset was trained by applying classical predictive model training algorithms such as support vector machine, KNN, decision tree, Bagging, Random Forest, AdaBoost, and Gradient tree boost, each with default hyperparameters following a classic training pipeline [18,19]. Besides, deep learning architectures based on convolutional neural networks were explored, summarized in Table 2. Finally, in the case of models trained via GCN methods, a two-block architecture was designed combining GCN and batch normalization layers, activated by a ReLU activation function and a learning layer based on linear methods, which varied the number of hidden layers between 5 and 30. Combining the three strategies explored, we have 17 predictive model development methods. Considering the 60 processed datasets, a total of 1020 trained models were evaluated in this work.

All generated models were evaluated applying classical performance metrics like MSE and R^2 score. Figure 1 shows the MSE (Mean squared error) distribution for all tasks evaluated in the three predictive model training strategies. The

Table 2. CNN architectures designed for exploring predictive models in protein-protein network tasks

Architecture	Pattern extraction configuration	Learning configuration	Optimization	Loss Function
A	A block of three Conv1D layers with MaxPooling activated by ReLU function	Two dense layers activated by tanh and linear functions	Adam	MSE
B	A block of three Conv1D layers with MaxPooling and dropout, activated by ReLU function	Two dense layers activated by tanh and linear functions	Adam	MSE
C	Pattern extraction blocks of two Conv1D and one MaxPooling, activated with ReLU function	One dense layer activated by linear function	Adam	MSE
D	Pattern extraction blocks of two Conv1D, one MaxPooling, and one droput, activated with ReLU function	One dense layer activated by linear function	Adam	MSE

results obtained by the classical machine learning application present a better performance than those obtained by the GCN and CNN architectures. However, when analyzing the best performances individually for each task, differences are observed between the method or strategy that achieves the best performance. Table 3 summarizes the best models for each dataset explored with each methodology evaluated.

Developing predictive models by applying pre-trained models for protein sequence representation achieves better performance when applying classical machine learning algorithms. However, using protein language model strategies in CNN or GCN architectures only achieves better results than using descriptors based on physicochemical properties. In the case of amino acid coding methods, better performance results were observed when combined with FFT for signal space representations and transformed into variance maximization spaces using techniques such as PCA or kernel-PCA, which was observed in the different tasks, both for classical machine learning and CNN methods. In the case of predictive models based on GCN architectures, there are significant differences in tasks such as affinity estimation compared to classical machine learning and CNN methods, achieving R^2 score performances above 0.8.

When evaluating the explored methods in terms of tasks and strategies, no marked results allow for generating a guide of recommendations for constructing predictive models. However, some considerations can be generalized. In the case of estimating and predicting association rates, CNN or classical machine learning methods achieve better performance than GCN. In this work, differences were found between the results concerning the prediction models of changes in association rate constants since SKEMPI-Kon and PROXiMATE-Kon datasets present the exact type of measurements with different measurement strategies and with other records, not being feasible to unify them for the training of a single model. However, it can be seen that as the number of records increases,

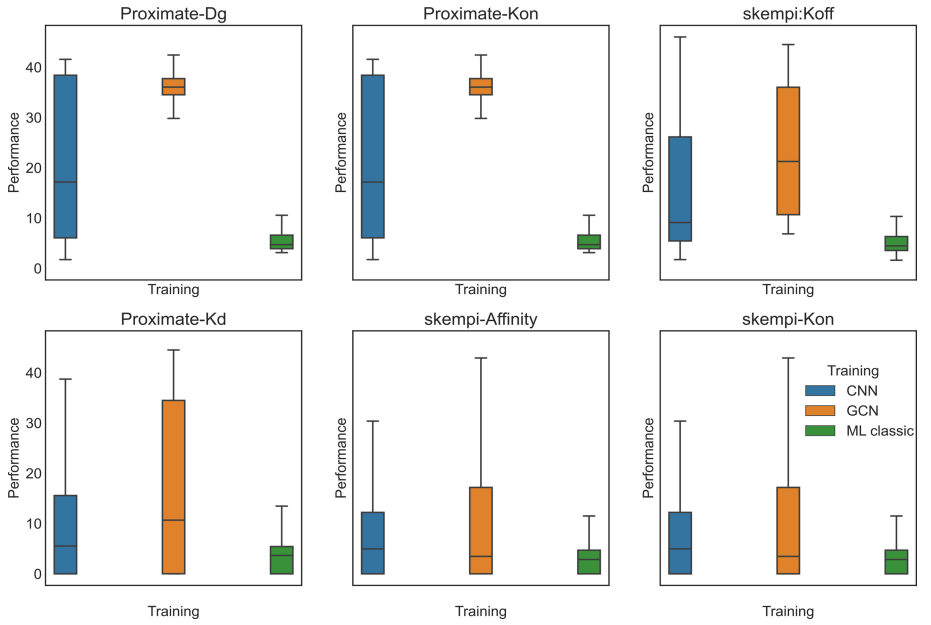

Fig. 1. Performances obtained for the predictive models in the six evaluated tasks for the three development strategies used to build predictive models. The methods explored in this work are summarized in performance distributions using boxplot visualization. The GCN and CNN methods present higher MSE distributions than the classical machine learning methods. Regarding distribution, CNN methods do not show significant differences compared to GCN methods. However, there tends to be a better performance in CNN methods. When analyzing the best versions of the models in each task, it is observed that there is no tendency for a specific process. However, in the cases of mutation evaluation and its effect on the interaction complex, CNN methods show better results than GCN methods. On the other hand, in protein interaction affinity methods, predictive models based on GCN achieve better performance and greater generalization of learning

deep learning methods are favored in terms of performance compared to classical machine learning methods, which is related to the generalization of behaviors, and the number of records in the dataset plays a crucial role in the training stages via deep learning methods. The GCN methods achieved the best performances with the same architecture in the affinity cases. However, the characterization or representation strategies of the nodes varied between amino acid encoders via physicochemical properties and numerical representations via pre-trained models.

The performances achieved by these models are similar, so employing GCN-based strategies for training this type of model allows the learning of the existing relationships between positive and negative interactions. However, further studies are required to discover and interpret these patterns from a biological point

Table 3. Summary of the best models for the explored tasks, evaluating the three strategies employed to develop predictive models

Task	Training strategy	Description	R2 performance
Proximiate-Dg	ML classic	OneHot as pretrained model and Random Forest as supervised learning algorithm	0.92
	CNN methods	**Hydropathy as encoder combined with PCA representations and using A architecture**	**0.94**
	GCN methods	Alpha structure combined with FFT transformation and 15 hidden layers in the created architecture	0.31
PROXiMATE-Kon	**ML classic**	**Bepler as pretrained model and Decision tree as supervised learning algorithm**	**0.72**
	CNN methods	Alpha structure as encoder combined with FFT representations and kernel-PCA as transformation and using D architecture	0.41
	GCN methods	Secondary structure as encoder and 15 hidden layers in the created architecture	0.01
SKEMPI-Koff	**ML classic**	**Alpha structure property as encoder combined with PCA transformation and Random Forest as supervised learning**	**0.85**
	CNN methods	Volume as encoder combined with FFT representations and PCA transformations and using D architecture	0.76
	GCN methods	Beta structure as encoder and 10 hidden layers in the created architecture	0.76
PROXiMATE-Kd	ML classic	Glove as pretrained model and Decision tree as supervised learning algorithm	0.76
	CNN methods	Bepler as pretrained models and architecture A to train the predictive model	0.26
	GCN methods	Bepler as pretrained model and 20 hidden layers in the created architecture	**0.88**
SKEMPI-Affinity	ML classic	Encoder Hydrophobicity combined with FFT applications and AdaBoost as supervised learning algorithm	0.05
	CNN methods	Fasttext as pretrained model and architecture A to train the predictive model	0.01
	GCN methods	**Secondary structure as encoder and 20 hidden layers in the created architecture**	**0.87**
SKEMPI-Kon	ML classic	Bepler pretrained models and KNN as supervised learning algorithm,	0.64
	CNN methods	**Secondary structure as encoder combined with FFT applications and kernel-PCA transformation and using the architecture C**	**0.72**
	GCN methods	Energetic as encoder strategy and 30 hidden layers in the created architecture	0.01

of view. Finally, in the case of mutation-related tasks, classical machine learning and CNN methods perform better than NGS methods in the case of tasks related to PROXiMATE-Dg, PROXiMATE-Kon, and PROXiMATE-Kd. However, the representation strategy needs clear patterns, which require evaluation and exploration to design the best models.

The effect of epochs was analyzed for the GCN methods in two models with the highest performance and a predictive model with the worst performance compared to the rest of the models developed. The different representation strategies are analyzed, and the epochs needed to present no significant changes in learning are compared. Figure 2 A shows the learning progress regarding epochs and the decrease in error for the loss function used in the GCN methods, and Fig. 2 B shows a zoom between the epochs 10 and 40 for the tasks SKEMPI-

Affinity and PROXiMATE-Kd. The predictive models for estimating protein interaction affinity (SKEMPI-affinity) and association constants (PROXiMATE-Kd) achieve learning and lower the prediction error in a minimum number of epochs, whereas the models for estimating interaction complex stability changes (PROXiMATE-Dg) exhibit higher instability in their learning, presenting perturbations during the process for all three types of protein representations. However, from epoch 70 onwards, a stabilization of the error is observed, except for the NLP-based methods, which continue to present variations during the process. With this information, it can be mentioned that FFT and NLP-based representations generate a positive synergy for affinity estimation problems and the use of predictive model training via GCN architectures, promoting better performance compared to classical methods and GCN architectures, which was also observed in Table 3. In the case of SKEMPI-Affinity and PROXiMATE-Kd the predictive models based on FFT and NLP representations achieve better performances in early training epochs.

3.3 Using the Explored Strategies to Build a Predictive Model

The development of the predictive model for predicting thermal stability changes in interaction complexes using the PROXiMATE-Kd dataset illustrates the methodologies proposed in this work. First, the design of architecture A of the CNN methods (See Table 2 for more details) is based on three pattern extraction blocks composed of conv1D and pooling layers, followed by a flattening layer and two learning layers. The pattern extraction layers are activated by the ReLU functions, while the learning layers are activated by tanh and linear functions. With the architecture designed, the next step consists of preparing the data sets. We work with the representation based on amino acid encodings via physicochemical properties for this. Specifically, the hydropathy property of those reported in [17] is used, and all the proteins in the dataset are encoded. The interaction network is assembled, and the interaction matrix is created, concatenating the protein representations indicated by the network. Once the network is made, a linear transformation is applied using PCA techniques. This transformation allows the two proteins to be "mixed" into a single representation via the linear combination developed by PCA to create its components. Then, the model is trained following a classical training pipeline [19], and performance metrics are obtained. Next, the results are reported, and the model is exported. The presented combination achieved the highest performance in the thermal stability evaluation problem, with a 0.94 R^2 score in the testing stage and a 0.96 R^2 score in the validation stage. Finally, the model can be used to explore new evaluations and stability changes to discover mutational variants with improved affinity. To do this, the mutational landscape of the protein of interest must be generated and encoded with the selected strategy. Then, the PCA transformation and the predictive model must be applied to predict the effects on the association constant (Kd), which can be validated using bioinformatics techniques such as simulations and molecular dynamics.

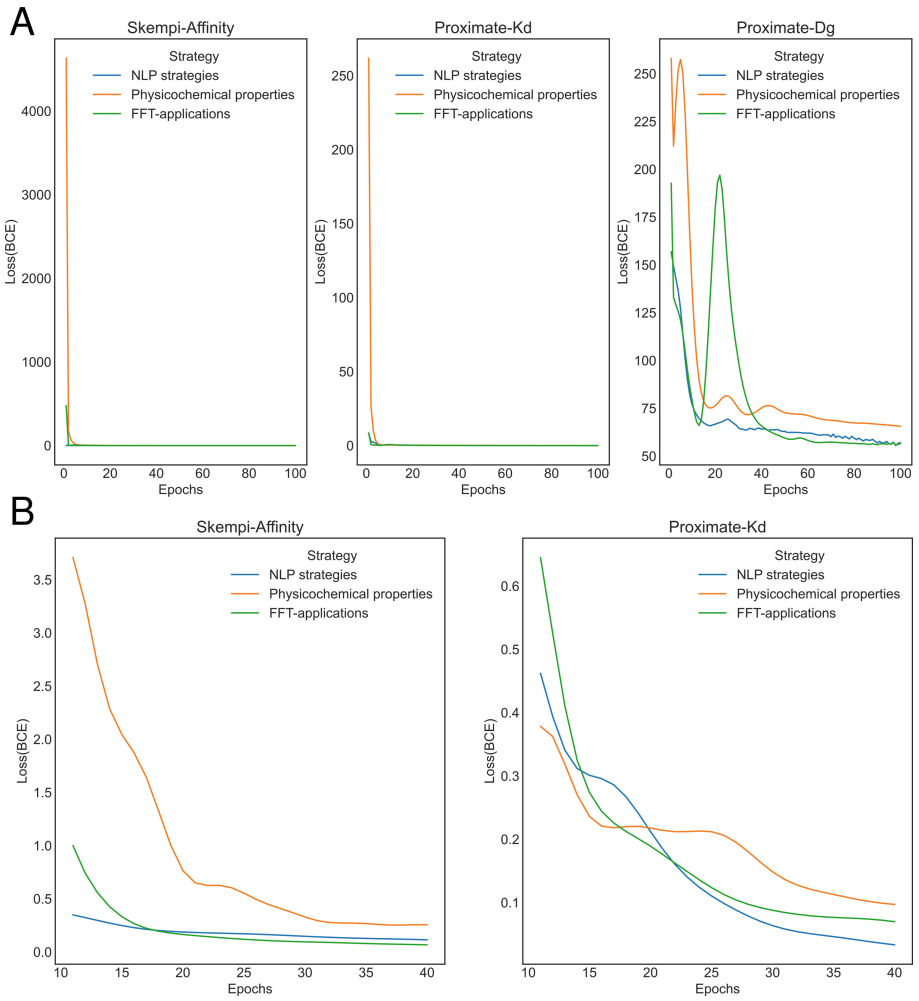

Fig. 2. Loss function v/s number of epochs for different GCN models employing varied numerical representation strategies to represent the protein sequences. A. Visualization of loss function for protein-protein affinity models (left), association constant estimation models (center), and stability estimation of structural complexes (right). Both affinity and association models achieve better performances in fewer epochs, exhibiting stability with a minimum of 10 runs, independent of the type of representations. In the case of the thermal stability predictive systems, more significant learning disruptions are observed, not achieving stability until 70 epochs of execution, and in the NLP representations, no stability of the learning process is achieved during all the epochs executed, this may be related to the low performance of the stability model, in comparison with the affinity estimation models. B. Zoom to the evaluation of learning between epochs 10 and 40. Significant improvement is observed at epochs 30 for PROXiMATE-Kd and 20 for SKEMPI-Affinity to benefit the models trained using amino acid coding and subsequent FFT transformation

4 Conclusions

This work explored methods of training predictive models for problems related to affinity interactions in protein complexes. Six tasks were evaluated, divided between evaluating protein-protein affinity and assessing the effect of mutations on interaction complexes. In addition, three predictive model training strategies and nine alternative numerical representations of interaction complexes were explored. The best models for interaction affinity evaluation were achieved using GCN architectures. In contrast, the best results for evaluating changes in complex structures were obtained by building CNN architectures or applying classical machine learning algorithms. In both problems, performances above 0.8 of R^2 score were achieved with no significant differences in the training and validation stages. No marked patterns were found when numerically representing the interaction complexes. However, using amino acid encoders combined with linear transformations via PCA improves performance in the evaluation models of stability changes. The FFT representations combined with physicochemical properties present better performances in evaluating changes in the interaction complex due to the effects of point mutations in the proteins. Finally, all the exposed strategies demonstrate the feasibility of the development of predictive models for protein interactions via linear information, becoming an alternative to the methods of structural representation and utilization of data from simulations and molecular dynamics, facilitating the efficient exploration of mutations, allowing navigation in landscapes and their reconstruction via the predictions of the generated models. In future work, the training strategies developed should be optimized, improving the construction of the architectures to promote higher performance. In addition, efficient landscape reconstruction and latent space exploration techniques should be incorporated to support protein design and enhancement techniques.

References

1. Bouvier, B.: Protein-protein interface topology as a predictor of secondary structure and molecular function using convolutional deep learning. J. Chem. Inf. Model. **61**(7), 3292–3303 (2021)
2. Bunkute, E., Cummins, C., Crofts, F.J., Bunce, G., Nabney, I.T., Flower, D.R.: PIP-DB: the protein isoelectric point database. Bioinformatics **31**(2), 295–296 (2015)
3. Cadet, F., et al.: A machine learning approach for reliable prediction of amino acid interactions and its application in the directed evolution of enantioselective enzymes. Sci. Rep. **8**(1), 16757 (2018)
4. Dallago, C., et al.: Learned embeddings from deep learning to visualize and predict protein sets. Current Protoc. **1**(5), e113 (2021)
5. Das, S., Chakrabarti, S.: Classification and prediction of protein-protein interaction interface using machine learning algorithm. Sci. Rep. **11**(1), 1–12 (2021)
6. Gaillard, T.: Evaluation of AutoDock and AutoDock vina on the CASF-2013 benchmark. J. Chem. Inf. Model. **58**(8), 1697–1706 (2018)

7. Gapsys, V., et al.: Large scale relative protein ligand binding affinities using non-equilibrium alchemy. Chem. Sci. **11**(4), 1140–1152 (2020)
8. Gil, C., Martinez, A.: Is drug repurposing really the future of drug discovery or is new innovation truly the way forward? Expert Opin. Drug Discov. **16**(8), 829–831 (2021)
9. Gupta, P., Mohanty, D.: SMMPPI: a machine learning-based approach for prediction of modulators of protein-protein interactions and its application for identification of novel inhibitors for RBD: hACE2 interactions in SARS-CoV-2. Briefings Bioinf. **22**(5), bbab111 (2021)
10. Huang, L., et al.: LGFC-CNN: prediction of lncRNA-protein interactions by using multiple types of features through deep learning. Genes **12**(11), 1689 (2021)
11. Jankauskaitė, J., Jiménez-García, B., Dapkūnas, J., Fernández-Recio, J., Moal, I.H.: SKEMPI 2.0: an updated benchmark of changes in protein-protein binding energy, kinetics and thermodynamics upon mutation. Bioinformatics **35**(3), 462–469 (2019)
12. Jemimah, S., Yugandhar, K., Michael Gromiha, M.: Proximate: a database of mutant protein-protein complex thermodynamics and kinetics. Bioinformatics **33**(17), 2787–2788 (2017)
13. Kairys, V., Baranauskiene, L., Kazlauskiene, M., Matulis, D., Kazlauskas, E.: Binding affinity in drug design: experimental and computational techniques. Expert Opin. Drug Discov. **14**(8), 755–768 (2019)
14. Kerrien, S., et al.: The intact molecular interaction database in 2012. Nucleic Acids Res. **40**(D1), D841–D846 (2012)
15. Liu, J., Gong, X.: Attention mechanism enhanced LSTM with residual architecture and its application for protein-protein interaction residue pairs prediction. BMC Bioinf. **20**, 1–11 (2019)
16. Luo, Y., et al.: ECNet is an evolutionary context-integrated deep learning framework for protein engineering. Nat. Commun. **12**(1), 1–14 (2021)
17. Medina-Ortiz, D.: Generalized property-based encoders and digital signal processing facilitate predictive tasks in protein engineering. Frontiers Mol. Biosci. **9** (2022)
18. Medina-Ortiz, D., Contreras, S., Quiroz, C., Asenjo, J.A., Olivera-Nappa, Á.: Dmakit: a user-friendly web platform for bringing state-of-the-art data analysis techniques to non-specific users. Inf. Syst. **93**, 101557 (2020)
19. Medina-Ortiz, D., Contreras, S., Quiroz, C., Olivera-Nappa, Á.: Development of supervised learning predictive models for highly non-linear biological, biomedical, and general datasets. Front. Mol. Biosci. **7**, 13 (2020)
20. Mewara, B., Lalwani, S.: Sequence-based prediction of protein-protein interaction using auto-feature engineering of RNN-based model. Res. Biomed. Eng., 1–14 (2023)
21. Parvathaneni, V., Kulkarni, N.S., Muth, A., Gupta, V.: Drug repurposing: a promising tool to accelerate the drug discovery process. Drug Discov. Today **24**(10), 2076–2085 (2019)
22. Rao, R., et al.: Evaluating protein transfer learning with tape. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
23. Sable, R., Jois, S.: Surfing the protein-protein interaction surface using docking methods: application to the design of PPI inhibitors. Molecules **20**(6), 11569–11603 (2015)
24. Siedhoff, N.E., Illig, A.M., Schwaneberg, U., Davari, M.D.: Pypef-an integrated framework for data-driven protein engineering. J. Chem. Inf. Model. **61**(7), 3463–3476 (2021)

25. Szklarczyk, D., et al.: The string database in 2021: customizable protein-protein networks, and functional characterization of user-uploaded gene/measurement sets. Nucleic Acids Res. **49**(D1), D605–D612 (2021)
26. Tsukiyama, S., Hasan, M.M., Fujii, S., Kurata, H.: LSTM-PHV: prediction of human-virus protein-protein interactions by LSTM with Word2Vec. Briefings Bioinf. **22**(6), bbab228 (2021)
27. Wang, R., Fang, X., Lu, Y., Yang, C.Y., Wang, S.: The PDBbind database: methodologies and updates. J. Med. Chem. **48**(12), 4111–4119 (2005)
28. Yang, F., Fan, K., Song, D., Lin, H.: Graph-based prediction of protein-protein interactions with attributed signed graph embedding. BMC Bioinf. **21**(1), 1–16 (2020)
29. Yu, J., Vavrusa, M., Andreani, J., Rey, J., Tufféry, P., Guerois, R.: InterEvDock: a docking server to predict the structure of protein-protein interactions using evolutionary information. Nucleic Acids Res. **44**(W1), W542–W549 (2016)
30. Yun, S., Lee, S., Park, J.P., Choo, J., Lee, E.: Modification of phage display technique for improved screening of high-affinity binding peptides. J. Biotechnol. **289**, 88–92 (2019)
31. Zeng, M., Zhang, F., Wu, F.X., Li, Y., Wang, J., Li, M.: Protein-protein interaction site prediction through combining local and global features with deep neural networks. Bioinformatics **36**(4), 1114–1120 (2020)
32. Zhang, H., et al.: Deep residual convolutional neural network for protein-protein interaction extraction. IEEE Access **7**, 89354–89365 (2019)
33. Zhao, L., Wang, J., Hu, Y., Cheng, L.: Conjoint feature representation of go and protein sequence for PPI prediction based on an inception RNN attention network. Mol. Ther. Nucleic Acids **22**, 198–208 (2020)