A Deep-Learning Approach for Predicting Protein-Glycan Interactions

Logan Woudstra, Paul Bakshi, Maxym Wojnowskyj, Evan Maloney, Sevryn Robinson, Ayyub Abdullahi, Shea McCormack, and Sheldon Roberts
Department of Computing Science
University of Alberta
{lwoudstr, paulanje, wojnowsk, ejmalone, sevryn, ayyub, samccorm, sheldon5}@ualberta.ca

Abstract

Glycans and proteins are fundamental molecular structures with a wide range of biological functions. Understanding how these two compounds interact has resulted in advancements in the development of vaccines and effective strategies for drug delivery. In this study, we introduce an adaptable pipeline for training deep-learning models to predict protein-glycan interactions, addressing limitations of the more rigid constraints of previous approaches. We achieve better performance than baseline models, but further investigation is required to overcome the challenges posed by the low frequency of high-binding interactions inherent in real-world protein-glycan datasets.

1 Introduction

Glycans are complex carbohydrate structures that are present on the surface of every living cell [29]. As such, these molecules play a significant role in major biological processes, including mediating host-pathogen interactions [23] and facilitating immunity recognition [4]. Structurally, glycans are composed of monosaccharides arranged in a non-linear branching manner. As a result, they are challenging to study as previous analytic techniques developed for linear molecular compounds, such as DNA or RNA, cannot be applied to glycans [17]. Glycan oligomers are few and far-fetched, making them difficult to chemically synthesize compared to proteins [9]. One way to ease these shortcomings is by predicting the potential strong binding points of glycans and proteins.

Understanding these relationships is essential as they provide key insights relevant to vaccine development [15], cancer therapeutics [28], and drug delivery strategies [12]. Recent machine learning approaches have achieved impressive results predicting these bindings [9, 8, 24, 19]. However, these existing models are limited, either being constrained to a fixed set of proteins or failing to account for variations in protein concentration.

In this paper, we investigate a range of methods for encoding glycans and proteins, and propose strategies to leverage these embeddings for predicting molecular binding strength using deep neural networks. To this end, we design a modular training pipeline that integrates interchangeable glycan encoders, protein encoders, and binding predictors. This pipeline trains and validates embeddings on a stratified dataset, enabling robust evaluation on unseen protein–glycan pairs. Using this framework, we achieve improved performance over existing architectures for this prediction task.

Preprint. Under review.

2 Background and Related Work

Pretrained Protein Encoders Evolutionary Scale Modeling (ESM) [21, 14, 18] is a series of pretrained transformers that have learned useful relationships between proteins to predict structural and functional properties. ESM2 and its drop-in replacement ESM Cambrain (ESMC) are popular models that have been successful when incorporated into protein-glycan interaction prediction systems [24, 19]. Thus, we investigate their performance as protein encoders in this work.

Glycan Representations Glycans can be represented with either the nomenclature proposed by the International Union of Pure and Applied Chemistry (IUPAC) or with the Simplified Molecular Input Line Entry System (SMILES) specification. IUPAC strings work at the monosaccharide level, representing the graph structure over sugars and the linkages connecting them. SMILES strings work at the atomic level, representing the graph structure over atoms and bonds. While these two representations characterize the same compound, they reflect different chemical properties. As such, different glycan encoders have been developed using these representations as input.

A common encoding of glycans over these graph representations is Morgan fingerprints. These fingerprints are vectors counting the occurrences of all unique subgraphs within the molecule, with each subgraph having a radius no greater than some specified value. Morgan fingerprints are useful as they are fast to compute and allow for a simple measure of similarity between molecules. However, they fail to capture global structures, focusing only on local connectivity.

Pretrained Glycan Encoders Several deep learning models have been developed to encode glycans. SweetTalk [6] is a long short-term memory recurrent neural network (LSTM) working at the monosaccharide level that takes into account glycan connectivity and composition. This was then followed by SweetNet [7], which again works at the monosaccharide level, but instead uses a graph convolutional neural network (GCNN). GIFFLAR [19] is a graph neural network (GNN) working over a glycan's atomic representation, and has been shown to achieve better performance than SweetNet in various downstream tasks. Finally, ChemBERTa [10, 3] is a transformer model working over general molecules at the atomic level, and has been shown to learn representations useful for molecular property prediction tasks.

While these models have achieved impressive results, they have several limitations when applied to our task. First, they are large, so training models from scratch for each unique architecture they are a component of is computationally infeasible. Furthermore, given our limited dataset, fine-tuning these large models will likely result in overfitting to our training set and overall decreased performance. Therefore, we only use the available pretrained versions of these encoders.

Deep Learning for Protein-Glycan Interaction Prediction Several prior works have used deep neural networks (DNN) to predict protein-glycan interactions, differing primarily in how they encode molecular inputs and measure binding strength outputs. GlyNet [9] uses Morgan fingerprints of glycan IUPAC strings to predict relative fluorescent units (RFU) for a fixed set of proteins across varying concentrations. MCNet [8] similarly applies Morgan fingerprints to glycan SMILES strings, but predicts fraction bound (f-bound) values instead, with performance comparable to GlyNet. LectinOracle [24] encodes glycans using SweetNet and proteins using ESM2 to predict a single RFU value per interaction. Finally, the GIFFLAR paper shows that when the glycan encoder is paired with ESM2, it achieves a lower mean absolute error in RFU prediction than LectinOracle.

Although these works approach the same problem, they all formulate the task differently, varying in whether the protein concentration is considered, whether predictions are for a given protein or a fixed set, the level at which the glycan is represented, and the measure of binding strength. This paper addresses these limitations by developing a flexible approach that uses the informative concentration constant, supports predictions for any protein, allows interchangeable glycan representations, and outputs the more universal f-bound. Due to these differences in problem formulation, prior models cannot serve as direct baselines. However, for rough comparison, we reimplement components from existing classifiers (their encoders and prediction head) and retrain them with our pipeline tailored to our specific task.

3 Methodology

3.1 Task Definition

To predict protein-glycan interactions, the input is comprised of a glycan represented by either its SMILE or IUPAC string, a protein represented by its amino acid string, and a positive scalar value for the protein concentration. Given these inputs, we train a model to predict the binding strength measured by the fraction bound (f-bound) [8], a scalar value between 0 and 1 measuring the fraction of glycans bound to the protein. We choose to use the f-bound due to its universality, as it unifies disparate quantitative measures for binding strength, including RFU, half-maximal inhibitory concentration (IC_{50}), and dissociation constant (K_d). For measuring performance, mean square error (MSE) is used as our evaluation metric, as it was utilized to evaluate GlyNet [9] and MCNet [8].

3.2 Dataset

Data from the glycan array experiments conducted by the Consortium for Functional Glycomics (CFG) [5] is used to train and test our models. This dataset contains 116,954 unique interactions between 611 glycans and 52 proteins at varying concentrations. However, this dataset has notable limitations, as it is relatively small and has a low frequency of strong binding interactions. The average f-bound is 0.017, with only 2.85% of the interactions being above 0.2. While this distribution is to be expected given that most proteins and glycans do not bind [28], it presents a challenge for training models to accurately predict both low and high affinity interactions.

To address the limitations of the CFG dataset, we experiment with adding supplementary training data from the GlycanML protein-glycan dataset [30] and BindingDB [22]. An additional 215,395 interactions from GlycanML and 171,820 interactions from BindingDB were gained. Given the disparity in dataset sizes, resampling the interactions from the smaller datasets was conducted to ensure balanced training across all three. Moreover, these datasets contain a higher proportion of strong binding interactions, with GlycanML and BindingDB, respectively, having average f-bounds of 0.107 and 0.329, and 13.0% and 50.0% of interactions exceeding an f-bound of 0.2. Since our goal is to achieve accurate predictions on the CFG test set, overfitting to the higher distributions of GlycanML and BindingDB was mitigated by training a separate output layer for each dataset. This design choice allows the model to learn from the varying distributions while preserving evaluation performance.

3.3 Data Stratification

To learn generalized patterns, it is important to train on a diverse set of proteins and glycans. At the same time, maintaining a balanced distribution across the training and testing sets is essential for reliable evaluation. We therefore use a stratified data splitting strategy to achieve both objectives. To perform a stratified data split, we separate both the glycans and proteins into several classes. Since the physical features of glycans and proteins contain valuable information for binding inference, we divide them into classes based on subcomponent similarity.

For glycans, we generate feature vectors of dimension 1024 using Morgan fingerprints with radii of at most 3 over SMILES strings, computed via the RDKit[1] Python library. The dimension size of 1024 and radius of 3 were selected based on examined cheminformatics literature [25]. With this vector, we use Euclidean distance as a similarity measure since it captures magnitude, the most intuitive and commonly suggested comparison between count vectors of Morgan fingerprints [1]. To classify the glycans into 3 clusters, we utilize unsupervised k-means clustering on the pairwise differences between the glycans' Euclidean distances. After randomly sampling an equal amount of glycans from each class, substantial differences are observed between classes in the values of various molecular features calculated using RDKit, such as the number of atoms, number of rotatable bonds, number of rings, molecular weight, and topological polar surface area (TPSA). Notably, glycan size emerges as the most prominent differentiating feature, with a clear separation of the 3 classes into common small glycans, medium-sized glycans, and large-sized glycans.

Our proteins are represented under the simplifying assumption that they consist of a single, linear amino acid sequence. However, in reality, the proteins used in our datasets are quaternary structures

[1] RDKit

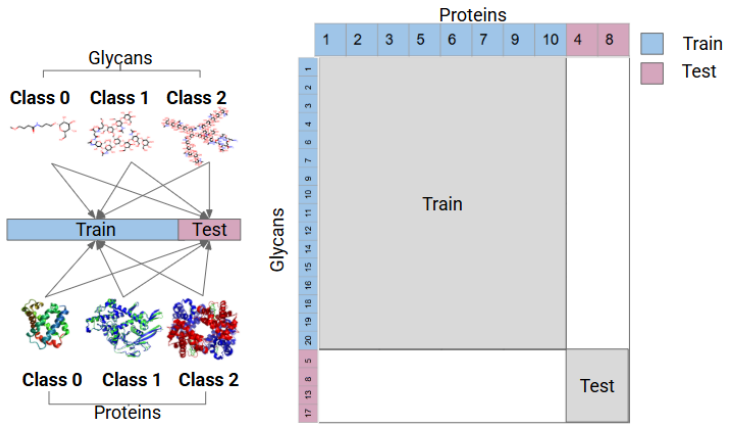


Figure 1: Data stratification using the 'AND' selection method. Protein images are adapted from Western Oregon University [11], and glycan structures are generated using RDKit.

of multiple polypeptide chains that contain their own amino acid sequence[2]. These chains fold and weave through each other to create complex three-dimensional structures. Since this complexity is not included in our dataset, we cannot break our protein down into subcomponent structures like we can with glycans. Instead, we utilize the BioPy[2] library to extract feature vectors describing general molecular properties, including amino acid sequence length, aromaticity, instability-index, and net-charge-pH7, among others. Again, we calculate the Euclidean distance between these vectors to cluster the proteins into 3 classes. Analyzing select samples, we observe that most feature values aside from sequence length are similar across the classes. Thus, the proteins are effectively split by size, similar to what resulted from splitting the glycans.

With class labels for all of our proteins and glycans, we can perform our train-test split using stratification. We do this by first specifying the percentage of glycans and proteins to take from each class and put into our test set. For this split, we experimented using two different approaches: 'AND' ~~and~~ 'OR', which we detail in Algorithm 1 available in the appendix.

The 'AND' selection method prevents the model from memorizing how specific glycans or proteins bind by ensuring that both components in every test sample are never seen during training. However, this lower bias comes at the cost of discarding samples with one molecule in the testing set and one in the training set, limiting the amount of training data available to the model. The 'OR' selection method allows individual molecules to appear in both sets, enabling full utilization of the dataset. For example, if $(protein_1, glycan_1)$ is in the training set, then $(protein_1, glycan_5)$ could appear in the testing set. This is visualized in Figure 1.

3.4 Training Pipeline

The training pipeline, presented in Figure 2, is a sequential stage algorithm that ~~can be outlined by~~ first encod~~ing~~ the glycans and proteins, followed by splitting the training and validation sets through stratified clusters, and finally training by optimizing the combined layers of our encoders and our binding predictor.

We begin by generating fixed-length embeddings for each glycan and protein using their respective encoders. These encoders, along with our binding predictor, are initialized with PyTorch neural network layers that are fine-tuned during training. After our initial encoding, we perform the training-validation set split utilizing the stratification strategy mentioned in Section 3.3. These samples are then passed into our binding predictor and the predictions are evaluated using our metrics calculation functions.

[2]BioPy

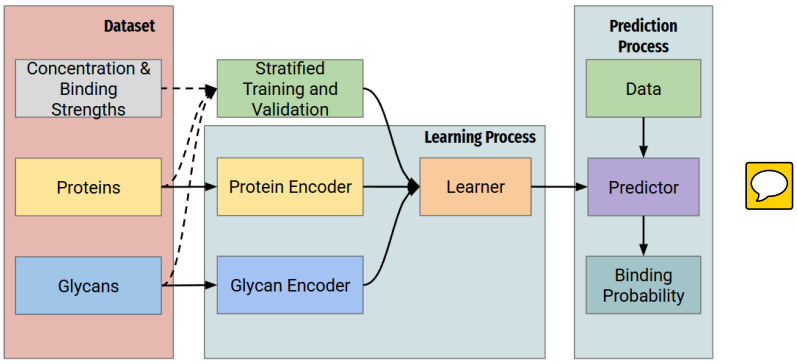


Figure 2: Diagram of our training pipeline.

3.5 Custom Encoders

3.5.1 Feature-Based Protein and Glycan Encoders

As detailed in subsection 3.3, we utilize global molecular features extracted via RDKit and BioPy to characterize glycans and proteins, respectively. Rather than limiting their use to data-stratification alone, we re-purpose their use as feature vectors as well. Given the intuitive and valuable structural information these feature vectors offer, we used them as our starting encoders, which we name after the libraries used to compute them (BioPy and RDKit).

3.5.2 Protein and Glycan GNN Encoders

We developed custom Graph Neural Network (GNN) encoders for proteins and glycans. For proteins, we extended a typical GNN architecture with convolutional layers and global mean pooling to incorporate amino acid feature representations, including chemical properties and position embeddings designed for protein sequence analysis. For glycans, we implemented a similar architecture with atom and bond features.

3.5.3 MPNN Glycan Encoder

Our Message Passing Neural Network (MPNN) [16] encoder models each glycan as a graph where nodes carry atom features and edges carry bond features. Node embeddings are iteratively updated using summed edge-conditioned messages passed from neighbors, with both message and update functions implemented as two-layer MLPs. After several rounds of message passing, node embeddings are aggregated via sum pooling and passed through a linear readout to produce the final glycan representation. Positional and structural information is concatenated to the node embeddings to improve expressivity [13].

3.5.4 Atomic Connectivity-Based Glycan Encoders

Motivated by the authors of MCNet [8], who identify atom-connectivity as the most critical information for understanding protein-glycan interactions, we design two graph convolutional neural networks leveraging glycan atomic connectivity. These encoders aim to preserve and utilize the structural features of glycans by modeling them as graphs at the atom and bond level.

The first encoder is the Atom-Only Atomic Connectivity Encoder (AConn-V1). ~~AConn-V1~~ does not explicitly use atomic or bond features, with the model instead learning structural patterns purely from atom connectivity and position. Thus, this encoder emphasizes the topological structure of the glycan. The second encoder is the Atom-and-Bond Atomic Connectivity Encoder (AConn-V2) ~~AConn-V2~~ extends the atomic connectivity framework by explicitly incorporating bond attributes into the message-passing process, such as bond type or involvement in ring structures. Leveraging this detailed chemical information allows AConn-V2 to produce more expressive embeddings that reflect both atomic identity and the nature of interatomic connections.

4 Results

4.1 Validation Model Performance

We evaluated multiple glycan and protein encoder combinations on the validation set (using the 'AND' method for data stratification) to establish initial candidates for our best model. In Table 1, we report the best encoder combination alongside a baseline model that always outputs the mean f-bound value of the dataset. Detailed results of all protein and glycan encoder combinations are available in the appendix (Table 5).

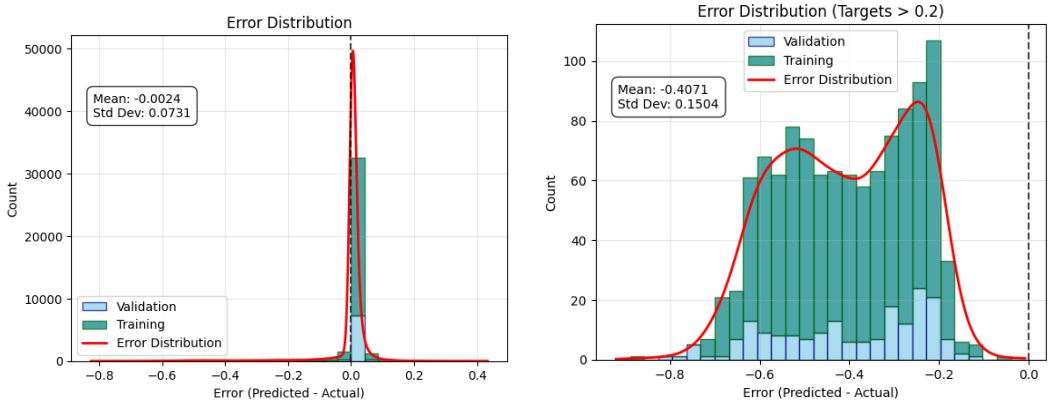
Encoder Combinations	Validation MSE ($\times 10^{-3}$)
MPNN + Protein GNN	4.26
Mean f-bound Predictor	4.89

Table 1: Results contrasting MPNN glycan encoder and Protein GNN encoder with the mean f-bound predictor.

We found that the best model on the validation set was the MPNN glycan encoder paired with the Protein GNN encoder, with this combination seeing a 12.9% improvement over the mean f-bound predictor. With this candidate encoder combination, we conducted further experiments on other parameters of our pipeline. We first compared the performance between the 'AND' and 'OR' methods for data stratification. Our results (Table 6 in the appendix) reveal that the 'AND' method achieves better validation performance, despite using only 60.21% of the available data. This highlights the importance of minimizing data leakage. We also experimented with different architectures for our DNN prediction head (Table 7 in the appendix). However, we found that our default DNN, comprised of three hidden layers of sizes 512, 256, and 128 with ReLU activations, outperforms all other settings we tested.

4.2 Investigating Prediction Patterns and Class Imbalance

Although the MSE of the MPNN glycan encoder paired with the Protein GNN encoder seemed adequate, it only achieves minimal improvement from the mean f-bound predictor. This motivated us to examine the actual prediction pattern to better understand the model's limitations.



(a) Error bin histogram for all samples.

(b) Error bin histogram for predictions on samples where the true binding values are greater than 0.2.

Figure 3: Error distributions showing model prediction patterns. The left histogram is across all samples. The right histogram is on samples with true binding values greater than 0.2.

This analysis revealed a significant contrast in prediction accuracy. While the overall error distribution (Figure 3a) appeared balanced, examining samples with binding values above 0.2 (Figure 3b) exposed a systematic bias. For these higher-binding samples, the model consistently underpredicts.

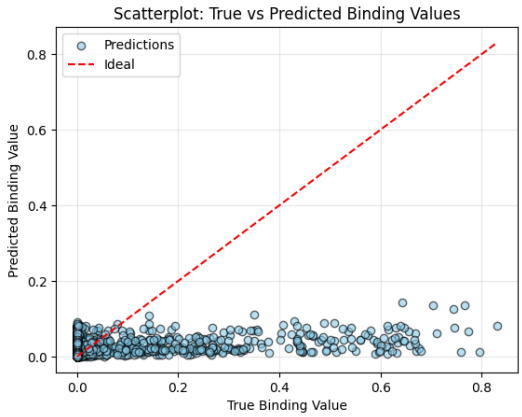


Figure 4: Scatterplot showing true binding values compared with the predicted binding values. The samples were evaluated using MPNN and Protein GNN encoders. The dashed red line represents ideal predictions where predicted values equal true values.

Figure 4 supports this claim, showing little relationship between true binding values and predicted binding values. Even for interactions with very high f-bound values, our model never predicts above 0.18. This is supported by our earlier mention that a majority of samples have a binding value below 0.2. The severe class imbalance in our dataset is the primary factor driving the model's predictive behavior. With such overwhelming representation of non-binding or weak-binding interactions, the model has effectively learned to predict near-zero values for most inputs, as this strategy minimizes overall error.

The inherent data imbalance represents a fundamental limitation of available training data. Our model misses high-binding interactions, despite having a low overall MSE.

4.3 Approaches to Overcoming Class Imbalance

To address the class imbalance problem in our dataset, we explored several approaches to improve our prediction accuracy. First, we utilized supplementary data from the GlycanML and BindingDB datasets. As shown in Table 2, despite a significantly greater number of training samples with a higher frequency of high-binding interactions, the validation MSE increased with supplemental data.

Datasets	Validation MSE ($\times 10^{-3}$)	Training Samples
CFG	4.26	29,582
CFG + BindingDB + GlycanML	4.88	416,430

Table 2: Validation set results from training using different collections of datasets.

We also investigated various loss functions, shown in Table 3 (Acronym definitions in Appendix 9). We found that no loss function outperformed our default MSE in capturing high-binding interactions, with MSE achieving the highest Pearson correlation score. This indicates that the model's difficulty in predicting high-binding values is likely the result of our dataset.

Evaluation Metric	MSE	RMSE	RMSLE	MAE	LMAE	Huber	Smooth-L1
Pearson	**0.35**	0.07	0.27	0.01	0.04	0.31	0.31

Table 3: Pearson correlation coefficient results for the MPNN and Protein GNN encoder combination on the validation set using different loss functions.

The consistent failure of these approaches emphasizes the challenges created by the class dominance of non-binding or weak-binding samples. The model continues to minimize overall error by pre-

dicting values close to zero in our validation set, capturing the dominant pattern in the training data. Unfortunately, our models still struggle to perform on samples with higher binding interactions.

4.4 Test Set Evaluation

We run all of the encoder combinations on the test set (available in Table 8 in the appendix) and list the top three performers along with an additional three baseline models in Table 4.

Encoder Combination	MSE ($\times 10^{-3}$)
RDKit + ESM2	**6.44**
MPNN + LSTM	6.63
MPNN + Protein GNN	6.65
GIFFLAR Paper Architecture	6.66
Mean f-bound	6.83
LectionOracle Architecture	12.7

Table 4: Test set results for different encoder combinations and baseline models.

Our final and best-performing model combined the RDKit encoder with the ESM2 encoder, achieving only a 6% improvement over the mean f-bound predictor's MSE. In addition, the pre-trained models constituting the architecture tested in the GIFFLAR paper (GIFFLAR and ESM2) achieved a very similar performance. However, the pre-trained models constituting the architecture tested in LectinOracle (SweetNet and ESM2) performed much worse. The small performance gap between these established baselines and our encoders suggests that our encoders are not the main limiters in our results. Instead, these results further support our observations about the limitations of our dataset mentioned in subsection 4.2.

Interestingly, our best-performing glycan encoder, RDKit, is also our simplest. Rather than using deep end-to-end learning of molecular structure like other encoders, it uses a set of manually engineered global features capturing the holistic chemical and structural properties instead. This supports the hypothesis that simpler models are less prone to overfitting and can generalize better on data with low signal.

Moreover, the gap between validation and test performance (4.26 $\times 10^{-3}$ versus 6.44 $\times 10^{-3}$ for our best models) further suggests some degree of overfitting despite our best efforts to avoid doing so. While our model may capture some patterns of protein-glycan interactions, its predictive power is limited, particularly for high-binding interactions.

5 Conclusion and Future Work

In this paper, we developed a modular pipeline to train deep-learning models for protein-glycan interaction prediction. We explored several different configurations and successfully trained a model that outperforms previous baseline architectures in this task. However, our increased performance is minimal, with the classifiers struggling to make accurate predictions for the rare but biologically important class of high-binding interactions.

Given the class imbalance in the dataset, anomaly detection presents a promising direction for future work, proving effective in related biological classification tasks [27, 26, 20]. Using these techniques, one could divide the prediction process into two distinct stages. First, an anomaly detection classifier could identify interactions with potentially high-binding values. Next, a regression model could be used to predict the strength of these selected interactions. This approach could help improve the model's accuracy on the underrepresented class.

Acknowledgments

We would like to thank Russ Greiner and Weijie Sun for their invaluable guidance and support throughout the entire research process. We also would like to thank our domain experts, Ratmir Derda

and Eric Carpenter, for their assistance in helping us navigate the unfamiliar field of computational glycobiology.

References

[1] *Daylight Theory Manual, 6. Fingerprints - Screening and Similarity*. Daylight Chemical Information Systems, Inc., PO Box 7737, Laguna Niguel, CA 92607, 1st edition, 2011.

[2] Quaternary (4°) structure. *Levels of Protein Organization, Foundations of Clinical Sciences*, 2014.

[3] Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta-2: Towards chemical foundation models, 2022.

[4] Ron Amon, Eliran Moshe Reuven, Shani Leviatan Ben-Arye, and Vered Padler-Karavani. Glycans in immune recognition and response. *Carbohydr Res*, 389:115–122, February 2014.

[5] Ola Blixt, Steve Head, Tony Mondala, Christopher Scanlan, Margaret E. Huflejt, Richard Alvarez, Marian C. Bryan, Fabio Fazio, Daniel Calarese, James Stevens, Nahid Razi, David J. Stevens, John J. Skehel, Irma van Die, Dennis R. Burton, Ian A. Wilson, Richard Cummings, Nicolai Bovin, Chi-Huey Wong, and James C. Paulson. Printed covalent glycan array for ligand profiling of diverse glycan binding proteins. *Proceedings of the National Academy of Sciences*, 101(49):17033–17038, 2004.

[6] Daniel Bojar, Diogo M. Camacho, and James J. Collins. Using natural language processing to learn the grammar of glycans. *bioRxiv*, 2020.

[7] Rebekka Burkholz, John Quackenbush, and Daniel Bojar. Using graph convolutional neural networks to learn a representation for glycans. *Cell Reports*, 35(11), Jun 2021.

[8] Eric J. Carpenter, Chuanhao Peng, Sheng-Kai Wang, Russell Greiner, and Ratmir Derda. Atom-level machine learning of protein-glycan interactions and cross-chiral recognition in glycobiology. *bioRxiv*, 2025.

[9] Eric J. Carpenter, Shaurya Seth, Noel Yue, Russell Greiner, and Ratmir Derda. Glynet: a multi-task neural network for predicting protein–glycan interactions. *Chem. Sci.*, 13:6669–6686, 2022.

[10] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: Large-scale self-supervised pretraining for molecular property prediction, 2020.

[11] Sourav Das. Chapter 2: Protein structure.

[12] Francisca Diniz, Pedro Coelho, Henrique O Duarte, Bruno Sarmento, Celso A Reis, and Joana Gomes. Glycans as targets for drug delivery in cancer. *Cancers (Basel)*, 14(4), February 2022.

[13] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875*, 2021.

[14] ESM Team. Esm cambrian: Revealing the mysteries of proteins with unsupervised learning, 2024.

[15] Andreas Geissner and Peter H. Seeberger. Glycan arrays: From basic biochemical research to bioanalytical and biomedical applications. *Annual Review of Analytical Chemistry*, 9(Volume 9, 2016):223–247, 2016.

[16] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[17] Mónica Guberman and Peter H. Seeberger. Automated glycan assembly: A perspective. *Journal of the American Chemical Society*, 141(14):5581–5592, 2019. PMID: 30888803.

[18] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *bioRxiv*, 2024.

[19] Roman Joeres and Daniel Bojar. Higher-order message passing for glycan representation learning, 2025.

[20] Dima Kagan, Juman Jubran, Esti Yeger-Lotem, and Michael Fire. Network-based anomaly detection algorithm reveals proteins with major roles in human tissues. *GigaScience*, 14:giaf034, 04 2025.

[21] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[22] Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N Jorissen, and Michael K Gilson. BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Res*, 35(Database issue):D198–201, 2006.

[23] Jon Lundstrøm and Daniel Bojar. Structural insights into host–microbe glycointeractions. *Current Opinion in Structural Biology*, 73:102337, 2022.

[24] Jon Lundstrøm, Emma Korhonen, Frédérique Lisacek, and Daniel Bojar. Lectinoracle: A generalizable deep learning model for lectin–glycan binding prediction. *Advanced Science*, 9(1):2103807, 2022.

[25] Darko Medin. Data science for drug discovery research -morgan fingerprints in python. *Medium*, 03 2022.

[26] Tomer Michael-Pitschaze, Niv Cohen, Dan Ofer, Yedid Hoshen, and Michal Linial. Detecting anomalous proteins using deep representations. *NAR Genom Bioinform*, 6(1):lqae021, February 2024.

[27] Laurent Perez and Mathilde Foglierini. RAIN: a machine learning-based identification for HIV-1 bNAbs. *Res Sq*, March 2024.

[28] Osamu Shimomura, Tatsuya Oda, Hiroaki Tateno, Yusuke Ozawa, Sota Kimura, Shingo Sakashita, Masayuki Noguchi, Jun Hirabayashi, Makoto Asashima, and Nobuhiro Ohkohchi. A novel therapeutic strategy for pancreatic cancer: Targeting cell surface glycan using rbc2lc-n lectin–drug conjugate (ldc). *Molecular Cancer Therapeutics*, 17(1):183–195, 01 2018.

[29] Ajit Varki, Richard D. Cummings, Jeffrey D. Esko, Pamela Stanley, Gerald W. Hart, Markus Aebi, Alan G. Darvill, Taroh Kinoshita, Nicolle H. Packer, Ronald L. Schnaar, and Peter H. Seeberger. *Essentials of glycobiology*. Cold Spring Harbor Laboratory Press, United States, 3rd edition, 2017.

[30] Minghao Xu, Yunteng Geng, Yihang Zhang, Ling Yang, Jian Tang, and Wentao Zhang. GlycanML: A multi-task and multi-structure benchmark for glycan machine learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Appendix

A.1 Data Stratification Algorithm

Algorithm 1 Data Stratification

Inputs: Proteins - $P = \{(protein_i, class_j)\}_{i=1}^{m}, j \in \{0, 1, 2\}$
Glycans - $G = \{(glycan_i, class_j)\}_{i=1}^{n}, j \in \{0, 1, 2\}$
Interactions dataset - $D = \{(protein_i, glycan_i)\}_{i=1}^{k}$
Testing split - $s \in [0, 1]$
Selection method - $method \in \{\text{AND, OR}\}$
Outputs: Training dataset - D_{train}
Testing dataset - D_{test}

$P_{test} \leftarrow \text{SelectTestProteins}(P, s)$
$G_{test} \leftarrow \text{SelectTestGlycans}(G, s)$
$D_{train} \leftarrow \emptyset$
$D_{test} \leftarrow \emptyset$
for $protein, glycan$ in D **do**
    if $method$ is AND **then**
        if $protein \in P_{test}$ **and** $glycan \in G_{test}$ **then**
            add $(protein, glycan)$ to D_{test}
        else if $protein \notin P_{test}$ **and** $glycan \notin G_{test}$ **then**
            add $(protein, glycan)$ to D_{train}
        else
            discard $(protein, glycan)$
    else if $method$ is OR **then**
        if $protein \in P_{test}$ **or** $glycan \in G_{test}$ **then**
            add $(protein, glycan)$ to D_{test}
        else
            add $(protein, glycan)$ to D_{train}

return D_{train}, D_{test}

A.2 Validation Set Results

A.2.1 Validation MSE for Different Encoder Combinations

	BioPy	ESM2	ESMC	LSTM	Protein GNN
AConn-V1	4.68	4.68	4.77	4.72	4.74
AConn-V2	4.77	4.78	4.83	4.76	**4.51**
ChemBERTa	4.64	4.58	4.63	4.65	4.64
GIFFLAR	4.77	4.87	4.71	4.69	4.82
Glycan GNN	4.81	4.78	4.81	4.77	4.93
MPNN	4.56	4.75	4.63	4.54	**4.26**
RDKit	4.78	4.66	4.72	4.64	4.69
SweetNet	4.73	4.72	4.73	4.67	**4.48**
SweetTalk	4.85	4.82	4.82	4.98	4.98

Table 5: Validation MSE ($\times 10^{-3}$) for different encoder combinations, with protein encoders being the rows and glycan encoders being the columns.

A.2.2 Validation MSE for Different Data Stratification Methods

Selection Method	Validation MSE ($\times 10^{-3}$)	Dataset Usage (%)
AND	4.26	60.21
OR	5.30	100.0

Table 6: Validation MSE ($\times 10^{-3}$) for 'AND' and 'OR' Data Stratification Methods on Best Protein and Glycan Encoder

A.2.3 Validation MSE for Different Prediction Heads

	Small	Medium	Large
ReLU	5.43	**4.26**	5.79
Leaky ReLU	5.46	5.24	5.54
Sin	5.87	5.87	5.90

Table 7: Validation MSE ($\times 10^{-3}$) for using the MPNN and Protein GNN encoder combination with different DNN architectures. Activation functions are the rows, and network sizes are the columns. Small = (64, 32), Medium = (512, 256, 128), and Large = (1024, 512, 256).

A.3 Test Set Results

	BioPy	ESM2	ESMC	LSTM	Protein GNN
AConn-V1	17.0	18.3	56.4	6.80	6.86
AConn-V2	11.1	7.30	6.75	6.90	6.93
ChemBERTa	7.52	6.73	7.48	6.90	6.90
GIFFLAR	7.02	6.67	6.88	6.82	6.68
Glycan GNN	7.84	6.84	38.1	8.33	7.56
MPNN	6.79	7.27	6.80	**6.63**	**6.65**
RDKit	7.22	**6.44**	7.55	6.76	6.77
SweetNet	6.66	12.7	27.1	6.98	7.02
SweetTalk	6.87	6.75	6.85	6.87	6.95

Table 8: Test MSE ($\times 10^{-3}$) for different encoder combinations, with protein encoders being the rows and glycan encoders being the columns.

Acronym	**Full Loss Name**
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
RMSLE	Root Mean Squared Logarithmic Error
MAE	Mean Absolute Error
LMAE	Log-Mean Absolute Error
Huber	Huber Loss
Smooth-L1	Smooth L1 Loss

Table 9: Mapping of loss function acronyms to their full name definitions.

A.4 Configuration File

We use a configuration file to specify all of the parameters for training the model, including the protein and glycan encoders, the type of stratification split, the size of the validation set created through stratification, the binding predictor, the loss function, whether to transform our predictions and targets using $log(1 + x)$, as well as other general training parameters like batch size, learning rate, number of epochs, and whether to train on the entire training set without a validation set. After the training is complete, we track our experiment and save all of the results and model weights for comparison with other models. The general mapping of our pipeline and all its components can be seen in Figure 2.