Thèse n° 10 309

A Geometric Transformer for Structural Biology: Development and Applications of the Protein Structure Transformer

Présentée le 30 octobre 2023

Faculté des sciences de la vie
Unité du Prof. Dal Peraro
Programme doctoral en physique

pour l'obtention du grade de Docteur ès Sciences

par

Lucien Fabrice KRAPP

Acceptée sur proposition du jury

Prof. F. Mila, président du jury
Prof. M. Dal Peraro, directeur de thèse
Prof. A. Bonvin, rapporteur
Prof. S. Ovchinnikov, rapporteur
Prof. B. Correia, rapporteur

École
polytechnique
fédérale
de Lausanne

2023

Acknowledgements

Completing this thesis has been an incredibly rewarding journey, one that would not have been possible without the support, encouragement, and camaraderie of so many people who have been a part of my life throughout this endeavor.

First and foremost, I would like to express my deepest gratitude to my family. In particular, my mom deserves special recognition for always being supportive and for introducing me to the wonders of science from an early age. I would not be where I am today without her unwavering belief in me. I also want to thank my dad for nurturing my innate curiosity and for suggesting that I delve into the field of physics. The path he encouraged me to take has been enriching beyond measure.

I extend my heartfelt thanks to my long-time close friends: Frederic, Yannick, William, Valentin, and Romain. Whether we were partying, biking, or navigating the complexities of an intricate board game, your friendship has been invaluable. Your constant support has served as a reminder that there is more to life than the academic world, grounding me in the best way possible.

A special mention goes to the previous lab members for their warm welcome when I first joined the lab. Giorgio, your positivity as a supervisor was infectious, making even difficult days at work seem like learning opportunities. Sylvain, sharing an office with you has been a delightful and educational experience. Your friendship made every day enjoyable.

I owe a tremendous debt of gratitude to Matteo, who has been the epitome of an ideal boss: supportive without being overly micromanaging. The freedom to pursue exciting projects under your guidance is a primary reason why I chose to continue my academic journey here after completing my master. Luciano, your mentorship and readiness to assist whenever needed has been instrumental, not to mention your truly amazing BBQs!

My thanks also extend to the nanopore team. Chan, it was an honor to work alongside you on the nanopore project. Your support helped me navigate the complexities of the work, and I have learned so much from our time together. Juan, your ever-present willingness to engage in discussions, both scientific and non-scientific, has made you a great colleague to have around.

Acknowledgements

I have had the good fortune of sharing an office with some incredible people. Zhidian, your kindness and willingness to engage have made our conversations truly enriching. Verena, you are not just highly competent but also a joy to be around, especially during our after-work beers. Fabio, your life stories and travel experiences have made our shared office space anything but dull.

Last but certainly not least, a big thank you to Fernando, your warmth and friendliness are contagious, and I have no doubt that you will make the office a great place in the coming years, to Sarah, whose positive energy always lighten up everyone, and to our administrative assistant, Magali, who has been a lifesaver on multiple occasions.

To everyone in the lab, you make up an extraordinary team of warm, friendly, and brilliant individuals. I consider myself lucky to have been a part of such an inspiring environment. Working here has not just been an educational experience, but also a great deal of fun.

Thank you all for being a part of this incredible journey.

Lausanne, October 16, 2023 Lucien Krapp

Abstract

Proteins, the central building blocks of life, play pivotal roles in nearly every biological function. To do so, these macromolecular structures interact with their surrounding environment in complex ways, leading to diverse functional behaviors. The prediction of these interactions, especially those involving protein-protein interfaces and other molecular interactions, has long been a major challenge in the field of structural biology. However, with the recent surge in advanced computational methods, we are now on the brink of making significant breakthroughs.

I developed the Protein Structure Transformer (PeSTo), a deep learning method that leverages a novel operation called geometric transformers. PeSTo only requires as input the atomic coordinates and element names of the structure. This general approach allows the model to be applied to many different tasks without requiring any computationally expensive data processing. The method demonstrated an impressive performance in accurately predicting the protein-protein binding interfaces, outperforming the state-of-the-art methods. I extended PeSTo to predict protein binding interfaces in general, detecting and distinguishing protein interfaces with nucleic acids, ligands, ions and lipids. I also show that PeSTo can be specialized for the prediction of interfaces with specific molecules such as carbohydrates and cyclodextrins.

The defining advantages of PeSTo are its low computational cost and robustness. Unlike many existing tools, PeSTo allows for high-throughput processing of structural data, including molecular dynamics ensembles. This ability to process large amounts of data efficiently enabled us to predict binding interfaces for all AlphaFold predicted structures. This ensemble of binding interfaces, which we call the "interfaceome", has the potential to help the identification of protein binding domains and accelerate research.

Beyond protein interacting interface prediction, PeSTo has been applied to another challenging problem in protein design: the prediction of protein sequences from backbone scaffolds. The newly trained model, called CARBonAra (Context-aware Amino acid Recovery from Backbone Atoms and heteroatoms), performs on par with the state-of-the-art methods for the in-silico sequence recovery rate. Unlike other methods, CARBonAra is able to predict amino acid sequences from a backbone scaffold with other non-protein atoms such as nucleic acids and ligands. This ability to consider non-protein entities in the design of protein sequences opens a myriad of possibilities, including the design of proteins that can interact with specific molecules, such as nucleic acids, leading to potential applications in therapeutics and

biotechnology.

The potential of PeSTo expands as the available protein structure data, or the "foldome", continues to grow. Given the rapid advancements in structure determination techniques, such as cryo-EM, the foldome is expected to expand significantly in the coming years. Complementing this, AlphaFold serves as a tool for bridging the gap between sequences and structure. The ability of PeSTo to utilize these expanding resources will further enhance the scope of applications.

In conclusion, the development of PeSTo represents a significant leap forward in the application of deep learning in structural biology. It not only provides an efficient and accurate tool for predicting protein interactions, but also opens a new frontier in protein design considering non-protein entities. By leveraging the rapidly expanding protein structure data, PeSTo holds vast potential for a broad spectrum of applications in structural biology and material science.

Keywords: structural biology, deep learning, protein-protein interactions, protein binding interfaces, protein design, inverse folding problem, geometric transformers

Résumé

Les protéines, en tant qu'éléments centraux de la vie, jouent un rôle essentiel dans presque toutes les fonctions biologiques. Ces structures macromoléculaires interagissent avec leur environnement de manière complexe conduisant à divers comportements fonctionnels. Un des défis majeurs dans le domaine de la biologie structurelle a longtemps été de pouvoir prédire les domaines impliqués dans ces interactions, essentiellement avec d'autres protéines mais également avec d'autres molécules. Maintenant, grâce à la poussée des récents progrès faits dans les méthodes computationnelles, nous sommes sur le point de faire une avancée significative.

J'ai développé le Protein Structure Transformer (PeSTo), une méthode d'apprentissage profond (deep learning) qui exploite une nouvelle opération appelée transformation géométrique (geometric transformers). PeSTo ne nécessite en entrée que les coordonnées atomiques et les noms des éléments d'une structure. Cette approche générale permet l'application du modèle à diverses tâches, sans nécessiter de calculs couteux dans le traitement des données. La méthode a démontré une performance impressionnante dans la prédiction précise des interfaces d'interaction entre protéines, surpassant les méthodes de pointe. J'ai étendu PeSTo à la prédiction des interfaces de liaisons protéiques en général, permettant ainsi la détection et la discrimination entre les interfaces impliquées dans la liaison avec des acides aminés, ligands, ions ou lipides. J'ai aussi montré que PeSTo pouvait être adapté dans la prédiction d'interfaces avec des molécules spécifiques tel que les glucides et les cyclodextrines.

Les avantages caractéristiques de PeSTo sont son faible coût de calcul et sa robustesse. Contrairement à de nombreux outils existants, PeSTo permet un traitement à haut débit des données structurelles, notamment les ensembles de dynamique moléculaires. Cette capacité à traiter efficacement de grandes quantités de données nous a permis de prédire les interfaces de liaison pour toutes les structures prédites par AlphaFold, que nous appelons «interfaceome». Cet ensemble d'interfaces de liaisons, a le potentiel de permettre la découverte de protéines d'intérêt et d'accélérer la recherche.

Au-delà de la prédiction des interfaces de liaison, PeSTo a aussi été appliqué à un problème épineux dans le domaine de conception de protéines (protein design) : la prédiction des séquences en acides aminés à partir de l'échafaudage du squelette protéique. Ce nouveau modèle, nommé CARBonAra (Context-awareAmino acid Recovery from Backbone Atoms and heteroatoms), fonctionne à la hauteur des méthodes de pointe en ce qui concerne le taux de ré-

cupération de séquences in-silico. Mais contrairement à d'autres méthodes, CARBonAra est capable de prédire la séquence en acides aminés à partir d'un échafaudage du squelette protéique avec des atomes non-protéiques, tel que des acides nucléiques ou des ligands. Cette capacité à prendre en compte des entités non-protéiques dans la conception de séquences protéiques ouvre une multitude de possibilités, y compris dans la conception de protéines pouvant interagir avec des molécules spécifiques telles que les acides nucléiques, conduisant à de potentielles applications thérapeutiques et biotechnologiques.

Le potentiel de PeSTo se développe avec l'expansion des données structurelles des protéines, ou «foldome». Étant donné les avancées rapides du développement des techniques dans la détermination des structures protéiques, tel que la cryo-EM, on s'attend à une expansion significative du foldome dans ces prochaines années. A cela s'ajoute des outils tels que AlphaFold qui permet de faire le lien entre la séquence et la structure. La possibilité de PeSTo à intégrer ces ressources en développement permettra encore d'élargir la portée de ses applications.

En conclusion, le développement de PeSTo représente une percée notable dans l'application des méthodes d'apprentissage profond en biologie structurelle. PeSTo fournit non seulement un outil efficace et précis pour la prédiction des interactions protéiques, mais ouvre également la voie à la conception de protéines en tenant en compte des entités non protéiques. En exploitant les données en expansion rapide des structures protéiques, PeSTo possède un vaste potentiel permettant le développement d'un large spectre d'applications en biologie structurelle et science des matériaux.

Contents

List of Figures

1 Introduction

1.1 Structural Biology

Proteins are the workhorses of the cell, playing crucial roles in virtually every biological process. They serve as enzymes that catalyze biochemical reactions, provide structural support, function as transporters, and perform myriad other tasks that are essential for life. Understanding proteins is central to biology, as they are the building blocks of cellular function and the ultimate executors of genetic information[1].

Over the past two decades, structural biology has risen as a key field within biology, offering unparalleled insights into the molecular architecture of proteins[2]. It moves beyond the one-dimensional string of amino acids that make up a protein to provide a three-dimensional view, revealing how proteins fold, interact, and function at the molecular level. These three-dimensional structures offer critical insights into the function of proteins, their interactions with other molecules, and provide a foundation for drug discovery, among other applications.

Structural biology has not only advanced our understanding of fundamental biology but has also provided actionable insights for medical research. The advent of technologies like X-ray crystallography, Nuclear Magnetic Resonance (NMR) spectroscopy, and more recently, Cryo-Electron Microscopy (Cryo-EM), has revolutionized the field, enabling researchers to visualize proteins and other biomolecules at atomic resolution. This has had a profound impact on drug discovery, disease understanding, and even biotechnology. Given its significance and the advances made in recent years, structural biology is positioned at the forefront of biological sciences[3].

1.1.1 Sequence, Structure, Function

Structural biology is a field that examines the molecular structure of biological macromolecules, particularly proteins, and their relationship to function. The fundamentals of structural biology are encapsulated by the sequence-structure-function paradigm, as illus-

trated in Figure 1.1.

Proteins are composed of a sequence of amino acids that fundamentally determines structure and function. The way this chain folds creates its 3D structure, which is dictated by the thermodynamics of interactions among amino acids and their environment. The solvent and thermal fluctuations drives the folding process[4, 5].

The encoding of protein structures in this sequence space strikes a good compromise between consistency, diversity and similarity. Specific sequences reliably fold into a unique 3D structure. The diversity in the amino acid chemistry results in a large structural and functional space. On the other hand, the chemical similarity between some amino acids allows for the gradual sampling of proteins through mutations allowing not only the optimization of proteins, but also the creation of novel specific functions. Proteins with similar sequences often results in similar structures, harboring similar or different functions[6, 7].

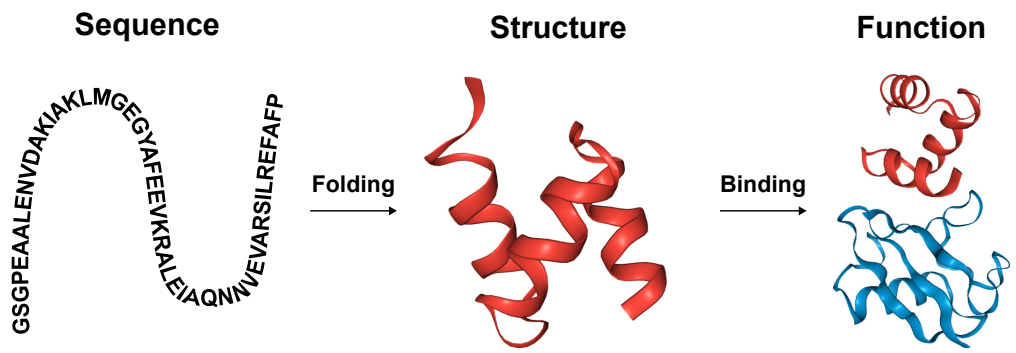

Figure 1.1: Illustration of the relationship between sequence, structure and function. The sequence describes the content and order of amino acids from the amino terminal of the protein, called the chain of amino acids. This protein chain of molecules folds into a specific structure. The structure in turn determines the function of the protein. In this illustration, the protein performs its function by binding to a given partner.

Protein sequences have been studied through an evolutionary lens which resulted in many breakthroughs over the decades. However, the sequence is not the closest causal link to the function. It can therefore be vulnerable to the correlation against causality problem. Therefore, protein structures have been studied to better understand the fundamental mechanism involved in protein stability and interactions with its environment such as the solvent, other proteins, membranes, nucleic acids and other molecules[7].

1.1.2 Protein structure

The organization of proteins is subdivides into four levels. The primary structure is defined as the amino acid sequence of the chain. The secondary structure describes the local arrangements such as α-helices and β-sheets, common patterns of folding driven by hydrogen bonding between backbone atoms. The tertiary structure refers to the overall three-dimensional

shape of the protein, defined by the arrangement of secondary structural elements and the spatial positioning of individual amino acids. Finally, the quaternary structure refers to the assembly of two or more protein subunits which can be identical or different. These different levels of structure, from the amino acid sequence to the assembled protein complex, provide a more detailed understanding of how protein structure relates to function.

Over decades, various experimental methods, including X-ray crystallography, NMR spectroscopy, and most recently, Cryo-EM, have been developed to elucidate these complex structures[3]. The structure of a protein is far from static; protein dynamics plays a vital role in function, often involving conformational changes. As we explore the structure-function paradigm, we understand that interactions are crucial, which leads us to the next level of complexity: protein interactions.

1.1.3 Protein interactions

Similar to cogs within a complex mechanism, proteins rarely function in isolation in cells. They cost energy to the cell to produce and have specific roles. They can have among others a structural role, catalyze chemical reactions, coordinate cell signalling pathways or transport molecules. To perform these tasks, they interact with other proteins, nucleic acids, small molecules, and other cellular components. These interactions can be transient or stable, and they have significant implications on the protein function[8].

Ultimately, the function of the protein is the most interesting. Understanding what a protein does in a living cell or an organism is a fundamental yet not simple question. Knowing the function of a protein in isolation often does not tell us how it functions in a living organism. However, knowing the interacting partners allows us to understand the biological context in which the protein is acting. This is of particular importance when considering diseases. For example, genetic disorders and cancers are often due to malfunction of proteins that have lost or acquired different binding partners leading to pathogenic consequences. Likewise, foreign agents interact with cellular proteins. Therefore, the understanding of the biological context of a protein is crucial in targeting and designing therapeutics against different diseases[9].

1.1.4 Structure modeling

Numerous computational methods have been developed to model and study protein structures, each with its unique advantages and applications. One such approach is molecular dynamics, which is primarily employed to understand the intricate behavior of proteins in a biological context[10]. This method simulates the motion, interactions, and conformational changes of proteins over time, providing insights into their function and interaction mechanisms. More generally, integrative modeling takes advantage of multiple types of computational methods and experimental evidence to model large protein complexes. By lever-

aging diverse data sources, integrative modeling can construct more accurate and reliable models, particularly for large or dynamic systems where single-method approaches may fall short[11].

The field of structural biology has seen a remarkable increase in available data, largely due to advancements in techniques like X-ray crystallography, nuclear magnetic resonance (NMR), and cryo-electron microscopy. Most of the known protein structures have so far been obtained by X-ray crystallography. This method yields a high resolution structure; however the structure represents a static form of the protein. The two other techniques, less widely used so far, provide information about the dynamics and intermolecular interactions of a complex, while the resolution due to the intrinsic noise of these methods is lower. This abundance of data, including much larger set of sequences with experimentally unknown fold, is especially conducive for deep learning methods, which require large datasets to train accurate models. The confluence of extensive structural biology data with advanced deep learning algorithms offers a powerful combination for tackling complex biological challenges, ranging from protein structure prediction (AlphaFold[12]), inverse folding problems (ProteinMPNN[13]) and protein binding interface prediction (MaSIF-site[14]).

1.2 Deep Learning

1.2.1 Model, Loss, Optimizer

In the domain of machine learning and deep learning, the model, loss, and optimizer are the tree main components, as illustrated in Figure 1.2.

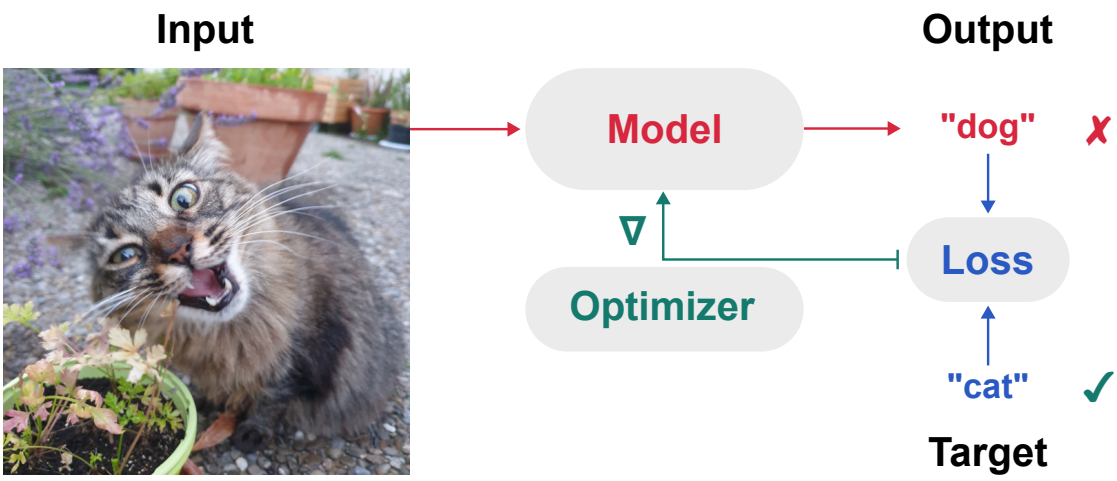

Figure 1.2: Illustration of a general deep learning framework. First, the model can be thought of a black box that can approximate any mapping from an input to an output. However, the model contains parameters that can be tuned to obtain a desired output. Then, the loss or objective function measure how far the output of the model is from the expected target. In conjunction with the model, the loss defines the optimization landscape of the system. Finally, the optimizer describes how to modify the parameters of the model to reduce the loss.

Starting with the model, it can be thought of as a function. It takes an input, processes it, and then outputs a result. The goal is to have this model transform the inputs into desired outputs, which usually represent some kind of prediction or classification based on the data the model was trained on. Models possess properties of interpolation and extrapolation, which refer to their capacity to predict outputs from inputs that are within or outside the range of the training data, respectively. However, models can encounter problems like underfitting and overfitting. Underfitting refers to a model's lack of fit to the training data, implying that the model fails to capture the underlying patterns, while overfitting suggests the model is so closely fit to the training data that it fails to generalize well to unseen data. The complexity of the model, governed by parameters or degrees of freedom, can be optimized to mitigate these issues[15].

The next piece of the puzzle is the loss function. It is essential as it defines the objective of a task. In essence, it quantifies the deviation of the predicted output from the actual output, thereby measuring the quality of a model. A lower loss indicates a better model, hence, the objective of optimization is to minimize this loss function. Furthermore, the loss function allows us to compare different models, serving as a metric to gauge their performance. However, defining a loss function can be challenging because it greatly depends on how well-defined the task and objectives are. The ideal loss function accurately reflects the priorities and requirements of the model's task.

The final component is the optimizer, which outlines the strategy to improve the model following the defined loss. The goal of an optimizer is to adjust the parameters of the model such that the loss is minimized. The space of optimization is defined by the tunable parameters within the model, forming a multidimensional landscape where each point represents a specific configuration of parameters. The optimizer navigates this landscape guided by the loss function. It iteratively updates the model parameters in a direction that leads a minimization of the loss function. By doing so, it improves the performance of the model until it reaches convergence.

1.2.2 The rise of deep learning

The impressive progress and popularity of deep learning in the world of artificial intelligence are attributed to several critical theoretical and technical advancements[16]. One of the fundamental breakthroughs in deep learning was the discovery of backpropagation. This algorithm, which is used during training, efficiently computes the gradient of the loss function with respect to the weights in the network. This concept was further advanced with the development of programming frameworks that can automatically compute gradients. These frameworks, such as PyTorch[17] and TensorFlow[18], store computational graphs representing the operations performed during forward propagation, as illustrated in Figure 1.3.

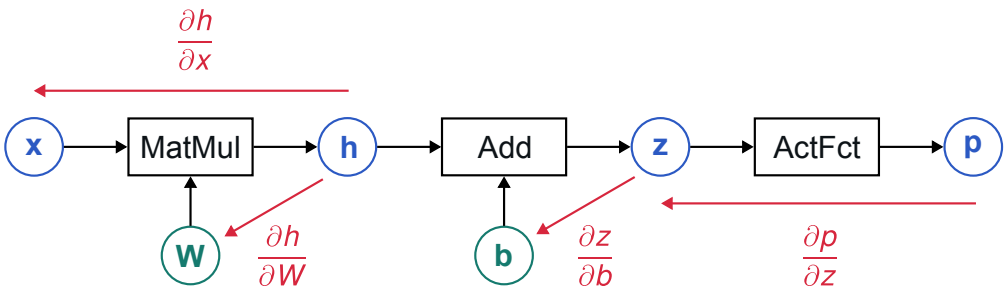

Figure 1.3: Illustration of the computational graph of a single layer neural network. The input x undergoes a series of operations, namely matrix multiplication (MatMul), addition (Add), and a non-linear activation function (ActFct), to produce the output or prediction p. Intermediate outputs h and z are referred to as hidden states. The learnable parameters of the model are W (weights) and b (bias). To enable backpropagation, all operations must be differentiable. Parameters are optimized using gradient descent, leveraging the chain rule for gradient computation in an algorithm called backpropagation[15].

Moreover, these computational graphs can be compiled and optimized for better performance. The power of deep learning models comes from their ability to learn complex relationships, but this requires substantial computational resources. The ability to optimize these computations for specific hardware has proven to be a major contributor to the success of deep learning. Graphical processing units (GPUs) was the perfect catalyst for the deep learning revolution. Initially designed to handle the computational demands of video games, GPUs have found a new purpose in powering the intensive calculations required by deep learning models. The parallel processing capabilities of GPUs are particularly well suited to the matrix and vector operations that are at the core of deep learning[19].

In conclusion, these key elements, backpropagation, automatic computation of gradients through stored computational graphs, the ability to compile and optimize these graphs and the use of GPU computing, have all been critical to accelerate the development, research and applications of deep learning. Each of these milestones enabled to make deep learning the powerful tool that it is today.

1.2.3 Designing neural networks architecture

When designing neural network architectures, several choices need to be made to build a model that will perform the specific task effectively. These decisions include identifying what aspects the model should learn directly from the data, and what aspects should be guided by the expert knowledge of the designer. The type of model chosen for the task will depend on the problem at hand, and might involve different flavors of neural networks, which will be discussed later. An equally critical consideration is the selection of a suitable loss function, a process often termed "loss engineering", which sets the benchmark for model performance. Finally, the optimizer needs to be chosen, which will guide the way the model adjusts its parameters to improve performance.

Neural networks, the cornerstone of deep learning, can be seen as an extension of linear models. They chain together multiple linear operations with non-linear activation functions to transform input data into a desired output. The parameters of these models represent the "knowledge" they have learned about the mapping from inputs to outputs. Single-layer neural networks, or perceptrons, are universal approximation function, given enough parameters. However, deep learning models opt for multiple layers to provide increased complexity and flexibility in the mapping[15].

Multiple layers in a neural network model constrain it to learn the desired output progressively. In other words, the model learns the process instead of memorizing the data. For instance, the analysis of the weights in a convolutional neural network (CNN) reveals how the network learns to detect edges, progressively evolving towards recognizing more complex and abstract shapes. The memorization is instead within the type of shape the model has to detect for a given task[16].

In the realm of deep learning, the choice of optimizer is somewhat constrained due to computational costs. Most optimizers used are variants of the Stochastic Gradient Descent (SGD), with the Adaptive Moment Estimation (Adam)[20] being one of the most popular and well-tested choices. In essence, the optimizer determines how the model will navigate the landscape of its parameters, seeking the path that will minimize the loss function and enhance its predictive performance.

1.2.4 Flavors of neural networks

Neural networks, in their underlying concept, are deceptively simple yet remarkably powerful. This strength stems from a combination of linear operations with non-linear activations, all linked together in a chain. The way these operations are assembled opens up an array of different neural network methods, each with its unique domain of application. The multitude of neural network types can be understood from a data structure perspective. The specific transformations and manipulations performed on the data within these networks are directly influenced by the architecture of the network itself, as illustrated in Figure 1.4. The properties of the data structure also contain information and assumptions on how quantities are connected.

Classical neural networks, also known as multilayer perceptrons[21] (MLPs), view data in a straightforward manner. An MLP with N inputs and M outputs treats each of the N channels as independent sources of data, and the M channels as independent data outputs. However, this simple approach is not always adequate for complex data types like images. Convolutional neural networks[22] (CNNs) are specifically designed to process grid-like data, such as images, where spatial relationships are crucial. Images, viewed as a 2D grid of pixels, retain specific patterns and structures that get lost if the image is transformed into a 1D array or shuffled. CNNs process information about neighboring pixels using parameter kernels and convolutions, and are specifically tuned for image data, making them more efficient at han-

dling this type of data[16].

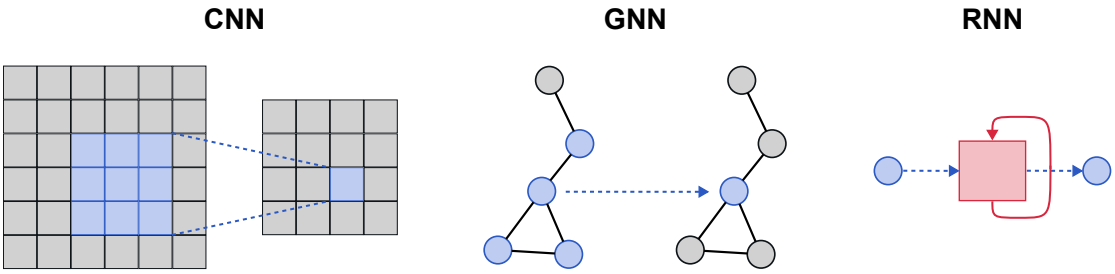

Figure 1.4: Illustration of different types of neural networks. Convolutional neural networks (CNN) are optimized for image data, utilizing convolution operations to capture local patterns in the data. Graph neural networks (GNN) work on graph-structured data, accommodating heterogeneous relationships within the data. Recurrent neural networks (RNN) maintain an internal memory-like state, making them suitable for time series data with an intrinsic direction.

Graph neural networks (GNNs) are a class of machine learning algorithms designed to manage complex data structures where the relationship between elements can be more diverse than simple linear sequences or two-dimensional grids. They are particularly efficient in capturing global properties of the entire graph, such as community structure, or local properties like the roles and groups of individual nodes. This ability to understand and represent intricate relationships within complex networks is central to their utility. One key domain where these models come into play is in working with graph-like structures, such as molecules in chemistry or biology, where atoms are nodes and bonds are edges[23].

Recurrent neural networks (RNNs), including Long Short-Term Memory (LSTM)[24] and Gated Recurrent Unit (GRU)[25], are particularly adept at processing sequential data. These networks are designed with the inherent assumption that the data is organized in a one-dimensional array of variable length and ordering direction. Their unique architecture allows them to retain information from previous elements in the sequence and use this to influence the processing of the next elements. This memory-like characteristic is what enables them to handle complex temporal dynamics and dependencies. They shine especially when temporal relationships are crucial, making them perfect for applications like natural language processing or time-series analysis. This ordering could represent time in the case of a time-series, or the sequence of words in a sentence for natural language processing[16].

In order to tackle the intricate task of managing long-range dependencies in text, the Transformer architecture was introduced. This powerful model was specifically designed to overcome the limitations of traditional recurrent models when it comes to processing sequences with long-term dependencies. At the core of the Transformer architecture is the attention mechanism[26], which dynamically assigns different weights to different words in a sequence, as illustrated in Figure 1.5. This implies that while processing a specific word, the model can take into account all other words in the sequence simultaneously. The result is a model that is not only more efficient at handling longer sequences but also better at

capturing the intricate relationships within them. As a result of these innovative features, Transformer models have significantly improved performance on tasks like language translation and text generation, where understanding the context from all parts of the sequence is key. This has effectively broadened the horizons of natural language processing and machine learning as a whole[27, 28].

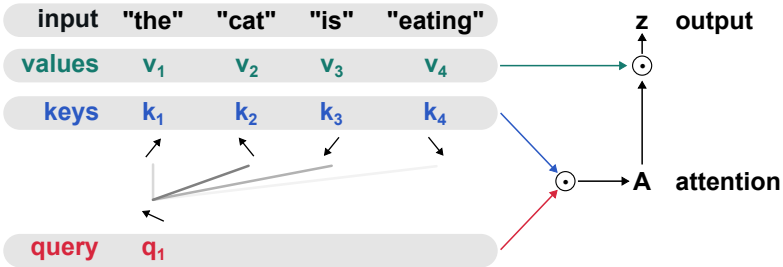

Figure 1.5: Illustration of the attention mechanism. From each word in a sentence, a value and key are encoded usually using a neural network. Similarly, one or multiple queries can be encoded from the same word in the case of the self-attention or from an external context. The key represents the embedding of a word that can be retrieved by a query when required. Keys and queries can be viewed as vectors in a high dimensional space; the attention put on a word is related to the cosine similarity of the key and query. For instance, in this example, if the query requires the subject of the sentence, more attention will be put on "cat" than on the other words.

In conclusion, neural networks can be tailored to suit specific tasks, making them versatile tools in the realm of machine learning. The design of these networks requires a certain level of expert knowledge to make correct assumptions about the task and the nature of the data. Once the framework is defined, the model can be trained and optimized to perform the task efficiently. Instead of making assumptions about the underlying model describing the data (e.g., linear, exponential), we can think more generally in terms of data structure, symmetry, and relationship between variables.

1.3 Deep Learning in Structural Biology

Deep learning has made considerable strides in structural biology, driving progress in areas that were previously challenging. The application of deep learning to this field arises from the recognition that protein structures exhibit patterns that can be learned and predicted by deep learning models. Existing deep learning approaches have brought us closer to understanding the complexity of protein structures. These methods range from predicting secondary structure, solvent accessibility, contact maps, to the tertiary structure of proteins.

The use of machine learning methods was accelerated by the availability, quantity and quality of datasets. The RCSB PDB contains more than 200'000 structures[29] and UniProt[30] contains just under 250 millions sequences as of August 2023. Given the rapid advancements in structure determination techniques, such as cryo-EM, the foldome is expected to expand significantly in the coming years. This database has driven the development of many deep

learning methods, most notably protein structure prediction methods such as AlphaFold[12].

1.3.1 Key concepts

Protein structure representation

Protein structures can be represented in multiple ways: surfaces describing electrochemical properties, volumes using voxels of densities, graphs of connected atoms or point clouds, as illustrated in Figure 1.6. The choice of a representation for a specific problem is important as it will affect the overall effectiveness of a machine learning model[31].

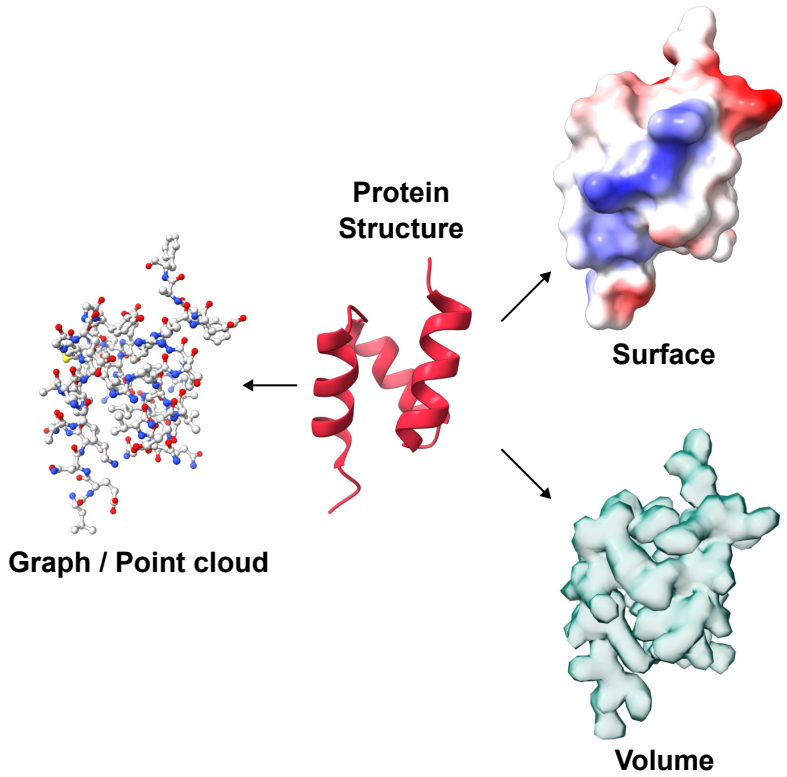

Figure 1.6: Illustration of various protein structure representations. A graph or point cloud of atoms provides the most detailed view, capturing both connectivity and spatial distribution of atoms. The surface representation depicts the parts of the structure that are accessible to a solvent. The volume representation uses image-like voxel-based segmentation to show density within the structure.

Surface representations of proteins provide a view of the electrochemical properties. This approach maps the topology of the protein, displaying the ridges, pockets, and crevices that play a vital role in biological interactions. By highlighting the areas of different charges and polarities, it can be useful in understanding protein-protein interactions, ligand docking, and overall protein function.

Volume-based representations make use of voxels to depict protein densities. This is akin

to a 3D version of pixels, where each voxel contains information about the protein structure within that small portion of space. This volumetric representation allows for a detailed rendering of the protein's interior and exterior, thereby providing a comprehensive 3D understanding of its structure.

Graph-based representations of proteins turn the attention to the interconnected nature of atoms within the molecule. In this approach, each atom is considered a node, and the bonds connecting them are represented as edges. This format is particularly suited to graph neural networks, offering a powerful approach to explore the relational information embedded in the protein structure.

Finally, point cloud representations view proteins as a collection of points in a 3D space, each point corresponding to an atom or a specific part of an amino acid. This method is advantageous for its simplicity and ability to handle large structures efficiently. While it lacks explicit connection information unlike the graph representation, it captures the spatial distribution and local structural motifs, making it a valuable tool in protein structure analysis.

Choosing the right representation for a specific problem in protein structure analysis is a crucial step. It should align with the nature of the problem, the available computational resources, and the assumptions made about the data for the machine learning model to perform optimally.

Geometric deep learning: importance of symmetry

In essence, geometric deep learning encompasses a broad spectrum of methods that seek to expand the traditional scope of neural networks by accounting for geometric and spatial information. These advancements have enabled deeper insights and more accurate predictions in numerous fields where complex geometric relationships play a central role[32]

Scalar quantities of proteins such as the energy or interactions interfaces are intrinsically independent of the choice of origin for the coordinate system. We say that these quantities are translation and rotation invariant. Vectorial quantities such as forces or velocities are invariant to translation but have to transform with the rotation of the reference frame making these quantities equivariant to rotation, as illustrated in Figure 1.7.

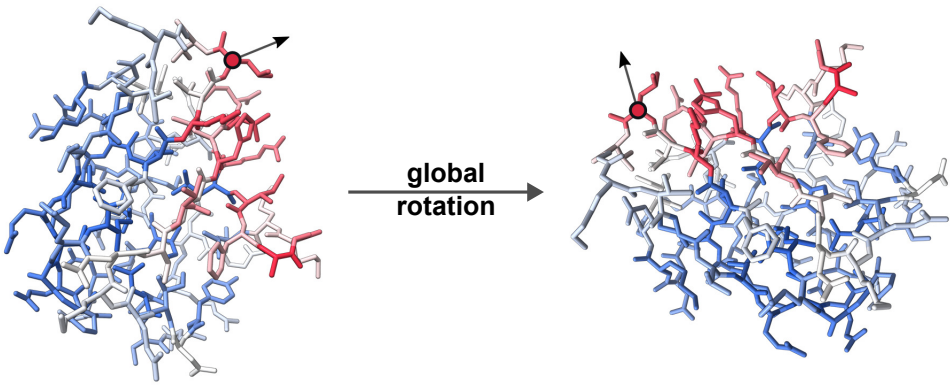

Figure 1.7: Illustration of invariant and equivariant properties of a protein structure. Certain quantities within a protein structure are either invariant or equivariant when the system undergoes a global rotation. Specifically, the identity of the amino acids at the interface, highlighted in red, remains the same despite rotation, making them invariant. On the other hand, vector quantities, as shown in the illustration, transform according to the applied global rotation, making them rotation equivariant.

Volumetric representation of proteins uses voxels to discretize the structure into density based features [33, 34]. The discretization requires a choice of origin and spacing for the grid of voxels. For each origin, new feature voxels are generated. It means that this representation is not rotation invariant. The translation invariance can be acquired by setting the origin at the center of mass of the structure for example. Because quantities which are interesting to predict are translation and rotation invariant, the model using voxels as input features has to learn a rotation and translation invariant mapping. In order to learn this mapping, the model is usually trained on many rotated representations, increasing the training time of the model. Moreover, to guarantee invariance, the model has to be trained on all rotations which is not tractable.

Respecting the symmetry of the system is especially important for predicting quantum interactions accurately as shown by multiple methods for predicting energy and forces at the quantum mechanical level. The ANI-1 [35] neural network uses Behler and Parrinello symmetry functions to encode angular features and the SchNet [36] uses continuous-filter convolutions. The energy predicted are translation and rotation invariant and the forces prediction are rotationally equivariant.

Alternatively, the representation of protein structures can be intrinsically translation and rotation invariant. For instance, representing molecules and protein structures as graphs is independent of the reference frame. Atoms or residues represent nodes and bonds or interactions are represented as edges. Graph representation of protein structures has been successfully applied to the folding quality assessment [37].

One of the most successful approach is to represent protein structures as point cloud. The most common approach is to define rotation equivariant convolution operations based

spherical harmonic as introduced by the Tensor Field Networks[38] (TFN). Spherical harmonics are commonly used in Quantum Mechanics for their ability to describe the wavefunctions of electrons in atoms. The Cormorant networks[39] further improved this idea with Clebsch-Gordan non-linearity enhancing the degrees of freedom of the model. TFN architectures have been successful to predict quality of generated protein-protein complexes[40].

Instead of using spherical harmonics, a simpler approach is to define operations applied directly to vectors and respecting rotation equivariance. The geometric vector perceptron[41] (GVP) uses linear operations to compose vector features with gating[42]. Graph neural networks have been extended to equivariant graph neural network [43].

Transformers

The introduction of transformers showed that recurrent neural networks are not necessary to process variable size input: attention is all you need[26]. Transformers couple dynamic input dimension to an arbitrary output dimension using the keys, queries, values principle. The values are direct or encoded representation of the dynamic input. The keys are a low dimensional embedding of the values used as fingerprint for the matching values. The queries have an arbitrary size of the desired output dimension. They are matched with the keys using dot-product attention in order to retrieve the corresponding values. The attention mechanism allows the model to dynamically filter information based on the queries, as illustrated in Figure 1.5. Transformers are heavily used in natural language processing[44].

The extension of the TFN with an attention mechanism on the message passing leads to the SE(3)-Transformers[45]. It enables the modulation of angular information through the spherical harmonics. Many successful methods combine transformers and geometric deep learning. The major breakthroughs come from the field of protein structure prediction. AlphaFold2[12] integrates attention in the Evoformer blocks and the structure module. The third track of the RoseTTAFold[46] model uses a SE(3)-Transformer to refine the atom coordinates during folding. The recurrent geometric network[47] (RGN2) leverages the Frenet-Serret formulas to represent the backbone of proteins.

1.3.2 Highlighted methods and applications

Protein structure prediction

Protein structure prediction was long considered a computationally complex problem, particularly for proteins without known homologues. Traditional methods struggled to offer accurate predictions for these types of proteins. However, the landscape of computational protein structure prediction underwent a significant transformation with the advent of deep learning technologies, most notably AlphaFold[12], we specifically refer to its latest version, formerly known as AlphaFold2. AlphaFold was further adapted into AlphaFold-multimer for predicting the structure of protein complexes[48].

The efficacy of AlphaFold is heavily dependent on the availability of both structural and sequence data. A substantial part of this sequence data comes from UniProt[30], a database that aggregates and cross-references protein sequences with various sources of experimental evidences, providing an enriched understanding of a protein's characteristics. For the problem of protein structure prediction, the sequences are particularly useful when they are either functionally similar or evolutionarily related. These sequences offer valuable insights, such as the evolutionary coupling of amino acids at contact points within the protein structure.

AlphaFold employs deep learning a complex architecture featuring multiple tracks to extract the relevant information from sequences and predict the 3D structure of the protein folds. One of the key components is the Evoformer block, which is responsible for processing multiple sequence alignments (MSA) and pair representations. Another critical module is the structure module, which utilizes an invariant point attention mechanism. This module generates a protein structure based on the processed MSA and pair representations.

Following the success of AlphaFold, several other deep learning-based methods have emerged in the field. Notable among these are RoseTTAFold[46] and ESMFold[49], which have also contributed to advancements in protein structure prediction prediction.

Protein-protein interfaces prediction

Most biological functions are fulfilled by protein complexes. In these complexes, the proteins interact with each other via interfaces. These interactions can be either stable, which is mostly the case for structural components or very transient as in the case of signal transductions. The strength of these interactions can also be modulated by post-translational modifications that affect either binding or the structure of the proteins. Being able to predict the interaction for two proteins would be useful either to help to understand their function, to manipulate their interfaces in order to alter their interactions, or finally to understand mechanistic aspects. Moreover, being able to predict a docking interface is an essential step towards the design of inhibitors.

Multiple machine learning based protein-protein interaction site prediction methods have been developed. The idea that fingerprints of protein-protein interactions surfaces can be used to predict interaction sites was introduced by the SPPIDER[50] model. The method combines features at the residue level with relative solvent accessibility. Since the structure of the protein is not always known and methods such as PSIVER[51] predicts interaction site based on the sequence. Finally, IntPred[52] tried to improve upon the existing methods with random forest algorithm using surface patches and more features.

The molecular surface interaction fingerprinting (MaSIF)[14, 53] method uses geometric deep learning on surface patches. In this method, proteins are described with geometric and chemical features of their surfaces. The deep learning model performs convolution operations on surface patches to obtain learned descriptors that can be used for multiple tasks. For

instance, MaSIF-site uses the fingerprint descriptors to predict protein-protein interfaces.

Inverse folding problem

The creation of proteins de novo, in order to engineer their properties for specific tasks is a massive undertaking. This endeavor has far-reaching implications for various fields including biology, medicine, biotechnology, and materials science. Traditional physics-based approaches have shown promise in identifying the amino acid sequences required to fold into a given protein structure. However, recent developments have witnessed a new player taking center stage in this field: deep learning.

Deep learning methods have catalyzed a significant advancement in protein design, enhancing both the success rates and the versatility of the design process[54]. These modern techniques are steadily redefining the landscape of protein design and significantly boosting the effectiveness of the process.

ProteinMPNN[13] is a particularly noteworthy example of the recent successes of deep learning. This tool leverages an encoder-decoder neural network to generate protein sequences. Impressively, these sequences have been proven through experimental validation to fold as intended. Furthermore, when coupled with denoising diffusion probabilistic models used for generating protein backbones, ProteinMPNN has shown significant success, as evidenced in its use within the RFdiffusion[55] method.

In addition to ProteinMPNN, ESM-IF1[56] is another outstanding model. Based on a protein language model, it is capable of generating a wide range of proteins that extend well beyond the known universe of natural sequences. Importantly, the model has not only proven effective in theory, but has also been experimentally validated, demonstrating a high success rate.

These examples represent just a fraction of the potential deep learning holds in the realm of protein design. Some models tackle specific tasks, such as the design of protein interacting peptides, like in the case of MaSIF[14] which specializes in protein surface fingerprints, to a host of other protein design tasks, deep learning techniques have found extensive application and are shaping the future of protein design[57]. Given the current trajectory, the integration of deep learning in the field of protein design promises to yield even more exciting developments in the future.

1.4 Research Aims

The first and primary aim of my thesis is the development and optimization of a deep learning method capable of accurately predicting protein-protein binding interfaces. This method also extends to detect interactions involving other molecules, such as nucleic acids, ligands, ions, and lipids, to offer a comprehensive understanding of protein interactions.

The second aim is to explore the potential applications of the developed method, particularly its utility in analyzing large ensembles of structural data and molecular dynamics simulations. This allows for the creation of an ensemble of predicted binding interfaces that can assist in identifying proteins of interest for various research applications.

The third aim to adapt the capabilities of the method to a different, but equally challenging problem in the domain of protein design: predicting amino acid sequences from protein backbone scaffolds, including those with non-protein atoms. The ability to consider non-protein entities in sequence prediction enriches the field of protein design, opening up new possibilities in therapeutics and biotechnology.

Altogether, these aims collectively contribute to expanding the toolset available for structural biology research, offering a method that not only excels in predicting protein interactions, but also paves the way for innovative applications in sequence prediction and large-scale structural analysis.

2 Methods

In this chapter, I will first discuss the dataset used for training our deep learning model. Notably, our focus is on structural data, as opposed to sequence data. The emphasis on structural data allows us to capture higher-order interactions and geometrical properties that are often not apparent or easily inferable from sequence data alone. This offers a more comprehensive understanding of the function, stability, and interactions of a protein with other molecules such as nucleic acids, ions, ligands or lipids. Moreover, by focusing on structural data, we are in a better position to model accurately the underlying physical laws that govern protein folding and protein-protein interactions.

Next, I will discuss the specific implementation choices made to make this project feasible. Given our focus on structural data, these choices have been tailored to effectively handle and process complex geometrical and topological information. Following this, I will outline the new deep learning operations and architectures I developed specifically for handling structural data. This section will offer insights into how these methods were designed and their unique capabilities in the context of structural analysis. Then, I will describe the training and evaluation protocols that I employed. This includes detailing how I optimized and validated our models, as well as the metrics used for assessment. Finally, I will elaborate on the various applications where our newly developed methods have been applied. These applications will serve as practical examples of use case of our approach.

2.1 Dataset

2.1.1 Protein structure database

Within the realm of deep learning, input data is central to the training process. One of the main resources for this data, especially when working with proteins, is the Protein Data Bank (PDB). This comprehensive database contains a vast collection of experimentally resolved protein structures. These structures are determined using various experimental techniques, such as X-ray crystallography, NMR spectroscopy, and cryo-electron microscopy.

PDB contains mainly structures obtained by X-ray crystallography and are hence in a non-native crystal cell structure. This representation does not always reflect the biologically accurate interactions of proteins since in its native environment, the protein might interact differently than in a crystal lattice. Luckily, the PDB also offers manually curated biological assemblies. These structures contain the biologically relevant assembly of biomolecules. This is especially relevant to give a more accurate representation of a complex structure, like the full capsid of a viral protein.

Additionally, the PDB is not limited to just protein structures. It contains many protein structures with other essential biomolecules like DNA, RNA, and a myriad of small molecules. This includes ligands, lipids, and carbohydrates, further broadening the scope of data available for deep learning endeavors in structural biology.

2.1.2 Data processing pipeline

Implementation choices

Our primary data source is the Protein Data Bank (PDB). From this repository, I acquired all biological assemblies, often referred to as bioassemblies. I chose the PDB[29] format primarily due to its ease in parsing and reading when compared to other alternatives like mmCIF[58, 59]. Handling this high quantity of data necessitated specific tools. Hence, I implemented, in Python, a data processing pipeline that allowed us to efficiently process and store these protein structures.

For parsing PDB files, I utilized the Gemmi[60] parser. As for the additional processing tasks, I employed the functionalities of both Numpy[61] and PyTorch[17], a powerful tool for numerical computing in Python, integrates seamlessly with PyTorch, a leading deep learning framework. This synergy ensures a smooth transition of data structures and operations between the two libraries, eliminating the cumbersome conversion processes that can sometimes plague data workflows. Furthermore, the capabilities of PyTorch are not just limited to deep learning. It is equipped to run heavy data processing computations, and what sets it apart is its innate ability to leverage GPU resources. By offloading complex calculations to the GPU, PyTorch ensures rapid data processing, vastly reducing the time required for computationally intensive tasks. This capability is invaluable, especially when dealing with large datasets like the ones encountered in our project, making PyTorch an optimal choice for our processing needs.

Our data processing efforts also involved the selection of an appropriate storage medium. Given the vastness and complexity of our protein structures data, I opted for the hierarchical data format version 5 (HDF5)[62], which allowed us to encapsulate all our structural data within a singular, cohesive dataset. The HDF5 format brings to the table a hierarchical data structure that resembles a conventional file system in its organization. This structure is advantageous for various reasons. Firstly, it ensures a streamlined data arrangement, facilitat-

ing easy data retrieval and manipulation. Furthermore, HDF5 has been optimized for high-throughput during read operations. This means that the time taken to access and read data is significantly minimized. Notably, the format is adept at handling parallel read operations, making it particularly useful for large-scale, simultaneous data access requirements. This combination of structural clarity and operational efficiency made HDF5 the ideal choice for our data storage needs.

Representation of the protein structures demanded a balance of simplicity and performance. In response, I depicted these structures using a Python dictionary. Within this dictionary, various fields such as coordinates, atom name, and residue name, to name a few, correspond to Numpy arrays. These arrays store the associated values for every atom present. This design choice offered us two primary advantages: it eliminated the need for extensive tooling and was tailor-made for efficient array slicing. An added bonus was the capability to transition these Numpy arrays into PyTorch arrays with minimal effort. Our framework is not just restricted to static representations. It also supported dynamic data, like that from molecular dynamics. The format I used can effortlessly incorporate such data by extending the coordinates array with a time dimension.

Curating the dataset

In the process of curating our dataset, a few key decisions and adjustments were imperative to ensure the consistency and utility of the data for our specific needs. Even with a curated database like the PDB, raw protein structures often require additional preprocessing to make them suitable for computational methods.

One of the first decisions was to omit experimentally resolved hydrogens from our dataset. This decision was driven by two main considerations. First, if needed, hydrogens can be added back into the protein structures with relative ease. More importantly, the presence of certain hydrogens is contingent upon the protonation state of specific amino acids. This state is, in turn, influenced by the pH of the solvent, reflecting the specific environmental conditions under which the experiment was conducted. Given these factors, excluding hydrogens was a logical choice for our purposes.

Another significant processing step involved standardizing residue numbering. In the raw PDB data, residue numbering can often be inconsistent due to factors like missing structure segments, deletions, insertions, and the presence of multiple chains or protein subunits. To bring uniformity to our dataset, I opted to renumber all residues, starting from 1, disregarding any gaps, insertions, or other anomalies.

A frequent inconsistency encountered in the PDB entries is the chain naming of non-protein molecules, commonly referred to as hetero atoms. These hetero atoms, which can include small molecules, often share the same chain name as proteins, despite being distinct entities. To address this, I introduced a tagging system for the chain names of all hetero atoms. This

system allows us to efficiently identify and segregate the various molecules present within a given structure, ensuring clarity and precision in our curated dataset.

Lastly, protein structures and biological assemblies can also have structural inconsistencies. For instance, I removed the duplicated subunits, molecules, and ions generated when concatenating multiple models within the bioassembly. Moreover, I only kept the first alternate location of the atoms, as it is most likely to represent the best quality fit.

Structures derived from the database occasionally include water molecules, especially those resolved through crystallography. Given the nature of our project, I opted to discard these water molecules during our general preprocessing. While these water molecules can play essential roles in the structural integrity and biological activity of the protein, for many computational tasks, their presence can introduce unnecessary complications. For instance, voids in the shell of resolved water molecules would trivially indicate binding interfaces. As a note, I did carry some experimentation with predicting protein-water interacting surface. The model showed potential to predict accurately strong water binding sites. However, since I did not find a relevant application for model, I did not pursue this direction.

Feature engineering

Prioritizing simplicity, our primary objective was to capture the essential details required to represent a protein from a structural point of view. I settled on atom coordinates for capturing geometry and element names for chemical identity: simplicity without reduction of the full description of a protein. All other features such as atom name, amino acid type and secondary structures can be recovered from the atom coordinates and element names, as depicted in Figure 2.1a. This method represents proteins as a point cloud of atoms, a broad representation applicable not only to proteins but also all biomolecules in general.

Instead of incorporating pre-selected features, I trusted the capacity of the model to discern and grasp key details directly from the raw structure. This holistic approach ensures no data gaps. Given the abundant data on hand, this direct approach is viable. However, for smaller datasets, it is usually advisable to use a curated set of features.

Understanding the symmetries of a system is essential as they shed light on the non-critical parameters and the intrinsic properties of that system. For instance, globally translating or rotating a protein structure does not impact key aspects such as its sequence, interaction interface, or thermal stability, as illustrated in Figure 1.7. While models can be trained to recognize these system symmetries, it demands a detailed sampling within this symmetric space. Specifically, I would need to sample a range of rotations and translations. However, there is an alternative approach. Instead of training the model through sampling, I can design it to be naturally insensitive to these transformations. This can be achieved either by focusing on invariant input features or by making the model processing the information in a way that remains consistent, regardless of the transformations applied to the system.

Drawing from the Protein Data Bank (PDB)[29], I extracted the 30 most frequently occurring atomic elements. I represented the element name using a one-hot encoding. To capture the spatial relationships between atoms, I utilized pair-wise distance matrices, ensuring translational and rotational symmetry. Additionally, guided by the principle of simplicity, I used vectorial quantities to represent the geometric context, as shown in Figure 2.1b. Therefore, I employed a normalized displacement vector tensor, offering a geometric perspective on the relative positioning of atoms. To effectively handle these quantities along with scalar features, I introduced new neural network based operations tailored for this purpose.

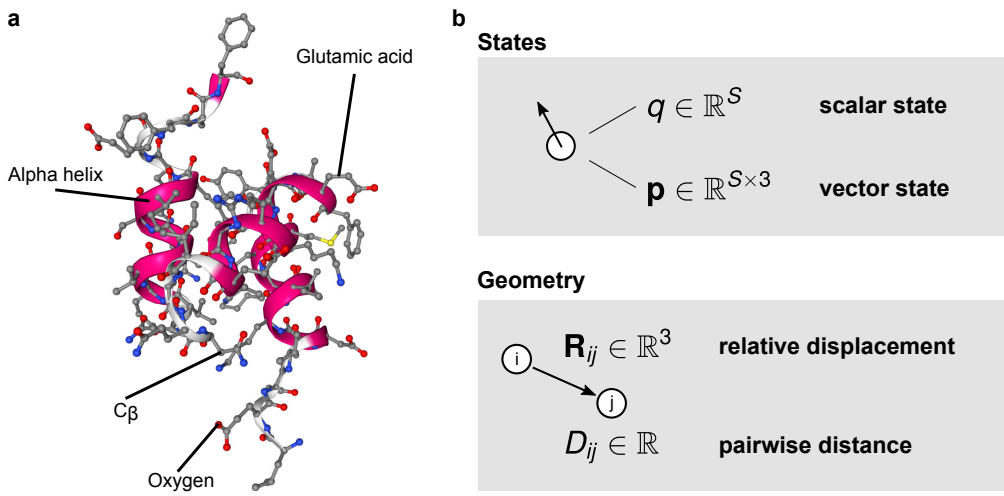

Figure 2.1: Features selection for protein structure. (**a**) Example of possible information to describe a structure, such as the secondary structure, the amino acid type, the atom name based on the chemical context or the atom element. (**b**) The chosen point cloud description of a structure. Each atom is associated with a scalar and vector state. The geometry of the structure is described using relative displacement and pairwise distances.

2.2 Protein Structure Transformer (PeSTo)

I introduce here a new geometrical transformer operation acting on protein structures at the atomistic level. The structure is represented as a point cloud of atoms. The geometry of the structure is described with pairwise distances and relative displacement vectors which guarantee the translation invariance. A scalar state and a vector state is assigned to each point. The geometrical transformer is a rotation equivariant operations updating the states of each atom using the states in the local neighborhood. This approach is computationally inexpensive for preprocessing of the structures compared to other representations such as surfaces and volumes. The method does not rely on any features requiring to have clean or well-defined atomic structures such as electrostatics. Therefore, it can be applied on any atomic structure and allow us to use all protein structures available for training.

2.2.1 Geometric transformer

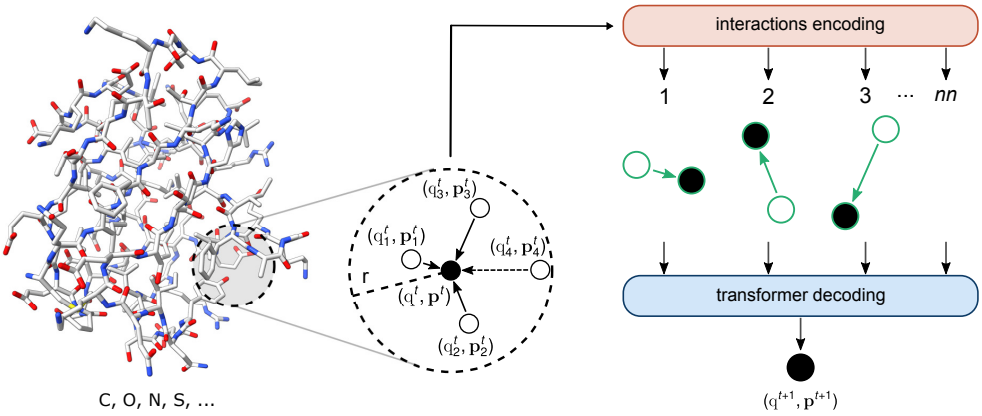

Figure 2.2: Geometric transformer workflow. For each atom, a local neighborhood is extracted. Within these neighborhoods, the geometric transformer encodes interactions with the center atom using scalar states, vector states, and the geometry. These encoded interactions are then aggregated through a transformer. The attention mechanism uses the interactions to define the values and keys and the state of the center atom defines the queries. Importantly, this transformer-based message passing operation is independent of the interaction order and can dynamically adapt to various neighborhood sizes.

At the core of the PeSTo architecture is the geometric transformer (G), see Figure 2.2 and Algorithm 1. This key operation updates the state of each atom by considering the local geometry and the state of atoms within a predefined neighborhood, defined by a set of nearest neighbors (nn). The state of each atom is represented by a scalar state (q) and a vector state (p), while the geometry is characterized by the pairwise distances (D) and normalized displacement vectors (R). In the PeSTo architecture, each layer (l) of geometric transformer processes and propagates the scalar, vector and geometrical information of the structure as described in Equation 2.1.

$$q_i^{l+1}, p_i^{l+1} = G(q_i^l, p_i^l, \{q_j^l, p_j^l, D_{ij}, R_{ij}\}_{j \in nn_i}) \tag{2.1}$$

The geometric transformer leverages the attention mechanism based on the queries, keys and values approach[26] as described in Equations 2.2 & 2.3. The queries for the scalar and vectorial tracks (Q_q, Q_p) are derived from the state of the central atom i (q_i, p_i). The keys (K) are encoded from the interactions of the central atom i with its neighboring atoms j, encapsulating the states of the central atom, the neighboring atom, and their spatial relation (q_{il} , p_{il} , $\{q_{jl}, p_{jl}, D_{ij}, R_{ij}\}_{j \in nn_i}$). Scalar value vectors (V_q) and vector value vectors (V_p) are respectively extracted from the computed scalar and vector quantities of these interactions. The transformer allows a flexible linear composition of the vector features and states such that the resulting vector state is equivariant to a rotation of the input vector. The attention is done over multiple heads and projected using learned weights for the scalar and vector tracks

(W_{ql}, W_{pl}).

$$q_i^{l+1} = \text{Attention}(Q_q^l, K^l, V_q^l) W_q^l \tag{2.2}$$

$$p_i^{l+1} = \text{Attention}(Q_p^l, K^l, V_p^l) W_p^l \tag{2.3}$$

Each geometric transformer is composed of 5 neural networks of 3 layers with an exponential linear unit (ELU) activation function. The characteristic dimensions are the number of atoms (N), the state size (S), the number of nearest neighbors (nn), the dimension of the embedding for the keys (N_k) and the number of attention heads (N_h). The neural networks have a flat architecture with hidden layers width equal to the input and output state size (S). The multi-layers perceptrons (MLP) are the node query model (f_{nqm}), encoding scalar key model (f_{eqkm}), encoding vector key model (f_{epkm}), encoding value model (f_{evm}), and scalar state projection model (f_{qpm}). The vectorial hidden state is projected over the attention heads with a weighted sum (W_{ppm}) to preserve the rotation equivariance of the operation. The output vector state belongs to the span of the geometry and vector states.

The geometric transformer is translation invariant, rotation equivariant and independent of the order of the atoms and order of the interactions. The attention operation allows for a dynamic number of nearest neighbors (nn). However, in practice, the operation is much more computationally efficient with fixed number of nearest neighbors. For structures with a number of atoms smaller than the set number of nearest neighbors, the additional non-existent interactions are sent to a sink node with a scalar and vector state set to zero. The residual connection provides a way for the gradient to flow better for deep neural networks and allows for a gradual update of the state.

Algorithm 1: Geometric transformer

Input:
Center node features: $q \in \mathbb{R}^{N\times S}, p \in \mathbb{R}^{N\times S\times 3}$
Context neighbors features: $q_{nn} \in \mathbb{R}^{N\times n\times S}, p_{nn} \in \mathbb{R}^{N\times n\times S\times 3}$
Geometry features: $d_{nn} \in \mathbb{R}^{N\times n}, r_{nn} \in \mathbb{R}^{N\times n\times 3}$

Output:
New state of center node: $q', \vec{p}'$

`// Node and edges features`
$X_n \leftarrow concat(q, \|\vec{p}\|) \in \mathbb{R}^{N\times 2S}$
$X_e \leftarrow concat(d_{nn}, q, \|\vec{p}\|, q_{nn}, \|\vec{p}_{nn}\|, \vec{p}\cdot\vec{r}_{nn}, \vec{p}_{nn}\cdot\vec{r}_{nn}) \in \mathbb{R}^{N\times n\times 6S+1}$

`// Queries from node state`
$Q_q, Q_p \leftarrow f_{nqm}(X_n) \in \mathbb{R}^{N\times N_h\times N_k} \times \mathbb{R}^{N\times N_h\times N_k}$

`// Keys from edges state`
$K_q \leftarrow f_{eqkm}(X_e) \in \mathbb{R}^{N\times n\times N_k}$
$K_p \leftarrow f_{epkm}(X_e) \in \mathbb{R}^{N\times 3n\times N_k}$

`// Values from edges state`
$V_q, V_p \leftarrow f_{evm}(X_e) \in \mathbb{R}^{N\times n\times S} \times \mathbb{R}^{N\times n\times S}$
$\vec{X}_g \leftarrow \text{concat}(V_p \odot \vec{r}_{nn}, \vec{p}, \vec{p}_{nn}) \in \mathbb{R}^{N\times 3n\times S\times 3}$

`// Scaled dot-product attention and projection`
$q_h \leftarrow f_{qpm}(\text{softmax}(\frac{Q_q K_q^T}{\sqrt{N_k}})V_q) \in \mathbb{R}^{N\times S}$
$\vec{p}_h \leftarrow W_{ppm}\text{softmax}(\frac{Q_p K_p^T}{\sqrt{N_k}})\vec{X}_g \in \mathbb{R}^{N\times S\times 3}$

`// Update state with residual`
$q' \leftarrow q + q_h$
$\vec{p}' \leftarrow \vec{p} + \vec{p}_h$

2.2.2 Geometric pooling

The geometric transformer focuses on updating the state of individual atoms without performing any reduction in the atom point cloud. However, in some applications, it becomes relevant to encode the state of a group of atoms. For instance, when predictions need to be made at the residue level rather than the atomic level. One of the challenges faced when trying to transition from atomic to residue-level predictions is the inherent variability in the number of atoms that constitute different residues. This variability means that achieving a consistent state size reduction is not straightforward. To address this, I introduced a self-attention based reduction method called Geometric pooling. This method leverages the

strengths of self-attention mechanisms to effectively reduce and combine the information from a variable number of atoms to encode a single state per group of atoms.

As described in Algorithm 2, the Geometric pooling is composed of 2 neural networks of 3 layers with an exponential linear unit (ELU) activation function. The characteristic dimensions are the number of atoms (N), the number of residues (N_r), the state size (S), the dimension of the embedding for the keys (N_k) and the number of attention heads (N_h). The neural networks have a flat architecture with hidden layers width equal to the input and output state size (S). The multi-layers perceptrons (MLP) are the self-attention model (f_{sam}) and residue scalar state projection model (f_{qpm}). The vectorial hidden state is projected with a weighted sum (W_{prpm}) to preserve the rotation equivariance of the operation.

Algorithm 2: Geometric pooling

Input:
Center node features: $q \in \mathbb{R}^{N \times S}, p \in \mathbb{R}^{N \times S \times 3}$
Atoms to residue map: $M \in \{0,1\}^{N \times N_r}$

Output:
States of residue nodes: $q_r, \vec{p}_r$

`// Node features and define atoms to residue attention filter`

1 $X_n \leftarrow concat(q, \|\vec{p}\|) \in \mathbb{R}^{N \times 2S}$

2 $F \leftarrow \dfrac{1 - M + \epsilon}{M - \epsilon} \in \mathbb{R}^{N \times N_r}$

`// Multi-heads residue-localized self-attention masks`

3 $Z_q, Z_p \leftarrow f_{sam}(X_n) \in \mathbb{R}^{N \times N_h} \times \mathbb{R}^{N \times N_h}$

4 $A_q \leftarrow \text{softmax}(Z_q + F) \in \mathbb{R}^{N \times N_r \times N_h}$

5 $A_p \leftarrow \text{softmax}(Z_p + F) \in \mathbb{R}^{N \times N_r \times N_h}$

`// Attention and projection to the input state`

6 $q_r \leftarrow f_{qrpm}(A_q q) \in \mathbb{R}^{N_r \times S}$

7 $\vec{p}_r \leftarrow W_{prpm} A_p \vec{p} \in \mathbb{R}^{N_r \times S \times 3}$

2.2.3 Translation invariance

The translation invariance is directly provided by the geometrical features used. The neighbors distances (d_{nn}) and normalized relative displacements (r_{nn}) are independent of the origin of the coordinate system. The distance is defined as $|x_i - x_j|$. The normalized relative displacement is defined as $r_{ij} = \frac{x_i - x_j}{|x_i - x_j|}$.

2.2.4 Rotation equivariance

In order to guarantee the rotation equivariance, all operations on vectors in PeSTo are a combination of three rotation invariant and equivariant operations: namely, the scalar product (invariant), the scalar multiplication (equivariant), and the linear combination of vectors (equivariant). It follows that the norm ($|\cdot|$), the element-wise product ($\odot$) and the projection from N to M vectors (i.e., PX with $X \in R^{N\times 3}$ a vector and $P \in R^{M\times N}$ a projection) are also rotation invariant, equivariant and equivariant, respectively.

Rotation equivariance of the geometric transformer

The packed nodes and edges features (X_n, X_e) are composed of rotation invariant features (d_{nn}, q, q_{nn}) and rotation invariant quantities derived from scalar product of (r_{nn}, p, p_{nn}).

The encoded geometric feature (X_g) is a vector quantity composed of the input vector states (p, p_{nn}) and element-wise scalar multiplication of the normalized relative displacement vectors (r_{nn}).

The vectorial hidden state (p_h) is obtained by two equivariant projections. First the attention is a projection from the $3n$ neighborhood vectorial features to Nh attention heads (equivariant). Then the W_{ppm} projects the states (S) for all heads (N_h) channels into the final output state of size S (equivariant).

The final output for the vector state (p') is obtained by linear combination of the previous state (p) and the vectorial hidden state (p_h)

Rotation equivariance of the geometric pooling

The packed node features (X_n) is composed of rotation invariant features (q) and rotation invariant quantities derived from scalar product of (p).

The final output for the vector state at the residue level (p_r) is obtained by two equivariant projections. First the attention is a projection from the N atoms to N_r residues vectorial features for N_h attention heads (equivariant). Then the W_{prpm} projects the states (S) for all heads (N_h) channels into the final output state of size S (equivariant).

Rotation invariance of output quantity

The input features of the multilayer perceptron of the interface model are the scalar state at the residue level (q_r) and the norm of the vector state at the residue level (p_r) both are rotation invariant.

2.3 Training and Evaluation

2.3.1 Training objective

In binary classification, a predictor classifies an input into one of two classes: negative (0) or positive (1). For neural networks, I need operations that are differentiable. This means that instead of directly predicting 0 or 1, the neural network produces a continuous value between 0 and 1, representing the likelihood of the input being positive. To ensure the output value stays between 0 and 1, I use the sigmoid function: $\sigma(x) = \frac{1}{1+e^{-x}}$. This function maps any real number to a value within the [0, 1] range, making it suitable for binary classification outputs.

For a continuous prediction $p \in [0,1]$ and a label $y \in \{0,1\}$, the appropriate loss function for this type of task is the binary cross entropy (BCE), see Equation 2.4.

$$l(p, y) = y\log(p) + (1-y)\log(1-p) \tag{2.4}$$

The continuous output from a model is often referred to as prediction confidence, indicating the certainty of a model in its decision. To classify this output as either positive or negative, a threshold is used. Typically, a value of 0.5 is the dividing boundary: if the prediction confidence is 0.5 or higher, it is considered a positive prediction; otherwise, it is viewed as negative.

2.3.2 Evaluation protocol

When training a model, it is important to properly curate both the training and testing datasets. The goal is to show the ability of the model to generalize to new, unknown data. To ensure this, I must prevent the training dataset from containing examples that are too similar to those in the testing set. Typically, I partition the dataset into three subsets: training, validation, and testing. The training set helps us build and refine our model. Meanwhile, the validation set is employed during training to monitor the progress of the model, detect overfitting, and determine when training should be stopped. Finally, the test set provides a means to evaluate the performance of the fully trained model.

In our case, to ensure the model is not memorizing specific patterns from similar structures, there should be low similarity between the training, validation, and testing datasets. This entails dividing our protein structures dataset into three distinct sets using specific criteria.

Sequence comparison

When comparing proteins, a common approach is to examine their sequences. Proteins can have sequences of different lengths due to variations in the protein size or because of insertions and deletions in the sequence. To address this, sequences are aligned to optimize

a specific alignment metric, ensuring accurate comparison. The primary measure used to determine the similarity of two sequences is sequence identity. This metric calculates the percentage of positions with the same amino acid in both sequences. However, simply using sequence identity can overlook the chemical similarities between different amino acids. To account for this, the BLOSUM[63] similarity matrix is used. This matrix weight the chemical similarities between amino acids, offering a more detailed view of a comparison between a pair of sequences.

Structure comparison

In protein comparison, besides sequence similarity, structural similarity presents a more challenging aspect. The C.A.T.H. classification is a commonly used method for comparing protein structures. This approach breaks down protein structures based on distinct attributes. At the Class (C) level, structures are categorized by their secondary structure content, distinguishing whether a protein is mainly helical, largely beta-stranded, or a combination of both. Next, the Architecture (A) level focuses on the spatial arrangement of these secondary structures, observing how they are positioned relative to each other. The Topology (T) level evaluates the connectivity between these secondary structures, identifying how various parts of the protein interact and relate. Lastly, the Homologous superfamily (H) level groups proteins based on their evolutionary relationships, ensuring structures with similar origins are grouped together.

In practice, following the protocol established by previous methods[14, 13], I set a sequence identity threshold of 30%. By using this criterion the training, validation, and testing datasets consist mostly of unrelated proteins. Additionally, when the protein fold is a significant factor, I incorporate structure similarity thresholds using the C.A.T.H. classification, adding another similarity constraint alongside sequence identity.

2.3.3 Assessing predictions

There are several scoring metrics available to evaluate the performance of binary classification models. These include accuracy (ACC), precision (PPV), negative predictive value (NPV), true positive rate (TPR), true negative rate (TNR), Matthews correlation coefficient (MCC), Receiver Operating Characteristic Area Under the Curve (ROC AUC), and Precision-Recall Area Under the Curve (PR AUC).

It is important to note that both ROC AUC and PR AUC stand out as threshold-free scores. This means that their values do not depend on a specific threshold used to define a positive and negative predictions from the prediction. Their main advantage is assessing the ranking ability of the model, making them particularly robust when evaluating models on imbalanced datasets. However, even these metrics can be influenced by dataset imbalances and should always be interpreted with the specific context of the dataset in mind. For instance,

the prevalence describes the ratio of positive examples in the dataset and gives a specific context to interpret all the metrics.

2.4 Applications

2.4.1 Protein binding interface prediction

One of the primary applications of the PeSTo architecture is predicting protein binding interfaces. The main goal is to develop a model capable of accurately identifying which amino acids within a protein structure are likely to interact with another biomolecule, as illustrated in Figure 2.3. A deeper aspect of this goal is to have the model discern interactions of a protein with other proteins, nucleic acids, ions, ligands, or lipids, predicting more details on the roles and means proteins have in various biological processes. The results related to this section can be found in Chapter 3.

To ensure the success of the model, I adopted a systematic approach. First, the dataset was selected and split to represent a wide range of proteins and their known binding interfaces. Input features were then chosen to capture essential information from the protein structures. The labels were defined based on the known interactions of the proteins. Using the Geometric transformer, I build, trained and tested various PeSTo architecture for the task of binding interface prediction. I carried out a thorough evaluation and comparison of the model to assess its performance. Finally, I experimented with various application of the model.

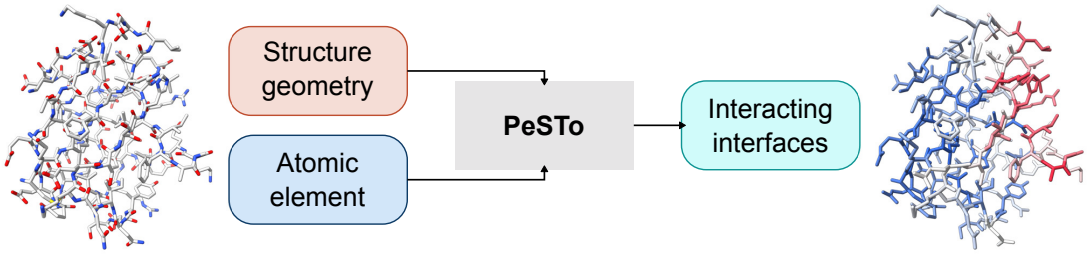

Figure 2.3: PeSTo. Protein binding interface prediction (on the right in red) from the atomic geometry and element using PeSTo.

Dataset

I curated the datasets for training, evaluating and testing the model. The dataset is composed of all the biological assemblies from the Protein Data Bank[29]. The subunits are clustered using a maximum of 30% sequence identity between clusters. The clusters of subunits are grouped into approximately 70% training set (376216 chains), 15% validation set (101700 chains), and 15% testing set (97424 chains). I selected the best hyperparameters by evaluating the model on the validation set. The testing set is composed of the clusters containing any of the 53 subunits from the MaSIF-site[14] benchmark dataset or 230 structures from the

Protein-Protein Docking Benchmark 5.0[64] (PPDB5) dataset. Additionally, I extracted a subset 417 structures common in the benchmark dataset of ScanNet[65] and the testing dataset of PeSTo. Unless specified, all the examples selected to assess the quality of the predictions from the model belong to the testing set.

Structure processing

The raw structures are not entirely clean and require some minimal processing. I define here the processing protocol applied on all structures. All models of the structures are loaded as a single structure. The chain name is tagged with the model identifier to distinguish subunits from different models. Moreover, the chain name of all non-polymer chemical molecules is tagged to have them in separate subunits. Duplicated subunits, molecules, and ions generated when concatenating multiple models are removed. The first alternate location of the atoms is kept. Water, heavy water, hydrogen, and deuterium atoms are removed from the structures.

Features and labels

The features and labels define the actual input and output of the model. Both have to be encoded in a way that is compatible with the deep learning model. I identified the 30 most common atomic elements on PDB. The element is used as the only feature as a one-hot encoding. The input vectorial features are initially set to zero. The distances matrices and normalized displacement vector matrices are used as geometrical features. Amino acids, nucleic acid, ions, ligands, and lipids are selected from a list of 20, 8, 16, 31, and 4 most common molecules, respectively. Non-native molecules used to help to solve the structure are ignored. An interface is defined as a residue-residue contact within 5 Å. All protein-protein interfaces as well as protein-nucleic acid, protein-ion, protein-ligand, and protein-lipids interfaces are identified. The details of the interface for each subunit are stored in the dataset as an interactions types matrix (79×79). This enables the selection of specific interfaces as labels at the start of the training session without having to rebuild the whole dataset. The interfaces targets can be selected from any combinations of subsets from the 79 molecules available.

Model architecture

I trained and evaluated many architectures and settled on the following one. The input features are embedded to an input state size of $S = 32$ with a 3 layers neural network with hidden layer size of 32. Each geometric transformer is composed of 5 neural networks of 3 layers to perform the multi-head self-attention ($S = 32$, $N_{key} = 3$, $N_{head} = 2$) as described in Algorithm 1. In the same fashion as applying convolution operations on an image, chaining geometric transformers can propagate information at a longer range than the local context of a single operation. The main architecture is based on a bottom-up approach, starting from a small

context of 8 nearest neighbors (≈ 3.4 Å radius) up to longer range interactions with 64 nearest neighbors (≈ 8.2 Å radius). The size of the context gradually increases allowing the model to progressively include more information while remaining cheaper in computations and memory for deep models. 4 sets of 8 geometric transformers with an increasing number of nearest neighbors for each set ($nn = 8, 16, 32, 64$) are applied in succession. For structures with a number of atoms smaller than the set number of nearest neighbors (nn), the additional non-existent interactions are sent to a sink node with a scalar and vector state set to zero.

The model predicts per residue if a residue is at the interface. Therefore, two additional modules are added to get the desired output. First, the geometric residue pooling module aggregates the encoding at the atomic-level of the structure to a residue-level description by using a local multi-head mask on the atoms forming each residues ($S = 32$, $N_{head} = 4$) as described in Algorithm 2. Lastly, a multi-layer perceptron with 3 layers of hidden size of $S = 32$ decoding the state of all residues and computing the prediction, returning a confidence score from 0 to 1.

Training

The final model is trained to predict protein interfaces with protein, nucleic acid, ligand, ion, or lipid. The best neural networks architecture was trained for 8 days on a single NVIDIA V100 (32 GB) GPU. Subunits with a maximum of 8192 atoms (~100 kDa) without hydrogens are used to limit the memory requirement during training. Subunits with less than 48 amino acids are ignored during training. I trained only on the first bioassembly provided by the PDB database. The effective generalized protein interfaces dataset after filtering is composed of 113805 subunits for training and 29786 subunits for testing.

Methods comparison

Our method was compared with ScanNet[65], MaSIF-site[14, 53], SPPIDER[50] and PSIVER[51]. ScanNet is the most recent geometry-based deep learning method for protein-protein interface prediction. MaSIF-site is the best available surface-based deep learning method for protein-protein interface prediction. SPPIDER is a long-standing and well-tested method used as a reference for protein-protein interface prediction. PSIVER only uses sequence information and is benchmarked to show the difference in performance between structure-based and sequence-based methods. The benchmarking of PeSTo was performed using structures taken from the testing dataset exclusively. I use 512 structures per interface type for the protein, ion and ligand interfaces predictions. The low amount of structures available limits the testing dataset to 391 and 161 structures for the nucleic acid and lipid interfaces prediction, respectively.

AlphaFold-multimer benchmark

I also compared our protein-protein interface predictions with AlphaFold-multimer. I identified 23 dimers (i.e., 46 interfaces) not present in the training set of PeSTo or of AlphaFold and with a maximum of 20% sequence identity with the AlphaFold-multimer training set (i.e., structures published up to April 30th 2018)[48]. I modeled the protein complexes using the implementation of AlphaFold-multimer by ColabFold with MMseqs2[66, 67] with the default parameters of 10 recycles and 5 predicted models. I extracted the protein-protein interfaces of the AlphaFold-multimer models (i.e., residue-residue contacts within 5 Å) and computed the average interfaces over the 5 predicted models. PeSTo was used to predict the protein-protein interfaces for the 46 subunits. Lastly, I computed the accuracy, precision, Matthews correlation coefficient (MCC), receiver operating characteristic (ROC) and precision-recall (PR) area under the curve (AUC) on the PeSTo predicted protein-protein interfaces and the average protein-protein interfaces of the AlphaFold-multimer predicted models.

Molecular dynamics simulations

I also predicted the protein-protein binding interface of protein in different conformations. 20 complexes from the PPDB5 dataset were selected based on the resolution of the structure and the difficulty to parametrize. For each, we performed a classical 1 µs-long MD simulation in the NpT ensemble (at 1 atm and 300 K, after NVT equilibration over 2 ns and with settings as in ref. [68]) of the subunits alone for the bound receptor (bR), unbound receptor (uR), bound ligand (bL), and unbound ligand (uL). All systems were set up using CHARMM36m[69] and its recommended TIP3P water model, and MD simulations were run with Gromacs 2020[70], 500 frames per simulation are used to evaluate PeSTo for a total of 400'000 frames, which are further clustered using the CLoNe algorithm[71] for the analysis of the unbound states.

Human interfaceome

In order to showcase the potential of PeSTo, we decided to predict the binding interfaces of all human proteins using structure predicted by AlphaFold. We call this database of interface the human interfaceome. To achieve this, I downloaded all the available 20'504 (at the time of writing) AlphaFold predicted structures version 2 for human sequences from the AlphaFold-European Bioinformatics Institute (AF-EBI) database[12, 72]. The same pipeline and data analysis can be applied to any organism. The most accurate AlphaFold structure models are selected with a minimum of 70% of the structure with a pLDDT > 70 and average PAE < 10 Å in the well-folded regions (pLDDT > 70). The analyzed dataset is composed of 7464 quality predicted structures from a total of 20,504.

PeSTo was applied to all models. For the analysis of interface residue composition and UniProt-annotated sequence regions, I considered only predicted interfaces with high con-

fidence (>0.8) at well-folded regions (pLDDT > 70). Interface residues are grouped into interfaces by connecting all residues at well-folded regions (pLDDT > 70) and at a predicted interface (>0.5) with α carbon within 10 Å. I selected only the predicted interfaces of quality with average predicted interface confidence above 0.8 for all analyzes. Two quality interfaces of different types are overlapping if they share at least 5 residues. Solvent-accessible surface area per atom of all models was computed using the Shrake and Rupley algorithm[73] implemented by MDTraj[74].

The UniProt-annotated features and GO terms for all corresponding 20'504 AlphaFold models were downloaded from UniProt website[30]. The features analyzed include a curated list of annotated features, the subcellular localization, the mutation sites, natural variants, and the GO biological process and molecular function. The pathogenicity of natural variants was extracted from the clinical significance of genetic variations available at dbSNP[75].

I downloaded all the 1102 predicted yeast protein complexes with AlphaFold and RoseTTAFold by Humphreys et al.[76], and extracted the interfaces from the predicted complexes with an interface defined as a residue-residue contact within 5 Å. PeSTo was applied to predict the protein-protein interfaces on the subunits of the predicted subunits of the complex alone.

2.4.2 Protein-carbohydrate binding interface prediction

After our initial work on predicting binding interfaces of proteins, Parth Bibekar and I aimed to further experiment with the capabilities of the PeSTo architecture by focusing on more specific interfaces. One area of interest was predicting the interaction between proteins and carbohydrates, we trained a new model called PeSTo-Carbo. The main challenge comes from the data availability. The structures of proteins interacting specifically with carbohydrates are limited in number, with around 6'000 available. This is significantly smaller compared to the approximately 100'000 structures we used for the protein binding interface prediction in general.

We also wanted to dive deeper by predicting binding interfaces between proteins and a particular class of molecules: cyclodextrins. The data for this is even more limited, with only about 150 structures. Due to this limitation, we experimented with transfer learning techniques. Our idea was to leverage the insights from our broader protein-carbohydrate interface predictions to enhance the performance of the model in the subset of this specific molecules. The results related to this section can be found in Section 3.2.5.

Dataset

PeSTo-Carbo was trained, validated and tested on protein-carbohydrate complexes collected from the Protein Data Bank (PDB)[29] and the subunits are clustered the subunits using a maximum of 30% sequence identity between clusters. The training, validation, and testing

datasets contain 5251, 408, and 343 subunits, respectively. All the subunits in the validation set have a resolution of less than 3 Å. Similarly, we collected biological assemblies containing protein-cyclodextrin complexes from the PDB. The training, validation, and testing datasets 138, 12, and 16 subunits for protein-cyclodextrin complexes, respectively. All performance scores and examples in this work are obtained from the test set.

Structure processing

Every model of the structure is loaded together as one entity. To distinguish them, non-polymer chemical molecules are given unique chain names for their separate subunits. Water molecules and hydrogen atoms are eliminated from the structures. In the dataset, cyclodextrin subunits are labeled distinctively from other glucopyranoses.

Features and labels

The input structures are described using the atomic elements, a matrix representing the pairwise distances between atoms, and the pairwise normalized relative displacement between atoms. I restrict the atomic elements to the 30 most common elements and represent them using one-hot encoding. The model works effectively without atom parameterization and can accommodate incomplete molecular structures.

The model aims to predict the residues that are in contact with carbohydrates. We defined an interacting interface between proteins and carbohydrates using a 4 Å cutoff distance: amino acids within 4 Å of a carbohydrate are considered in contact. We labelled protein interfaces with carbohydrates and cyclodextrin differently.

Model architecture

Similarly to the main PeSTo architecture, we first use a three-layer neural network to convert one hot encoding of the atom element into a scalar state size of 32. The initial vector state is initialized to a zero state. Then 24 geometric transformers are applied in series, each having two attention heads, a key size of 3 and a neighborhood of 8 to 64 nearest neighbors. Lastly, a four-headed self-attention within each residue aggregates the atomic-level encodings into a residue description. A three-layer neural network decodes the residue-level scalar and normed vector state to predict the binding interfaces using a sigmoid activation function.

Training

Two models with the same architecture were trained with different values of thresholds for contacts between proteins and carbohydrates. We employed binary cross entropy loss (BCE) as the objective function for the model. We used the Adaptive Moment Estimation (Adam)[20] with a learning rate of 1e-4. Furthermore, we assigned a weight of 0.9 to the posi-

tive label in the loss function to account for class imbalance in our dataset.

2.4.3 Sequence prediction from a backbone scaffold

Our exploration continued as I looked into the potential of the PeSTo architecture in the domain of amino acid sequence prediction from a backbone template. Specifically, the objective was to predict the likelihood of particular amino acids when provided with only the atom coordinates from the backbone of a protein, as illustrated in Figure 2.4. For this purposed, I trained a model for the Context-aware Amino acid Recovery from Backbone Atoms and heteroatoms (CARBonAra). Our primary objective was to test if sequences sampled from the prediction of our model could result in structures that can be recovered in-silico through methods like AlphaFold and AlphaFold-multimer. Subsequently, I looked into the capability of the model to improve sequence prediction when I introduce structural information of specific non-protein biomolecules. Lastly, I was interested to determine if the predicted likelihoods from our model were consistent with findings from deep sequencing analysis. The results related to this section can be found in Chapter 4.

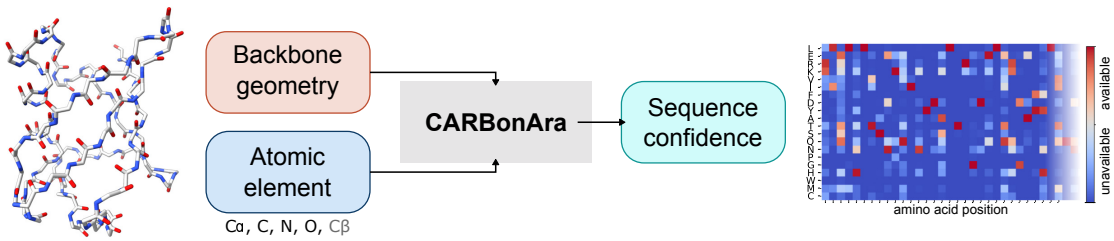

Figure 2.4: CARBonAra. Protein sequence prediction from a backbone scaffold.

Dataset

The training dataset is composed of ~370'000 subunits and the validation dataset contains ~100'000 downloaded from RCSB PDB biological assemblies. The test dataset is composed of ~70'000 subunits (single chain proteins) with no C.A.T.H similarity with the training set and less than 30% sequence identity with the test set. Within the test dataset, I extracted subunits without any C.A.T.H similarity and maximum 30% sequence identity with any training set of PeSTo (~370'000 subunits), ProteinMPNN (~540'000 subunits), or ESM-IF1 (~18'000 subunits). This comparison dataset is composed of 228 subunits: 76 monomers, 37 dimers, and other 22 multimers. Note that ProteinMPNN and ESM-IF1 both use C.A.T.H and 40% sequence identity clustering for training and testing.

Features and labels

During the processing phase, I kept only the backbone of proteins (C_α, C, N, O), disregarding the hydrogen atoms, while adding the virtual C_β using the ideal angle and bond length

in the same way as in ProteinMPNN[13]. The structures I used to train the model can contain any type of molecule including waters, ions, nucleic acids, and any other non-protein molecules. The input scalar state contains the one-hot encoded 30 most frequent atomic elements in the PDB database. The last one-hot channel represents any other or unknown element. The input vector state is initialized randomly drawn from an isotropic normal distribution. I incorporated the geometric features using the pair-wise distance matrices and normalized displacement vector tensor. The output of the model is a prediction confidence for each amino acid position among the 20 possible amino acid types. These types are represented as one-hot encoded labels. I optimized the model for multi-class classification of the 20 possible amino acids per position using a binary cross-entropy loss function.

Model architecture

I first embedded the input features into an input state size (S) of 32 using a three-layer neural network with a hidden layer size of 32. I then applied sequentially four sets of eight geometric transformers[77] ($S = 32$, $N_{key} = 3$, $N_{head} = 2$), see Algorithm 1. Each set of geometric transformers corresponds to an increased number of nearest neighbors ($nn = 8, 16, 32, 64$). In instances where the number of atoms is less than the set number of nearest neighbors (nn), I assigned any additional non-existent interactions to a sink node. I configured this sink node with a constant scalar and vector state of zero. Next, the geometric residue pooling module reduced the atomic-level encoding of the structure into a residue-level description. This aggregation used a local multi-head mask on the atoms that constitute each residue ($S = 64$, $N_{head} = 4$). Finally, I employed a multi-layer perceptron in the last module, which, using a three layers of hidden size ($S = 64$) decoded the state of all residues and computed the prediction, consequently generating a confidence score of the 20 possible amino acids through a sigmoid function ranging from 0 to 1.

Training

I trained our neural network architecture for 16 days on a single NVIDIA V100 (32 GB) GPU. To manage memory usage during training, I limited the subunits to a maximum of 8192 atoms (approximately 100 kDa), excluding hydrogen atoms. Furthermore, subunits containing fewer than 48 amino acids were not considered in the training process. The post-processing effective dataset contains 86610 structures in the training dataset and 24601 structures in the validation dataset.

Sequences sampling

I sampled the optimal sequence by taking the highest confidence amino acid per position from the prediction. To generate sequences with minimum sequence identity to the scaffold, I selected the highest confidence predicted possible amino acid above the positive prediction

threshold of 0.5, which is not the original amino acid from the scaffold. The original amino acid is only used in the sequence generated if it is the only possible option within the positive predictions. Our criterion for defining similarity between two amino acids relies on their BLOSUM[63] 62 score. I considered them as similar if this score is above zero. I sampled low similarity sequence to the original scaffold by restricting the positive predicted amino acid. When the options are available, I selected the amino acid with the highest BLOSUM 62 score below or equal to zero compared to the reference scaffold. If there are no options with a BLOSUM 62 score below or equal to zero, I sampled the positive predicted amino acid with the lowest BLOSUM 62 score. I noted that taking only the minimum BLOSUM 62 similarity score generates sequences with a bias towards special amino acids (i.e., cysteine, proline, glycine). I performed a BLAST[78] search to measure the novelty of the generated sequences with minimum identity and low similarity using the non-redundant protein sequences database (nr) with a expect value (e-value) cut-off at 100.

AlphaFold and AlphaFold-multimer validation

In the case of monomers, I sampled the highest confidence sequence from the predictions of CARBonAra for 142 subunits of the testing dataset. I also generated sequences using ProteinMPNN and ESM-IF1 both with a sampling temperature of 1e-6. I modelled the structures from the generated sequences with ColabFold20 (version 1.5.2) using the `alphafold2_ptm` model, in single sequence mode and with 3 recycles[12]. In the case of dimers, I generated sequences for one subunit given the sequence of the other subunit. I sampled the sequence with the highest confidence from CARBonAra for the 31 dimers in the testing dataset for a total of 62 complexes with conditioning. I predicted the structures from the generated sequences with ColabFold (version 1.5.2) using the `alphafold2_multimer_v2` model, in single sequence mode and with 5 recycles[48]. To evaluate the sampling flexibility of CARBonAra, I sampled sequences with maximum identity, minimum identity and low similarity using CARBonAra's multi-class predictions. In this case, I used AlphaFold using multiple sequence alignment since a low sequence identity or similarity negates the sequences matching the reference scaffold in the multiple sequence alignment information. I assessed the predicted structures from the generated sequence with the original scaffold using the TM-score[79] and local Distance Difference Test[80] (lDDT) on the C_α coordinates.

Molecular dynamics simulations

Luciano Abriata selected 20 complexes from the Protein-Protein Docking Benchmark 5.0 dataset[64] based on structure resolution and parameterization difficulty. For each complex, we conducted a standard 1 µs-long molecular dynamics (MD) simulation in the NPT ensemble (at 1 atm and 300 K, following a 2 ns NVT equilibration and using settings as per ref. [68]) for the bound receptor, unbound receptor, bound ligand and unbound ligand. We set up all systems using CHARMM36m[69], running MD simulations with GROMACS 2020[70] (single chain structure) MD, we sampled 500 frames for each simulation and computed the average

prediction confidence.

Comparison with deep sequencing

As a case study, Fernando Meireles and I showed that residue-wise estimated probabilities of CARBonARa can be reliably correlated with experimentally determined mutations for the class A β-lactamase TEM-1. This widely studied enzyme has been subjected to deep mutagenesis[81], where the effect of consecutive triple point mutations along the whole extension of the protein was analyzed, covering all 20 naturally occurring amino acids per position. The generated libraries were introduced in E. coli and selected based on ampicillin resistance. These data were used to compute a statistical change in free energy of binding ($\Delta\Delta G_{stat}$) of mutation of all wild-type residues in the protein. This value was calculated as $\Delta\Delta G_{stat} = RTln(\frac{p_{wt}}{p_{mut}})$, where p_{wt} and p_{mut} are the probabilities of finding the wild-type and mutant amino acids, respectively, at the analyzed sequence position. The same calculation was performed on an MSA of 156 sequences of class A β-lactamases, to compare the conservation profile of this family with the requirements imposed by the mutagenesis assays. Aiming at assessing the ability of CARBonARa to recover evolutionary-related residue profiles, we used its residue-wise estimated probabilities to compute the $\Delta\Delta G_{stat}$ per position of TEM-1. We used two structures of TEM-1 as input for the model: TEM-1 in the apo state (PDB ID: 1JTG, removing all non-protein atoms) and TEM-1 retaining its catalytic water and β-lactam nitrocefin at the catalytic pocket. Docking of this ligand to TEM-1 was carried out with AutoDock Vina[82] and the analyzed pose was selected based on the proximity of the carbonyl group of the β-lactam ring to the catalytic residue S70. We then calculated Pearson's correlation coefficient (ρ) of the $\Delta\Delta G_{stat}$ per sequence position between the mentioned approaches, assessing if the overall ranking of the $\Delta\Delta G_{stat}$ obtained could be similarly explained by the deep sequencing, the conservation profile, and the estimated probabilities of CARBonAra.

3 PeSTo: parameter-free geometric deep learning for accurate prediction of protein binding interfaces

Krapp L.F., Abriata L.A., Cortés Rodriguez F., Dal Peraro M.
PeSTo: parameter-free geometric deep learning for accurate prediction of protein binding interfaces
Nat Commun 14, 2175 (2023).
https://doi.org/10.1038/s41467-023-37701-8

Disclaimer

Contributions

L.F.K. and M.D.P. conceived and designed the research project. L.F.K. designed and implemented PeSTo. L.F.K., L.A.A., and M.D.P. analyzed the data. L.F.K. and F.C.R. developed the PeSTo webserver. L.F.K., L.A.A., and M.D.P. wrote the paper.

3.1 Introduction

Molecular interfaces are ubiquitous in biology and of utmost relevance beyond their central role in establishing cell boundaries and intracellular organization[83, 84, 85]. Especially so around proteins, which perform their functions by interacting with other proteins as well as with nucleic acids, membranes, and molecules and ions of various kinds.

Predicting the interactions that a given protein can establish with other molecules remains a major challenge in biology, still open despite numerous developments along various fronts[86, 87, 88, 14]. The most modern methods for predicting protein interactions currently target the prediction of either specific pairs of interacting residues/atoms, relying intensively on the analysis of residue-residue coevolution patterns and thus limited to protein-protein interactions, or predicting only which regions of a protein are prone to interaction[14, 89, 90, 91, 92, 93, 94, 95]. Even the latter, presumably a simpler problem, is yet far from solved, and most methods aim mainly at discovering protein interfaces tailored to interact with other proteins, with a strong focus on features of the protein surface and in some cases also exploiting their sequence conservation. These methods are thus limited, because the calculation of protein surfaces and mapping of their properties are time-consuming, complicating their high-throughput application at the proteome scale; besides, they require parametrizations and are very sensitive to details and errors of the 3D structure or model[14, 91, 93, 94, 95, 65]. Meanwhile, methods that rely on sequence conservation or residue coevolution often perform poorly for shallow sequence alignments. Approaches based on folding protein complexes de novo simultaneously discovering the interaction interfaces and subunit conformations, such as AlphaFold-multimer[48], are limited to protein-protein interactions, are far slower than predicting the interaction interface from structures and will fail if the folding protocol itself fails.

Here, building on the recent successful application of transformers[26, 44, 12, 46] to various problems in natural language processing and protein structure prediction, we developed a rotation-equivariant transformer-based neural network that acts directly on protein atoms predicting interaction interfaces with high confidence, without the need for parameterization of the system's physics, running fast enough to process large structural datasets such as ensembles from molecular dynamics simulations or entire foldomes. We build on this transformer to develop PeSTo—the Protein Structure Transformer—a generalized predictor of protein binding interfaces.

Trained to predict protein-protein interaction interfaces, PeSTo outperforms the state-of-the-art. Training to predict other kinds of binding interfaces was straightforward as the method does not depend on any explicit parametrization of physicochemical features. Therefore, confident predictions of protein interactions with nucleic acids, lipids, ligands and ions are also easily produced. Given the computational performance of the method, we could provide it not only as standalone code but also implemented in a user-friendly webserver (pesto.epfl.ch). PeSTo runs fast enough to allow processing of large volumes of structural

data, such as molecular dynamics trajectories, enabling the discovery of cryptic interacting interfaces[96, 97], and the continuously growing foldome provided by AlphaFold predictions, which allows us to perform a detailed analysis of the human interfaceome.

3.2 Results

3.2.1 The Protein Structure Transformer (PeSTo)

Many successful methods combine transformers[26, 44] and geometric deep learning[14] representing structures as graphs or point clouds and integrate the requirement of the invariance or equivariance of the neural network[37, 43, 38, 39, 40, 98, 99]. The major breakthroughs come from the field of protein folding[31], where AlphaFold[12] integrates attention in the Evoformer blocks and the structure module and the third track of the RoseTTAFold[46] model uses a SE(3)-Transformer[45] to refine the atom coordinates during folding. Moreover, the recurrent geometric network[47] (RGN2) leverages the Frenet-Serret formulas to represent the backbone of proteins, and the geometric vector perceptron[41] (GVP) uses linear operations to compose vector features with gating[100]. Multiple other machine learning-based protein-protein interaction site prediction methods have been developed[14, 50, 53, 51].

We introduce here PeSTo, a parameter-free geometric transformer that acts directly on the atoms of a protein structure. As shown in Figure 3.1 and detailed in Methods 2.4.1, the structure is represented as a cloud of points centered at the atomic positions, and its geometry is described through pairwise distances and relative displacement vectors which guarantee translation invariance. The atoms are described using only their elemental names and coordinates without any explicit numerical parametrization such as mass, radius, charge or hydrophobicity. Each atom is associated with a scalar state (q) and a vector state (p) encoding the properties of the structure. We define a geometric transformer operation acting on this cloud of points to update these states using the states and geometry in their local neighborhood as shown in Figure 3.1a. The interactions between atoms for all nearest neighbors (nn) is encoded using the geometry (i.e., distance and displacement vector) and the state of the pair of atoms involved. A multi-head attention layer eventually decodes and regulates the propagation of the information (Algorithm 1).

The geometric transformer operation is translation-invariant, rotation-equivariant and independent of the order of the atoms and order of the interactions. In order to retain the rotation equivariance of the vector states (see Methods 2.4.1), the transformer attention linearly combines the scaled vectors from the local geometry and local state vectors to dynamically propagate vector state information based on the local context. The attention operation allows for a dynamic number of nearest neighbors (nn). However, in practice, the operation is much more computationally efficient with a fixed number of nearest neighbors. In the same fashion as applying convolution operations on an image, chaining geometric transformers can propagate information at a longer range than the local context of a single operation. There-

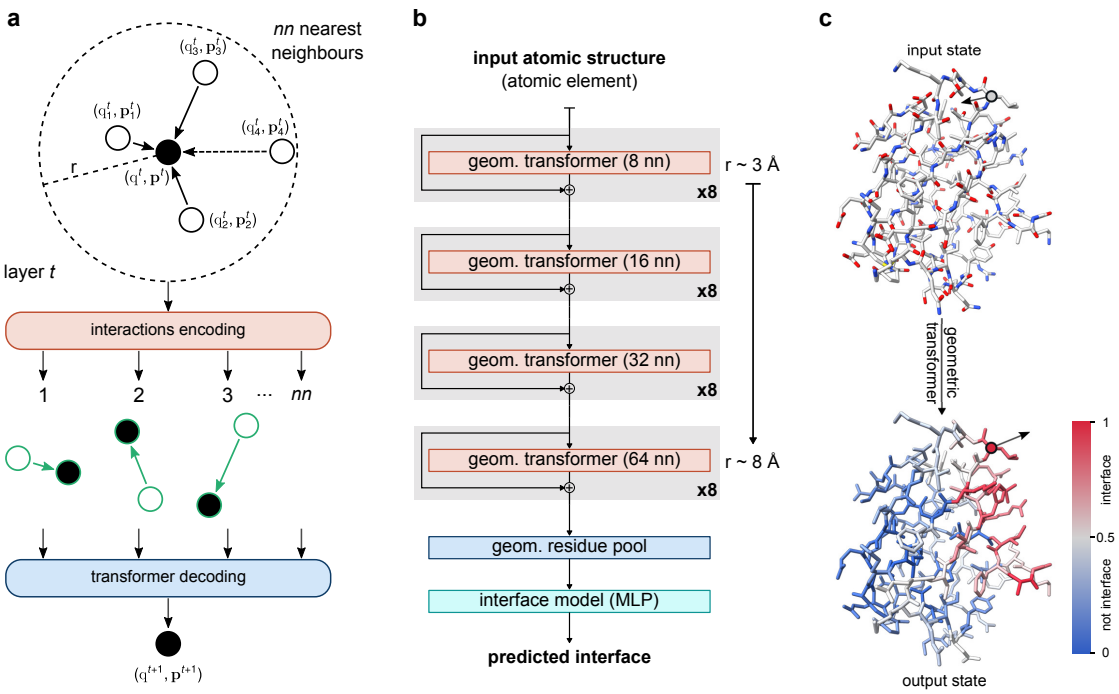

Figure 3.1: Overview of the PeSTo method. (**a**) Primary geometric transformer acting on the scalar and vectorial state of an atom at layer *t*. The interactions between the central atom and the nearest neighbors are encoded. A transformer is used to decode and filter the interactions information and to compute the new state of the central atom (Algorithm 1). (**b**) The architecture of PeSTo for the prediction of interaction interfaces. The model is composed of multiple layers of geometric transformers with a set number of nearest neighbors (*nn*) and residual connections. The structure is reduced to a residue representation through an attention-based geometric pooling (Algorithm 2). The residue states are collapsed, and the final prediction is computed from a multi-layer perceptron (MLP). (**c**) Example of application of the primary geometric transformer to all atoms in a structure.

fore, the main architecture is based on a bottom-up approach, starting from a small context of 8 nearest neighbors (~3.4 Å radius) up to long-range interactions with 64 nearest neighbors (~8.2 Å radius, Figure 3.1b). The size of the context gradually increases allowing the model to progressively include more information while remaining cheaper in computation requirements and memory for deep models. The residual connection between geometric transformers enables to train deeper neural network architectures. Two additional modules aggregate the atom-based geometric description at the residue level independently of the number of atoms within a residue (i.e., geometric residue pooling, Algorithm 2) and predict whether each amino acid is at an interacting interface or not (Figure 3.1c).

In comparison with previous approaches like the SE(3)-transformer[45] that uses spherical harmonics to encode geometrical context, our method simply uses vectors, modulating their information through the transformer attention. Compared to equivariant convolution, our method is based on graphs with geometry and performs message-passing using transformers.

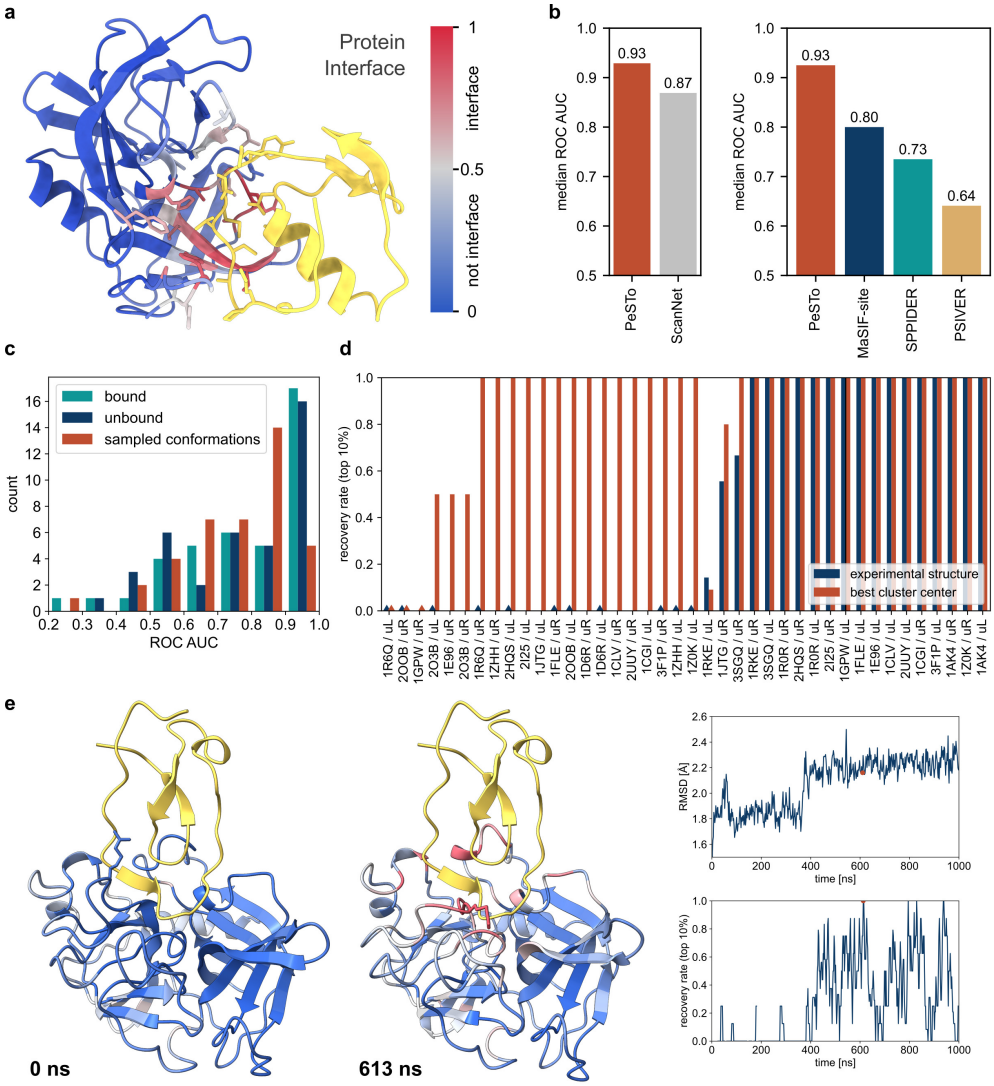

Figure 3.2: Assessment of protein–protein interface predictions with PeSTo. (**a**) Example of protein-protein interface prediction for the unbound conformation of Streptogrisin B (PDB: 2QA9) as can be retrieved at pesto.epfl.ch. The confidence of the predictions is represented with a gradient of color from blue for non-interfaces to red for interfaces. The ligand in yellow was subsequently added based on the structure of the complex (PDB: 3SGQ) to show the quality of the prediction. (**b**) Comparison against other methods for protein–protein interface prediction. The methods are evaluated on PeSTo groundtruth on two different testing datasets for ScanNet and MaSIF-site. (**c**) Benchmark of our method on bound and unbound experimental structures, as well as their conformations sampled by 1 µs-long MD simulation for 20 complexes taken from the PPDB5. (**d**) Recovery rate (considering top 10% predicted residues) for the clustering of predicted interfaces on 1 µs-long MD simulations of the unbound state only, compared to the predicted interface of the experimental structure for the unbound receptor (uR) and ligand (uL) for 20 complexes taken from the PPDB5. (**e**) Protein–protein interface prediction on the experimentally resolved structure of unbound porcine pancreatic elastase (left, PDB: 9EST) and an open conformation sampled using MD (center) and selected using clustering on the conformations. The ligand in yellow was subsequently added based on the structure of the complex (PDB: 1FLE) to show the quality of the prediction. R217 is shown in licorice to illustrate the rearrangement of the loop region. (Right) The root mean square deviation (RMSD) from the experimental unbound conformation and recovery rate average over 4 frames are shown as a function of the simulation time (the red dots indicate the selected snapshot shown in (**d**)).

3.2.2 Protein-protein interface prediction

We trained a PeSTo model using over 300'000 protein chains from the PDB (see Methods 2.4.1) to predict which residues are involved in a protein-protein interface as flagged by an output value ranging from 0 to 1 (Figure 3.2a). Zero means that the residue is predicted to not be engaged in interactions, while values of 1 predict the residue to be at an interface. In practice, the actual value of the prediction reflects the confidence of the prediction at the residue level, such that values farther away from 0.5 imply higher confidence, see Figure 3.3.

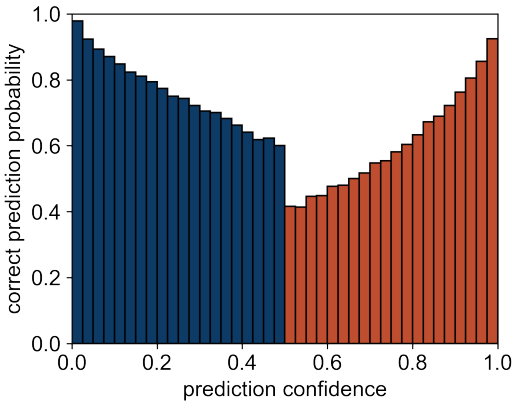

Figure 3.3: Prediction quality estimation. Estimated correlation between protein-protein interface prediction confidence and prediction quality. Evaluated on 8192 structures randomly sampled from the testing dataset.

We first evaluated the performance of PeSTo against the most recent method develop to address a similar task, namely ScanNet[65]. We used a benchmark dataset of 417 structures commonly shared by the two methods (see Methods 2.4.1). On this benchmark PeSTo outperforms ScanNet without multiple sequence alignment (MSA) with a median receiving operating characteristic (ROC) area under the curve (AUC) of 0.93 against 0.87 (Figure 3.2b). Moreover, we compared the speed of the two methods quantitatively (Figure 3.4), finding that the average runtime for PeSTo ($5.3 \pm 2.8s$) and ScanNet without MSA ($9.1 \pm 1.8s$) on CPU are comparable. However, the runtime of ScanNet with MSA ($160 \pm 83s$) is two orders of magnitude slower than PeSTo, providing no substantial improvement against PeSTo.

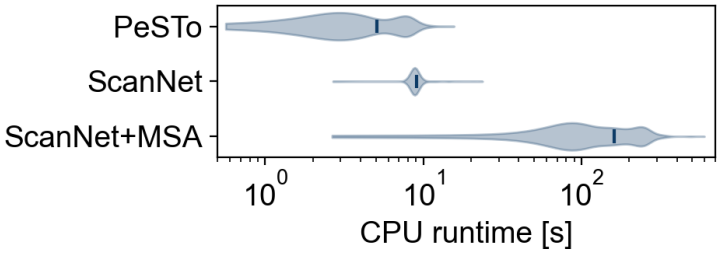

Figure 3.4: Runtime comparison of PeSTo with ScanNet. We compare PeSTo to ScanNet with and without multiple sequence alignment (MSA) on CPU (Intel i9-9900K) using 417 structures from the ScanNet benchmark dataset. We show the CPU runtime the distribution (shaded) and mean of the distribution (line) of each method

We further compared PeSTo on the same dataset used to benchmark MaSIF-site[13, 53]6 (one of the best algorithms currently available), which we excluded from our training set at 30% sequence identity. PeSTo reaches a median receiving operating characteristic (ROC) area under the curve (AUC) of 0.92 against 0.8 for MaSIF-site followed by SPPIDER[50] and PSIVER[51] (Figure 3.2b). The interfaces predicted by PeSTo have a higher ROC AUC than all other methods benchmarked here for 38 out of 53 structures.

Finally, we compared the protein-protein interfaces as predicted by PeSTo against those predicted by AlphaFold-multimer. We selected 23 dimers (i.e., 46 interfaces) from the structures within the validation set of PeSTo and AlphaFold (see Methods 2.4.1). We observed that PeSTo (0.94 ROC AUC and 0.84 PR AUC) performs almost as well as AlphaFold-multimer (0.94 ROC AUC and 0.88 PR AUC) without the additional cost of computing any multiple sequence alignment. These results show therefore how our method can be used to quickly screen for potential interfaces with an accuracy comparable to AlphaFold-multimer.

To further showcase the quality of the predictions in real-world applications, we tested proteins from the Protein-Protein Docking Benchmark 5.0[64] (PPDB5) dataset in their unbound conformations. The example in Figure 3.2a shows PeSTo recovering the interaction interface of Streptogrisin B with ovomucoid from its unbound conformation (0.93 Å RMSD from the bound state) with a ROC AUC of 0.96. Overall, on the whole PPDB5 dataset composed by a variety of targets of variable difficulty for the general task of protein-protein docking, PeSTo reaches a median ROC AUC of 0.78 for predictions on the unbound structures and 0.85 for the respective bound states.

Importantly, the short time needed to run the model (i.e., 300ms for a 100kDa protein from PDB load to prediction on a single NVIDIA V100 GPU, Figure 3.5) allows us to evaluate snapshots from large structural ensembles efficiently, extracted from molecular dynamics (MD) simulations. We applied PeSTo for protein-protein interface prediction on conformations sampled by 1 µs-long atomistic MD simulations of the experimentally derived unbound and bound subunits of 20 selected binary complexes taken from the PPDB5 (Figure 3.2c). The

bound and unbound structures along with the MD-sampled conformations reach a median ROC AUC of 0.85, 0.82 and 0.79, respectively. We observe that the model performs almost as well on experimentally solved bound and unbound conformations. Although overall the ROC AUC decreases with a higher RMSD from the bound structure (Figure 3.6), our method is still able to recover the interface with a ROC AUC higher than 80% for most structures and MD-sampled conformations.

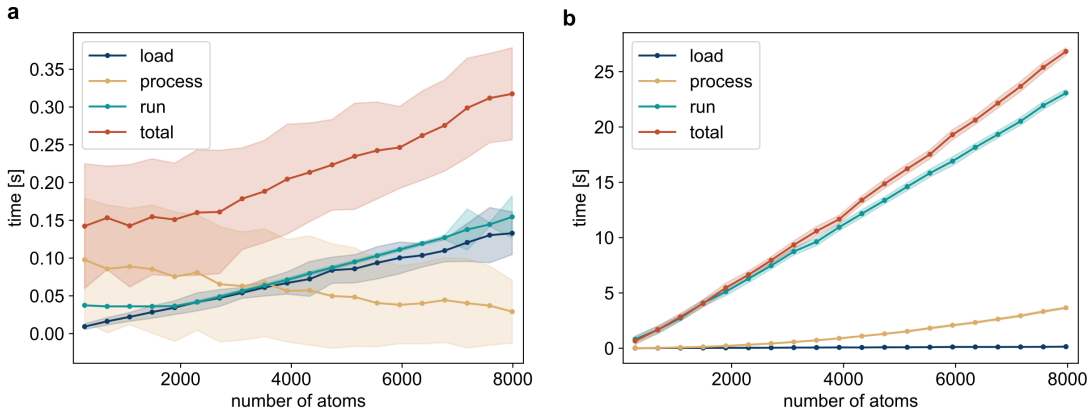

Figure 3.5: Profiling of the run time of PeSTo as a function of the size of the structure. Run time evaluated (a) on GPU (NVIDIA RTX 2080 Ti) and (b) on CPU only (Intel i9-9900K). For structures of around 100 kDa (8000 atoms), the average total runtime is 300 ms with 130 ms to parse the file, 30ms to process the structure and 140 ms to run the inference on a high-end GPU. Data are presented as the mean ± standard deviation using (a) n=194 and (b) n=19 randomly sampled structures from our test set per range of number of atoms.

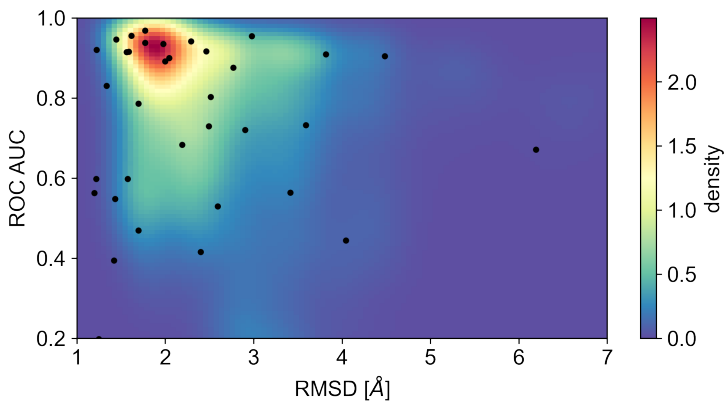

Figure 3.6: Effect of conformation on prediction quality. ROC AUC as a function of RMSD for different conformations for the 80 simulated subunits from the PPDB5 dataset. The RMSD is computed from the bound conformation of the subunits in the reference complex. Starting conformations are indicated with a black dot.

In some cases, processing MD trajectories of unbound proteins with PeSTo identifies certain

interfaces better than when PeSTo is run on the starting static structure, which suggests an impactful practical application of our method to real-life situations (Figure 3.2d). Striving to provide a protocol for every-day applications of PeSTo, we consider that a user might look for a handful of high-ranked residue predictions to characterize the binding interface. We define therefore the "recovery rate" as the ability to predict the 10% high-ranked residues, which in the case of our MD dataset correspond to 3 ± 2 residues. If all these residues are predicted correctly, we consider that the interface is fully recovered. Out of 20 complexes composed by 40 constituent subunits and relative interfaces, the model has a perfect recovery rate for 16 interfaces when applied straightaway on the experimental structures of the unbound subunits. Out of the remaining 24 cases, we show that it is possible to fully recover the binding interface for additional 16 subunits (80%) using MD to more extensively sample the protein conformation landscape and clustering to further group predicted interfaces.

For instance, PeSTo predicts no interface for the experimentally solved structure of the unbound porcine pancreatic elastase (PDB ID 9EST) (Figure 3.2e). The unbound experimental conformation has a backbone RMSD of 1.2 Å from the bound complex with elafin (PDB ID 1FLE). However, MD simulation starting from the unbound porcine pancreatic elastase alone shows a conformational switch leading to the recovery of the interaction interface with elafin with a cluster center ROC AUC of 0.92 and perfect recovery rate of predicted binding interface (i.e., 3 residues in this case). Inspecting the MD simulation unveils that the motion of a loop in elastase is required to allow elafin to enter the pocket and accommodate an inter-molecular β-sheet that stabilizes the complex as solved experimentally.

3.2.3 General protein binding interface prediction

In light of the results for protein-protein interface predictions, we extended the model to find and identify more types of interfaces, resulting in a generalized PeSTo model that predicts protein interaction interfaces with other proteins as well as with nucleic acids, ions, ligands, and lipids. We trained a generalized PeSTo model with PDB structures featuring all the kinds of expected interactions, as described in Methods 2.4.1. The interface predictions for protein-nucleic acid interfaces are almost as good as for protein-protein interfaces, reaching ROC AUC of 0.89 for the testing set (Figure 3.7a). The generalized model can also detect ion, ligand, and lipid interfaces with ROC AUCs of 0.87, 0.86, and 0.77, respectively on each testing set. The model does experience some confusion between ions and ligands as revealed by the confusion matrix (Figure 3.8). Poorer performance on protein-lipid prediction depends on the quite limited number of protein-lipid complexes available so far in the PDB (only 0.7% of the utilizable data we compiled). We note that retraining the model on the same dataset but with a maximum of 5% sequence identity instead of 30% between training, validation and test sets results in equivalent performances within $\pm 1\%$ ROC AUC in average over all interfaces prediction type, confirming PeSTo stability over homology reduction.

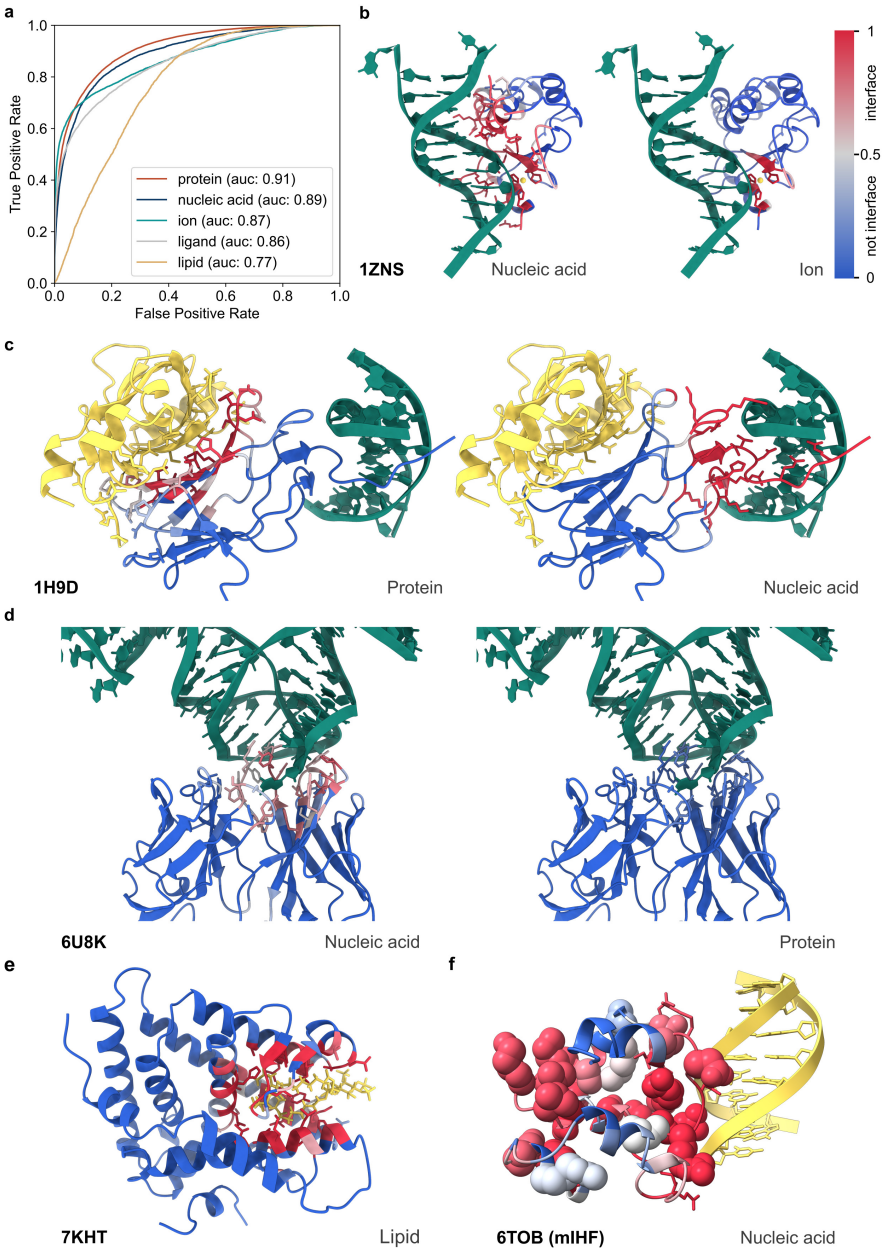

Figure 3.7: General protein binding interface prediction with PeSTo. (**a**) ROC curve for the predictions of different types of interfaces with PeSTo. (**b**–**f**) Examples of predicted binding interfaces. The confidence of the predictions is represented with a gradient of color from blue for non-interfaces to red for interfaces. The structures in yellow and green were added subsequently from the reference complexes. (**b**) Colicin E7 endonuclease domain in complex with DNA and a zinc ion (PDB: 1ZNS). (**c**) core-biding factor subunit alpha-2 in complex with core-binding factor subunit beta and DNA (PDB: 1H9D). (**d**) Antigen-binding fragment in complex with RNA (PDB: 6U8K). (**e**) Steroidogenic factor 1 bound to a phosphoinositide (PDB ID: 7KHT). (**f**) Predicted interface with nucleic acid for the mycobacterial integration host factor (PDB ID: 6TOB). Residues observed to bind DNA through solution-state NMR are represented with spheres. The DNA molecule is modeled from an X-ray structure of the protein homolog from S. coelicolor, crystallized with DNA (PDB ID: 4ITQ).

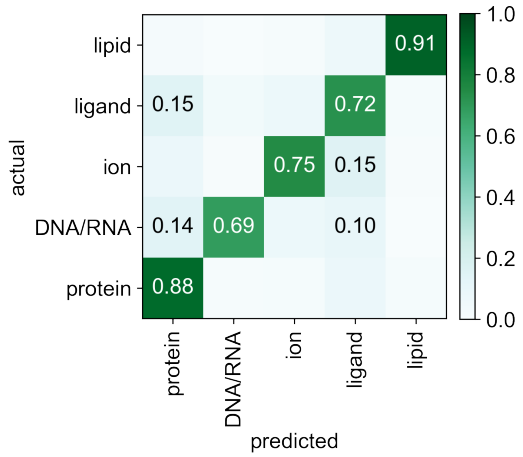

Figure 3.8: Confusion matrix between actual and predicted interfaces at the residue level. Each interface type is randomly sampled equally at ~3600 residues per class. The confusion matrix is normalized per actual interface. We observed nucleoside triphosphates (ATP, GTP) and diphosphates (ADP, GDP) pockets misclassified as nucleic acid binding regions, a reasonable confusion given the chemical similarity of all these molecules.

We next illustrate the generalized PeSTo model showcasing five examples from the testing set that attest to its capacity to discern among various interfaces, even when they are overlapping or under-represented in the PDB. The first example (Figure 3.7b) corresponds to the colicin E7 endonuclease domain, which binds DNA through an interface that includes a zinc ion (PDB ID 1ZNS). Running the apo-protein through the generalized PeSTo returns correct predictions for both interfaces, even in the overlapping part. The second case (Fig. 3.7c) corresponds to the complex formed by RUNX1 with a dsDNA bound to one end and the protein CBFβ bound to the other (PDB ID 1H9D). Running the isolated RUNX1 through the generalized model returns clear, accurate interfaces through the DNA and protein channels. In the third example (Figure 3.7d) we challenge the generalized model with the structure of an antibody that binds RNA (PDB ID 6U8K) as opposed to most available antibodies which are bound to other protein targets. The generalized model correctly predicts no interface for proteins and the correct interface for RNA.

Although on interfaces with lipids the generalized PeSTo performs less well, in practice we observe that the model is able to accurately detect lipid-binding pockets for soluble proteins (exemplified by the steroidogenic factor in Figure 3.7e) and even the membrane-spanning regions of transmembrane proteins (Figure 3.9). Despite not specifically trained for any of these, in both cases PeSTo is able to detect specific pockets for lipids with stronger scores. We note that many protein interfaces with lipids are only partially evident in PDB structures (for example a single lipid bound to a membrane-spanning region), resulting in low training data quality thus leading to an artificial drop of the ROC AUC.

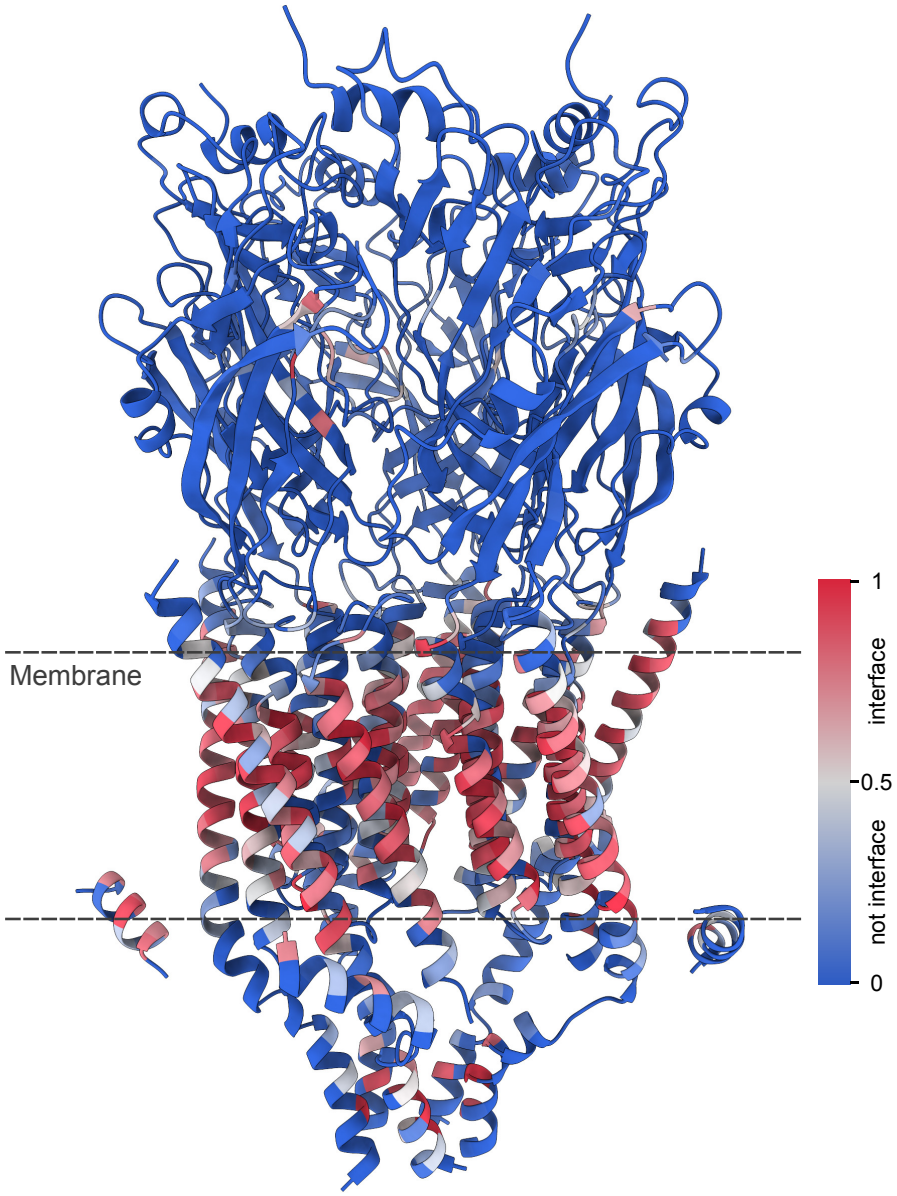

Figure 3.9: Example of lipid interface prediction for transmembrane protein. Homopentameric 5-HT 3A serotonin receptor (PDB ID: 6Y5B).

Interestingly, we also find that PeSTo extends its prediction power over its own training, as exemplified for the case of a DNA-binding bacterial integration host factor (mIHF) for which an X-ray structure of the DNA-bound form was available (Figure 3.7f). This structure presents in the biological assembly one DNA-binding interface[101] that was included in the training set, but solution-state NMR titrations have shown a far more extensive interaction surface, mainly spread over two surface patches as required to bend DNA as demonstrated by AFM[102]. PeSTo's predictions for this protein go beyond its training, pointing at two surface patches that match very well with the NMR data in solution.

3.2.4 High-throughput prediction of binding interfaces for the human proteome

We sought to explore the whole human proteome and analyze what we call hereafter the interfaceome, namely all the potential protein interfaces able to bind other proteins, nucleic acids, lipids, ligands and ions. For this task, we obtained all the structures and models for human proteins in the AlphaFold-European Bioinformatics Institute (AF-EBI) database[12, 72]. The database currently includes highly accurate structures, many actually containing domains with experimentally solved structures, models with no structures in the PDB or with little homology to PDB structures yet highly accurate as judged by AlphaFold predicted local distance difference test (pLDDT) and predicted alignment error (PAE), and also several models of very low pLDDT and PAE scores. We selected 7464 high-quality models for further analysis from the total of 20504 entries based on their pLDDT and PAE scores, as described in Methods 2.4.1.

We could immediately notice that our model produces robust results that further validate the quality of interface predictions. In particular, the amino acid distributions for specific molecular interfaces recapitulates known biochemistry (e.g., Arg and Lys residues are mostly engaged in nucleic acid interactions, hydrophobic amino acids in lipid-binding sites, etc., see Figs. 3.11, 3.10). Furthermore, mapping the predicted interfaces to UniProt-annotated features showed strong agreement with the expected functional roles of the binding interfaces (Fig. 3.12a). Additional support for the quality of the predictions came from the mapping of the predicted interfaces and their subcellular localizations, GO functions and processes (Figs. 3.13, 3.14, 3.15).

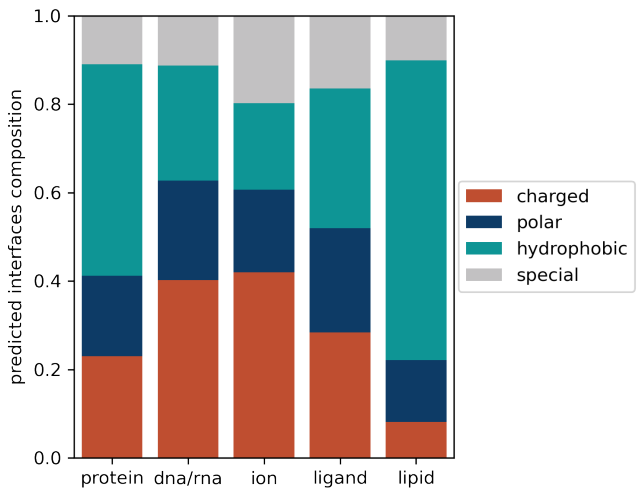

Figure 3.10: Predicted interface composition. Charged (ARG, HIS, LYS, ASP, GLU), polar (SER, THR, ASN, GLN), hydrophobic (ALA, VAL, ILE, LEU, MET, PHE, TYR), and special (CYS, GLY, PRO) residue composition for the different predicted interface types.

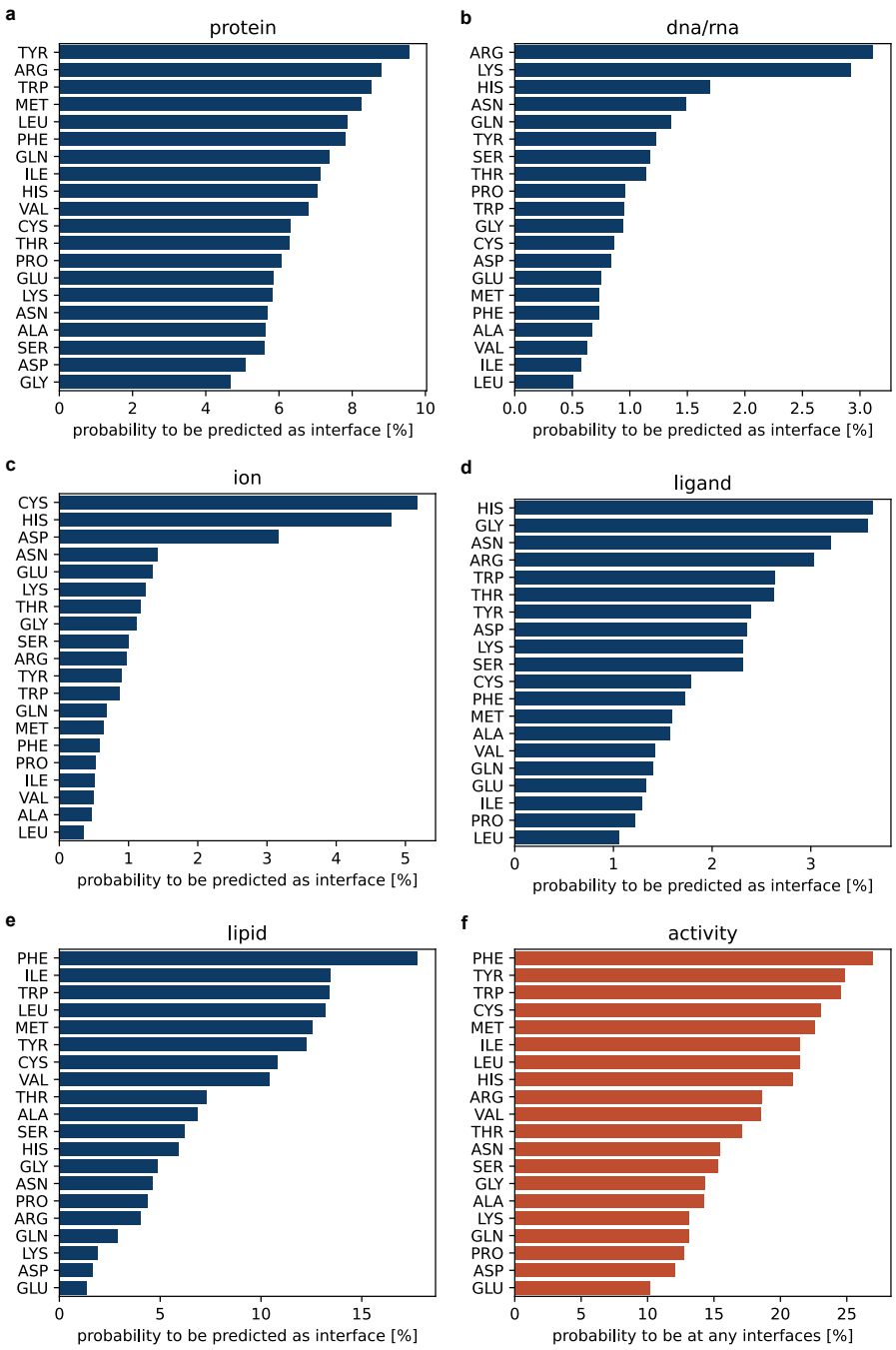

Figure 3.11: Probability of residues to be at a predicted interface. (a-e) Probability of different amino acids to be at a protein-protein, -nucleic acid, -ion, -ligand or -lipid predicted interface. (f) Probability of different amino acids to be at any interacting interfaces.

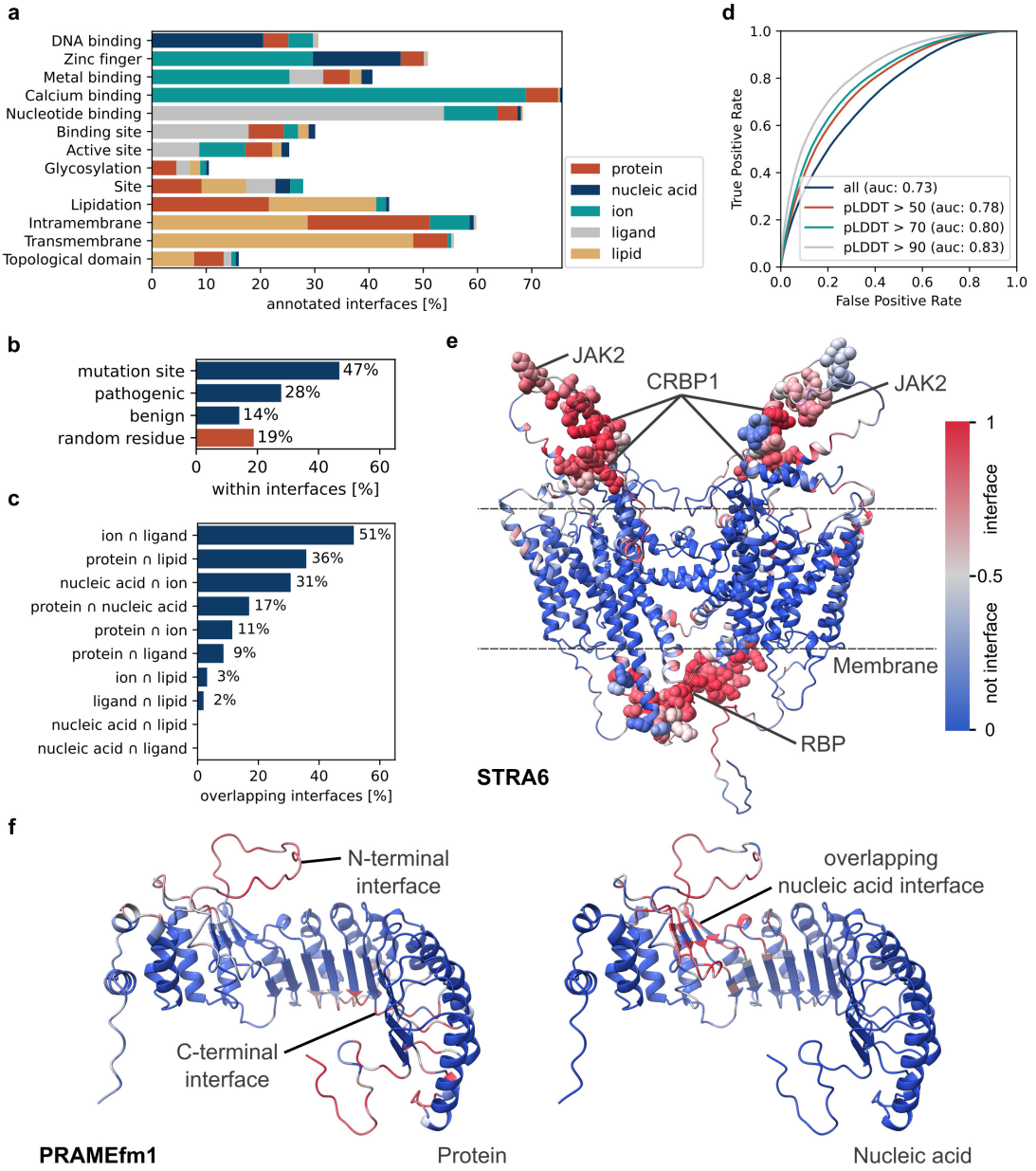

Figure 3.12: PeSTo-based analysis of the human proteome. (**a**) Percentage of entries with specific UniProt features for which PeSTo predicts an interaction interface at the annotated sequence region. (**b**) Percentage of sites with mutations, pathogenic or benign natural variants within a predicted interface. The baseline is the probability of a random residue being within an interface. (**c**) Percentage of overlapping interfaces for all 10 pairs of five interface types. (**d**) Comparison of predicted protein-binding interfaces from PeSTo using the models of yeast protein complexes predicted by Humphreys et al.[76]. Regions of the predicted structures are filtered out at different pLDDT thresholds. (**e**) Human receptor for retinol uptake (STRA6, UniProt Q9BX79). Protein interfaces predicted with PeSTo. Sites of interest as described by Berry et al.[103] are highlighted with spheres and are consistently found by PeSTo predictions.

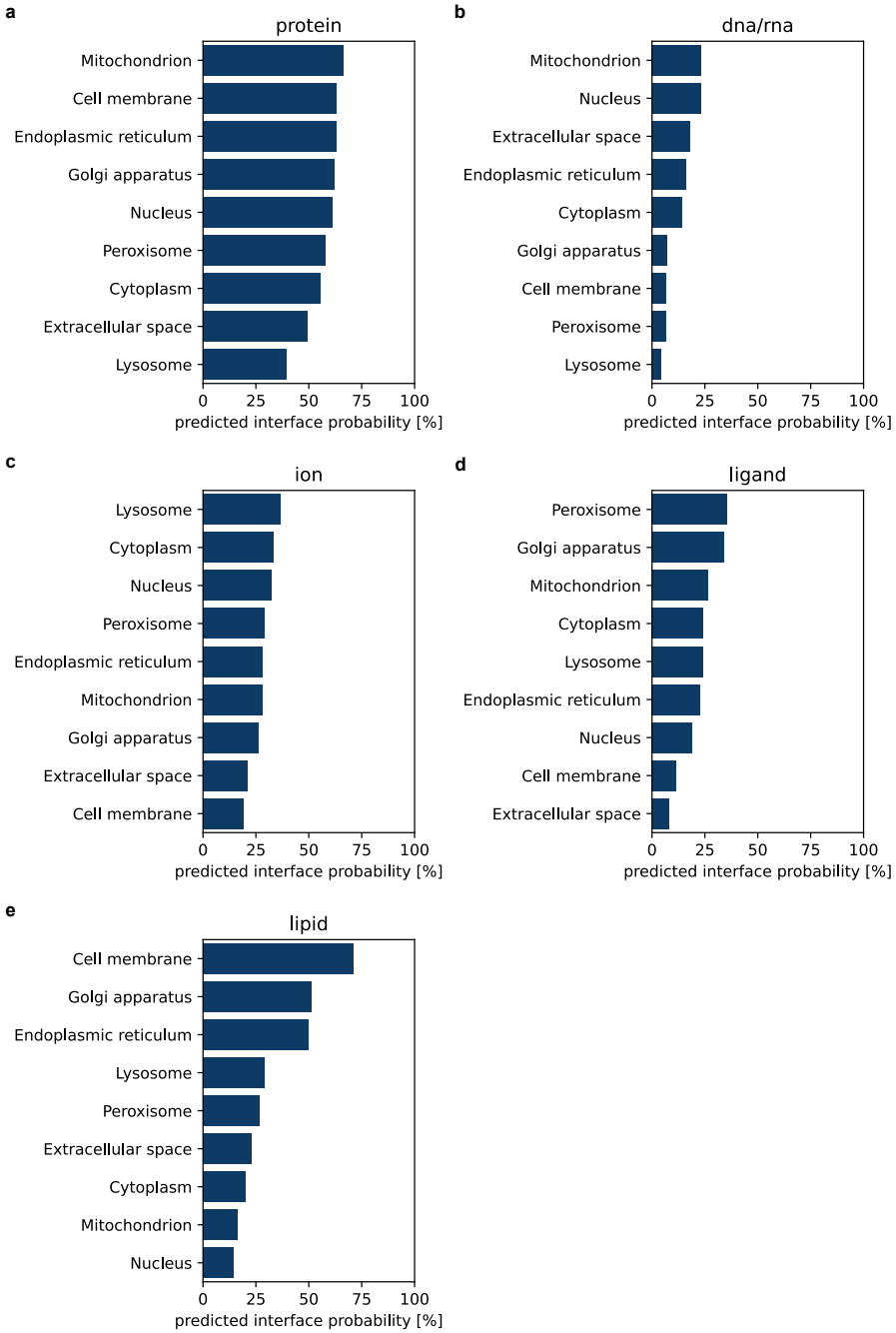

Figure 3.13: Subcellular localization. Probability of a protein within a specified subcellular localization to have an interface with (a) protein, (b) nucleic acid, (c) ion, (d) ligand, (e) lipid.

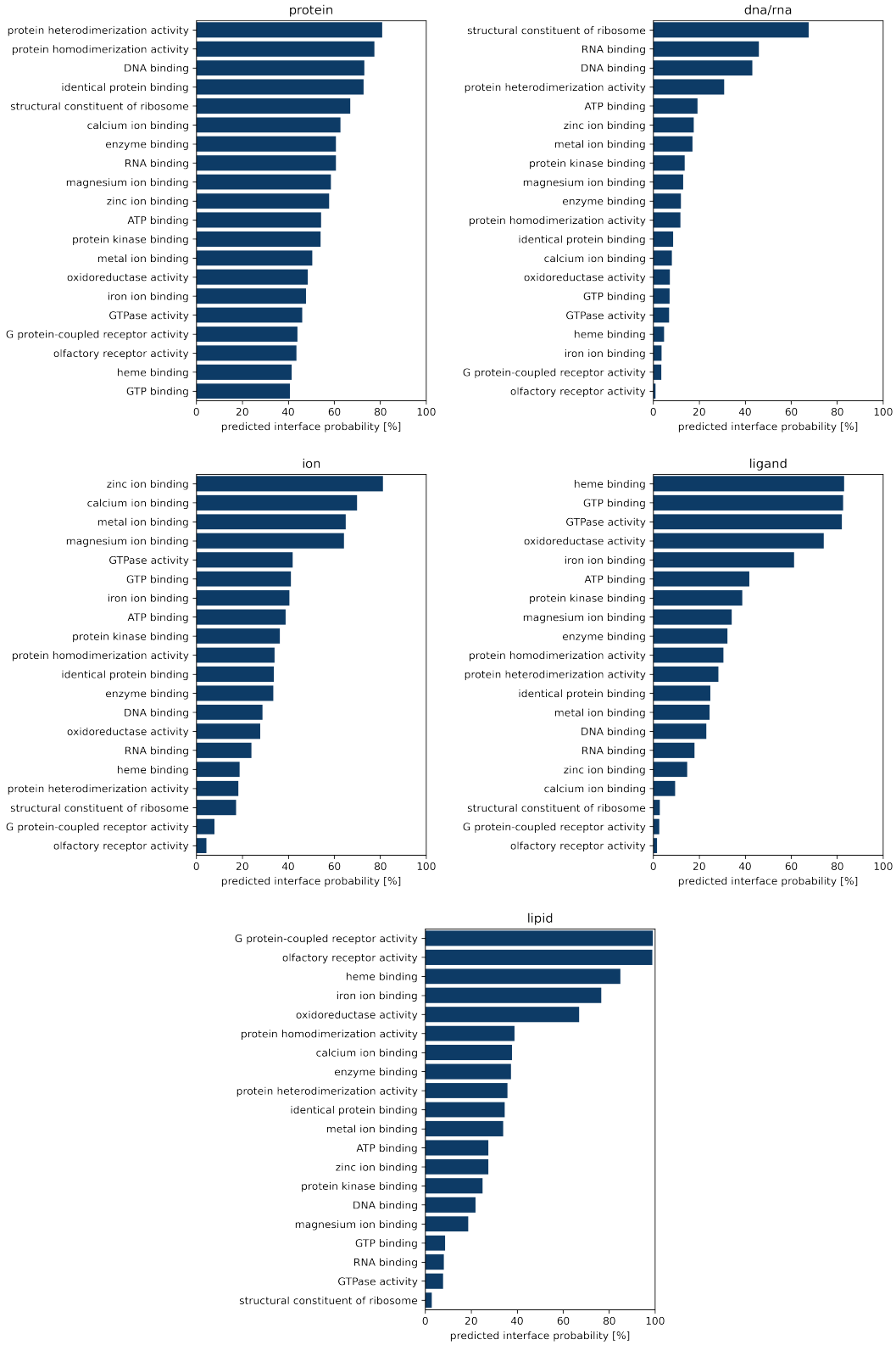

Figure 3.14: GO molecular function. Probability of protein with the specified molecular function to have a protein-nucleic acid interface (Minimum sampling of 200 examples per GO term).

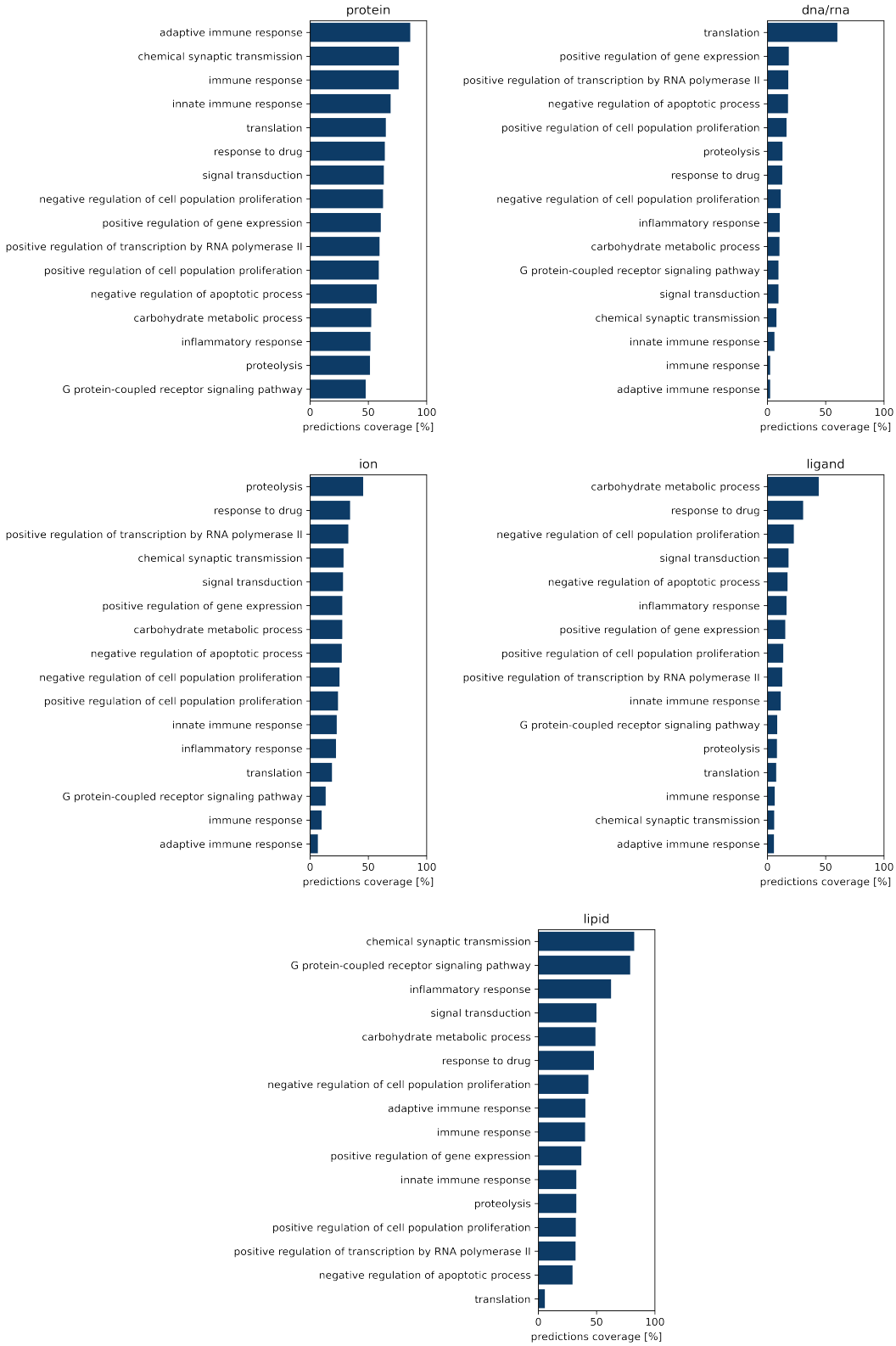

Figure 3.15: GO biological process. Probability of protein with the specified biological process to have a protein-lipid interface (Minimum sampling of 200 examples per GO term).

We interrogated further the human interfaceome for the geometrical features of the predicted interfaces and observed that when computing their solvent-accessible surface areas (SASA), interactions with proteins and nucleic acids involve the largest areas with 32 ± 22 and 29 ± 23 nm^2, respectively, while ligands and ions involve small pockets of 16 ± 7 and 7 ± 4 nm^2. The SASA distribution for protein-lipid interactions has instead a bimodal distribution that reflects specific lipid-binding sites (17 ± 9 nm^2) and large lipid coronas surrounding transmembrane protein domains (75 ± 19 nm^2, Fig. 3.16).

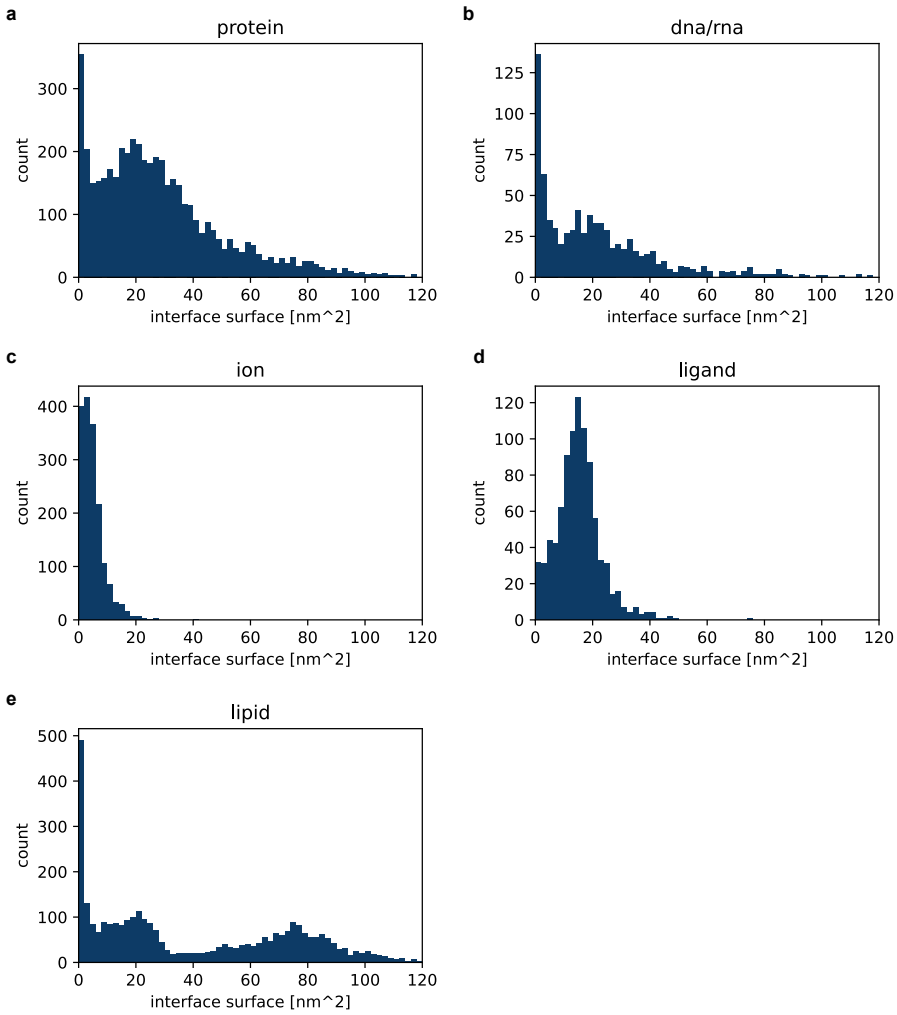

Figure 3.16: Solvent accessible surface area. Solvent accessible surface area distribution of predicted interfaces for protein (a) -protein (32 ± 22 nm^2), (b) -nucleic acid (29 ± 23 nm^2), (c) -ion (7 ± 4 nm^2), (d) -ligand (16 ± 7 nm^2), and (e) -lipid (17 ± 9 nm^2 and 75 ± 19 nm^2)

As further validation, extending the analysis to another eukaryotic proteome, we compared PeSTo predictions to the available predictions of protein binary complexes of the yeast proteome derived with AlphaFold and RoseTTAFold[76]. Also in this case, we observed a very good correlation between sets of residues involved in interfaces with the ROC AUC steadily

increasing as the analysis is limited to regions of the models of higher quality (Fig. 4d). Moreover, we identified additional binding interfaces that can extend further the interaction network of binary complexes and can be used as complementary means to better describe and model the architecture of large protein complexes (Fig. 3.17).

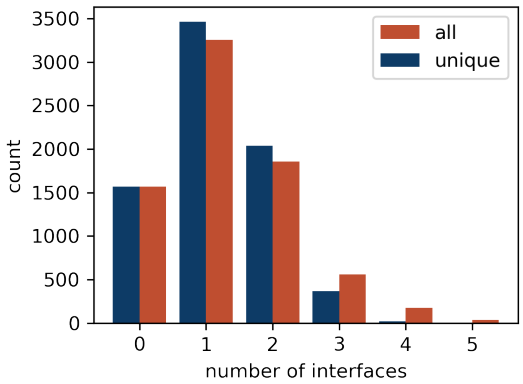

Figure 3.17: Number of interfaces. Distribution of the number of (different) interfaces per subunit. The total number of disjoint interfaces is shown as "all" interfaces count. The number of different type of interfaces (i.e. protein-protein, -nucleic acid, -ion, -ligand and -lipid interfaces) is indicated as the "unique" interfaces count.

Notably, 47% of the UniProt annotations for mutation sites fall in a predicted interface, 28% correspond to pathogenic natural variant sites, and 14% to benign natural variant sites with a baseline of 19% for random residues being within an interface (Fig. 4b). As we make all these predictions fully available in the PeSTo website and the underlying structural models are freely available in the EBI database, it is straightforward for cell biologists to consult where exactly these pathogenic mutations fall and what interactions they might compromise, in order to develop rational working hypotheses that could help further therapeutic development.

Carrying on to a large-scale analysis of the predicted interfaces, we observed strong segregation for certain kinds of interfaces and a quite large overlap for others (Fig. 4c and Fig. 3.18). An example of the former case is that of protein interfaces prone to interact with proteins or with ions/ligands, which are highly segregated. Studying these patterns further could potentially help in the discovery of allosteric regulation mechanisms. Among pairs of interfaces that feature a quite extensive overlap are those that mediate interactions with other proteins and with lipids, which could possibly point at reversible protein dimerization/oligomerization at membranes. On actual application of PeSTo to address biological questions, specific cases shall be looked at carefully, and overlaps or lack thereof might bring information as exemplified next.

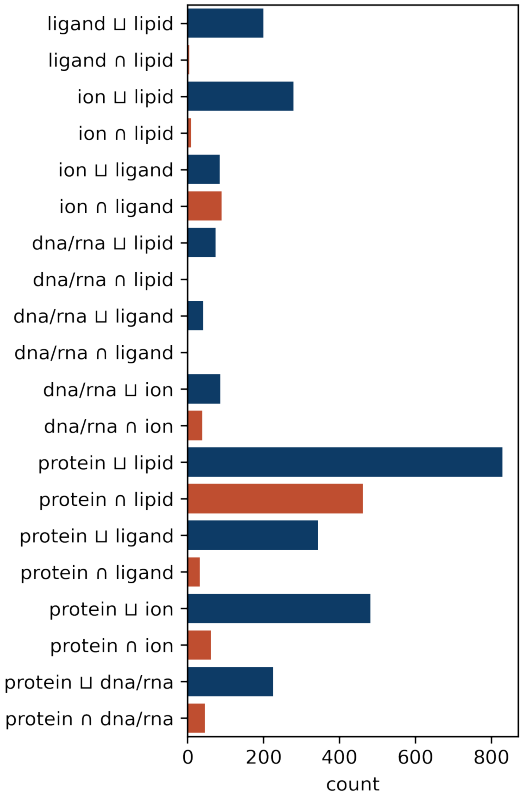

Figure 3.18: Intersecting and disjoint interfaces. Number of interfaces that contained two types of interfaces either overlapped (∩) or not (⊔).

Importantly, the availability of high-resolution structures and high-quality AlphaFold models of the human proteome, as well as other proteomes, provides biologists with the opportunity to immediately and easily interrogate specific interaction predictions of their proteins of interest, developing quickly working hypotheses, and designing new experiments, allowing in turn to discover new biology. Among multiple interesting examples, we highlight here two cases of proteins that lack structures in the PDB but where the application of PeSTo to AlphaFold models proposes clear prompts to drive forward biological studies: the human receptor for retinol uptake STRA6 (UniProt Q9BX79, Fig. 4e) and the PRAME family member 1 (PRAMEfm1, UniProt O95521, Fig. 4f).

STRA6 is modeled in the AlphaFold—EBI database as a monomer, although one would expect it to be most likely a dimer like most small-molecule transmembrane transporters. We applied PeSTo to the model as provided (i.e., as monomer) and to a model of the dimer built with AlphaFold-multimer. PeSTo predicts in both cases interfaces prone to interact with other proteins and with lipids. In the monomeric model, part of the interface predicted to interact with lipids overlaps with an interface predicted for proteins, suggesting this is the region for homodimerization within the membrane. Accordingly, this interface is not predicted for the

dimer, and the new set of residues predicted to interact with lipids makes perfect sense as the membrane-spanning region (Fig. 3.19). Another set of residues predicted to form interfaces for protein binding map to 4 locations outside the transmembrane region (Fig. 4e). On the cytoplasmic side of the membrane, three STRA6 segments with strong predicted potential for protein interaction map to a site made up of two folded elements that overlap with sequence segments that Berry et al.43 actually proposed as a binding site for regulator cellular retinol-binding protein 1 (CRBP1), next to a predicted interaction site that corresponds to a known kinase binding site (JAK2). On the extracellular side of the membrane, a binding site expected for the carrier retinol-binding protein (RBP) is also predicted. Therefore, residues with high protein interaction scores (e.g., K324-K348 for the reported RBP, L251-R257 and R638-L46 around the reported CRBP1 site, and D612-K626 for the kinase site, Fig. 4e) are potential candidates for mutagenesis studies aimed at probing the various interactions.

We finally compared protein-protein interface predictions of PeSTo with modeling protein-protein interactions using AlphaFold-multimer[48], a procedure richer in information as including also evolutionary couplings. On the STRA6 example, AlphaFold-multimer predicts binding of CRBP1 onto STRA6 around the same residues that we discuss from literature, i.e. essentially the same prediction as PeSTo. However, AlphaFold-multimer does not predict any interaction at all for JAK2 and predicts an incorrect binding site for RBP. In the case of PRAMEfm1, we detect a plausible interface for nucleic acid binding, which AlphaFold is not trained to predict, and we detect a protein interaction region of high confidence but without any information about the identity of the partners, precluding to test with AlphaFold any obvious, specific complex. These comparisons highlight a synergic intersection between PeSTo and AlphaFold-multimer for the prediction of protein-protein interactions. Namely, PeSTo can produce predictions that are consistent with the reported biochemistry, while AlphaFold-multimer can interrogate these binding interfaces when the network of interactions is known.

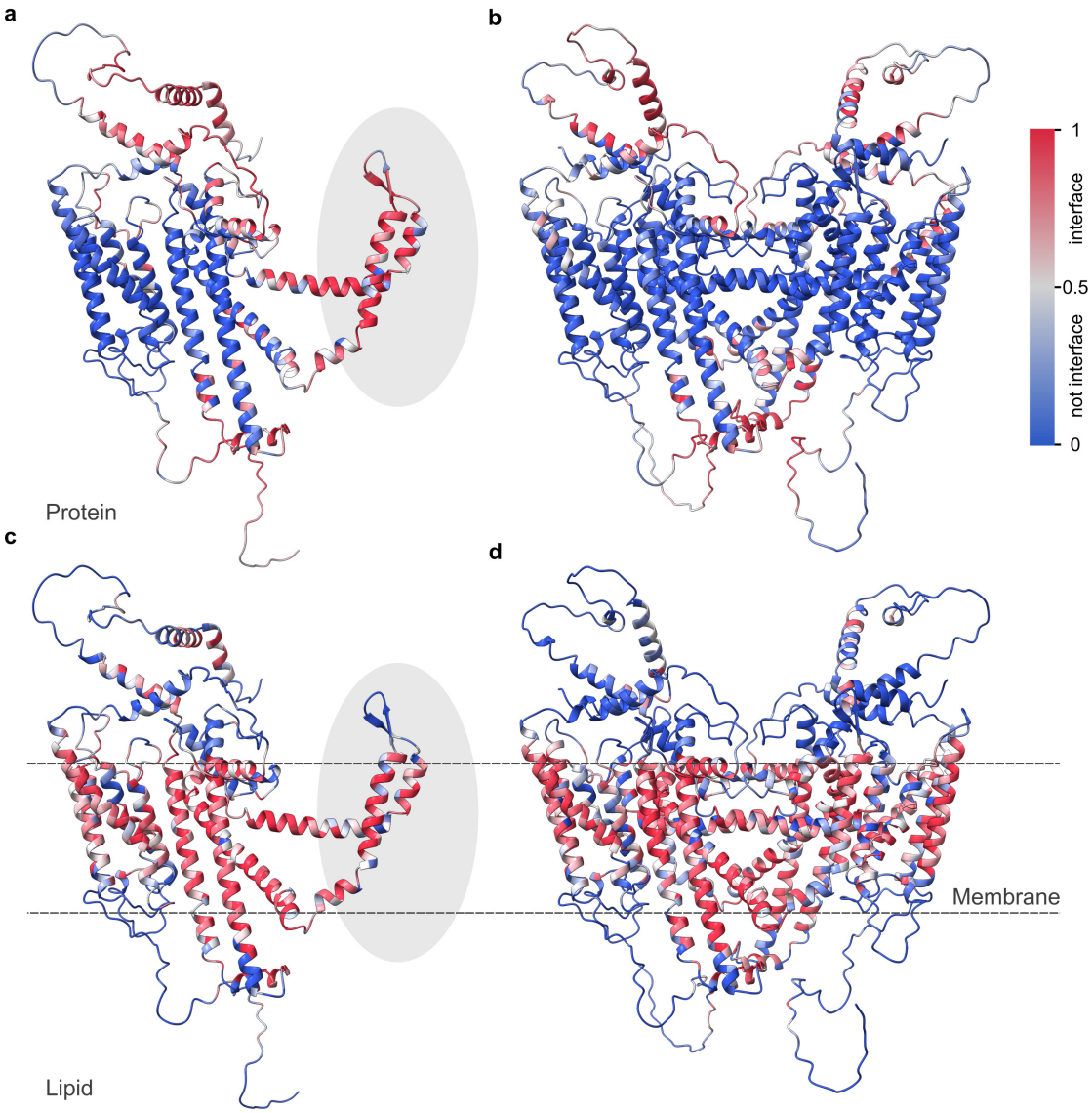

Figure 3.19: Details for STRA6 example (UniProt Q9BX79). PeSTo predicted interfaces for (a,b) protein-protein and (c,d) protein-lipid interactions with estimated membrane location. (a,c) AlphaFold predicted monomers with highlighted protein-protein and protein-lipid interfaces overlap. (b,d) AlphaFold-multimer predicted dimers.

3.2.5 Specialized protein binding interface prediction

Bibekar P., Krapp L.F., Dal Peraro M.

PeSTo-Carbo: geometric deep learning for prediction of protein-carbohydrate binding interfaces

Disclaimer

The following section is adapted from the unpublished work.

Contributions

B.P., L.F.K. and M.D.P. conceived and designed the research project. B.P. implemented PeSTo-Carbo. B.P., L.F.K. and M.D.P. analyzed the data. B.P., L.F.K. and M.D.P. wrote the paper.

Introduction

Carbohydrates are the primary source of energy for all organisms[100]. Studying the interactions between carbohydrates and protein through experimental techniques can be challenging due to their weak binding affinities[104]. Now, with the availability of large datasets containing experimentally solved protein-carbohydrate complexes[29, 105] and the rapid development of machine learning methods to learn from these data, there is a motivation for developing computational methods to study protein-carbohydrate interactions.

Results

Here, we introduce PeSTo-Carbo, a specialized application of PeSTo, trained to predict protein-carbohydrate interacting interfaces. In this case, since we are interested in a specific type of molecules, the interface is defined using a 4 Å distance threshold between an amino-acid and a carbohydrate, as described in Methods 2.4.2. The model was evaluated on 359 (with 343 carbohydrates and 16 cyclodextrins) randomly selected chains while ensuring that the sequence identity between the training and the test set at most 30%. The best model achieved a median ROC AUC of 0.92 and PR AUC of 0.54 for protein-carbohydrate interfaces. Furthermore, for protein-cyclodextrin interfaces, the model achieved a ROC AUC of 0.85 and a PR AUC of 0.28.

Further, to showcase the flexibility of our method, we also trained PeSTo-Carbo to differentiate protein-cyclodextrin interfaces specifically alongside protein-carbohydrate complexes. Cyclodextrins have been shown to stabilize proteins in liquid and dry states and inhibit the aggregation of proteins by protecting hydrophobic regions of the peptides in their apolar central cavity[106]. This makes cyclodextrins important molecules with various applications in

pharmaceutics, drug delivery, and chemical industries[107, 108]. Training the model on both carbohydrates and cyclodextrin demonstrated better performance for predicting interacting interfaces with cyclodextrin, despite the limited available training data (138 complexes with cyclodextrin). The model achieved a ROC AUC of 0.85 and a PR AUC of 0.28, showing promising performance for potential applications.

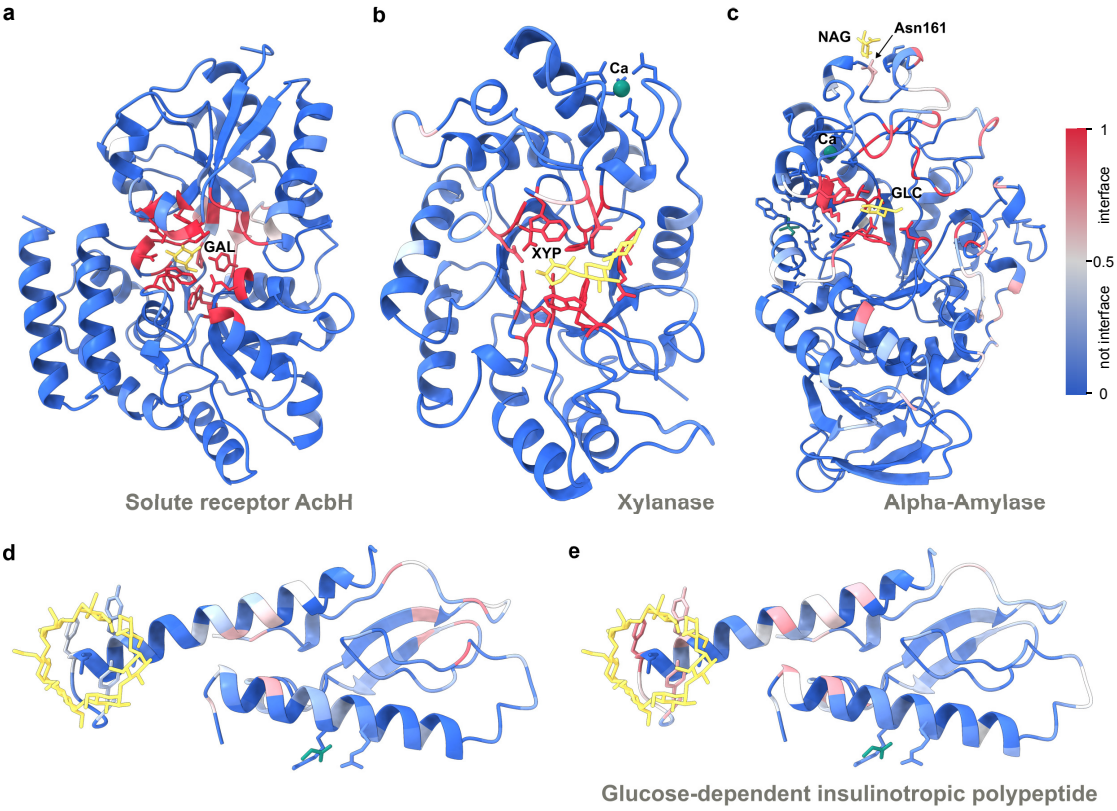

Figure 3.20: Example of protein-carbohydrate and protein-cyclodextrin interface prediction using PeSTo-Carbo. The model is applied on the protein structure alone. The confidence of the predictions is shown with a gradient of color from blue for non-interfaces to red for interfaces. The carbohydrates (yellow) and other small molecules (green) are subsequently added to assess the quality of the prediction visually. (a) Bacterial solute receptor AcbH complexed with beta-D-galactopyranose (GAL) (PDB: 3OO6). (**b**) Xylanase (XynB) complexed with beta-D-xylopyranose (XYP) and calcium ion (Ca) (PDB: 4PN2), (**c**) Alpha-Amylase complexed with alpha-D-glucopyranose (GLC) and calcium ion (Ca). The structure also contains a N-glycosylation site at Asn161 (PDB: 3VM7). Predicted protein-carbohydrate (**d**) and protein-cyclodextrin (**e**) for the glucose-dependent insulinotropic polypeptide and receptor in complex with beta-cyclodextrin (PDB: 2QKH).

To illustrate the performance of the method, Figure 3.20 shows the predicted interface with carbohydrates or cyclodextrin for some selected structures. Our model accurately predicts binding interfaces with different carbohydrates, as shown in Figure 3.20a, b, and c. It also correctly ignores non-carbohydrate binding sites, such as those with ions, demonstrated in Figure 3.20b and c. For the Alpha-Amylase protein (Figure 3.20c), the model identifies the Asparagine 161 as a carbohydrate-binding site, aligning with its known status as an N-

glycosylation site. This suggests the potential of the model for identifying glycosylation sites. In the case of the glucose-dependent insulinotropic polypeptide, we show that the model can predict specifically cyclodextrin binding, see Figure 3.20d and e. In thise case, the model predicts a binding with cyclodextrin but not with other carbohydrates.

3.3 Discussion

We showed here that a geometrical transformation of protein atomic coordinates suffices to detect and classify protein binding interfaces at high resolution, surpassing the prediction capabilities of other methods without the need of explicitly describing the physics and chemistry of the system, hence without the overhead of pre-computing molecular surfaces and/or additional properties. All this with modest computational resources and at a very high speed that enables the analysis of large structural ensembles, for example those produced by molecular dynamics simulations, which discloses the opportunity to investigate the dynamic features of protein interaction networks. Likewise, large structural datasets, like those being created by the latest generations of tertiary protein structure prediction tools, can be easily analyzed, as done here for the human foldome, with the possibility to quickly access new biological discoveries.

To make PeSTo-based predictions for proteins available to the community, we implemented it in a webserver at pesto.epfl.ch (Figure 3.21 and 3.22), accessible free of charge without registration. The server takes any protein structure and model in PDB format (uploaded or fetched from the PDB or the AlphaFold-EBI databases) and returns them with additional information reporting on the confidence of the prediction on a per-residue basis. Output files can be downloaded or visualized right within the website. Furthermore, we provide the source code as to facilitate application to large structural ensembles as done here for the human interfaceome.

Provided that sufficient training data are available, the method can be easily upgraded (as for instance to improve further protein-lipid predictions) and is reusable for other specific applications. Even in cases where data is limited, we showed that the method can be specialized: the approach demonstrated promising performance for predicting protein-cyclodextrin binding interfaces. In fact, the parameter-free PeSTo architecture is general enough that could be easily accommodated to pursue other structure-based problems such as docking or modeling interactions with materials. The description is totally agnostic to the exact physicochemical properties of the atoms in the structure, thus easily extendable to other materials and fields, and is probably also less sensitive to problems related to the starting structures such as missing atoms as compared to methods that require intermediate calculations of surfaces and volumes.

Given the ever-growing accumulation of structural information and rapid expansion of predicted foldome data, PeSTo stands as an accurate, flexible, fast, and user-friendly solution to dissect the vast and dynamic interaction landscape of proteins and can be readily used to

discover new and richer biological insights.

Figure 3.21: Homepage of the PeSTo website

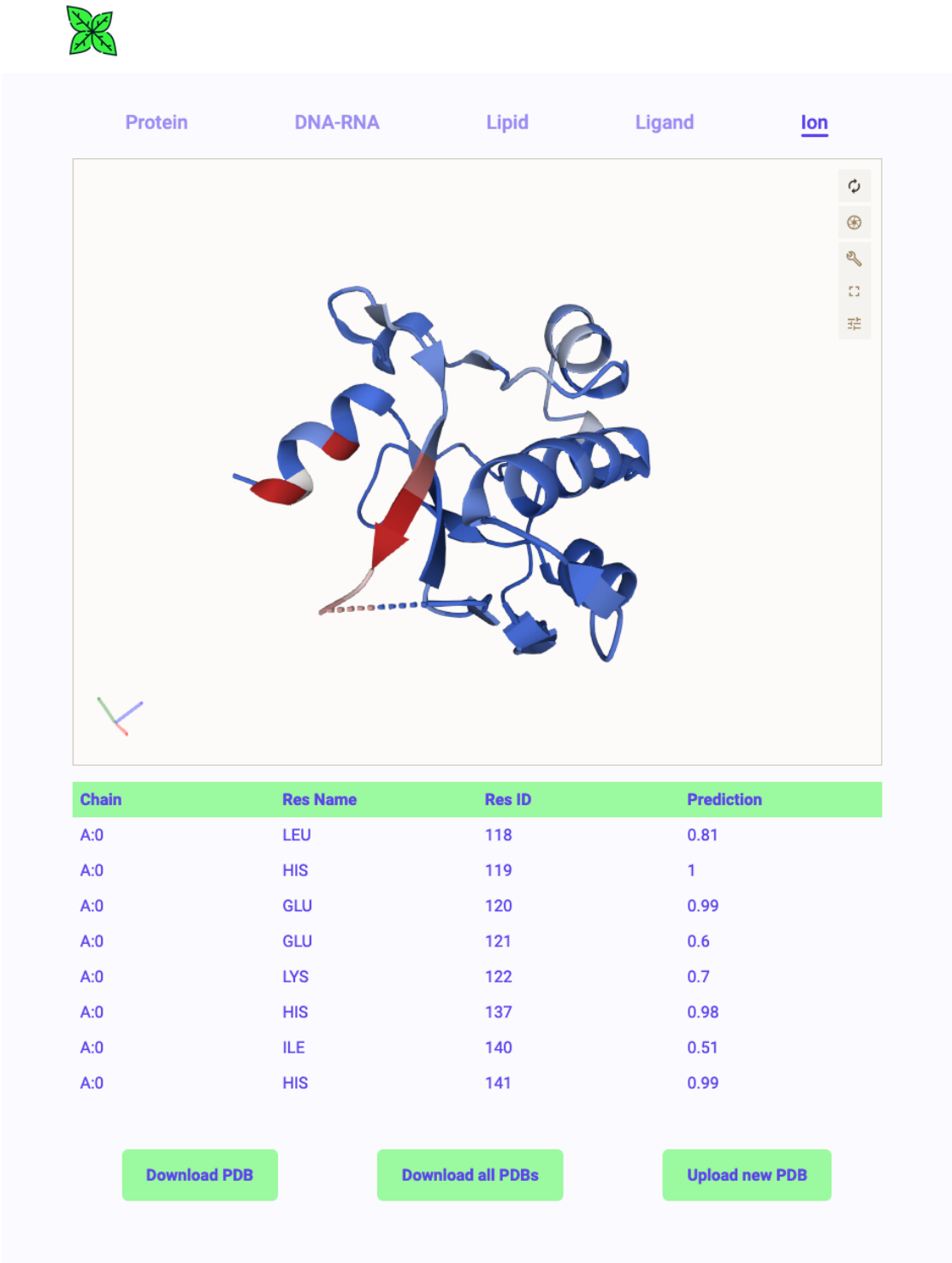

Figure 3.22: Example of results from the PeSTo website

4 CARBonAra: Context-aware geometric deep learning for protein sequence design

Krapp L.F., Meireles F.A., Abriata A., Dal Peraro M.
Context-aware geometric deep learning for protein sequence design
BioRxiv (2023).
https://doi.org/10.1101/2023.06.19.545381

Disclaimer

The following chapter is adapted from the preprint paper.

Contributions

L.F.K. and M.D.P. conceived and designed the research project. L.F.K. designed and implemented the CARBonAra code. L.F.K., F.A.M., L.A.A., and M.D.P. analysed the data. L.F.K., F.A.M., L.A.A., and M.D.P. wrote the paper.

Protein design and engineering are evolving at an unprecedented pace leveraging the advances of deep learning. Current models nonetheless cannot natively consider non-protein entities within the design process. Here we introduce a deep learning approach based solely on a geometric transformer of atomic coordinates that predicts protein sequences from backbone scaffolds aware of the restraints imposed by diverse molecular environments. This new concept is anticipated to improve the design versatility for engineering proteins with desired functions.

4.1 Introduction

Designing proteins de novo to engineer their properties for functional tasks is a grand challenge with direct implications for biology, medicine, biotechnology, and materials science. While physics-based approaches have had success in finding amino acid sequences that fold to a given protein structure, deep learning methods have recently brought a dramatic acceleration by enhancing the design success rates and versatility. Among the most recent and notable examples, ProteinMPNN, based on an encoder-decoder neural network, is able to generate protein sequences experimentally proven to fold as intended[13, 109]. More recently, coupled with denoising diffusion probabilistic models for the generation of protein backbones, ProteinMPNN in RFdiffusion has shown remarkable success[110]. In addition, ESM-IF1, based on a protein language model, is capable of generating highly diverse proteins well outside the known universe of natural sequences[56, 49]. The model has also recently found experimental validation reporting a very high success rate[111, 112, 113, 114, 115], like for example MaSIF which specializes in the design of protein interactions via learned protein surface fingerprints[14, 57].

Although these models can natively handle multiple protein chains in their inputs, and as such they can design the sequences of interacting proteins, they cannot natively consider non-protein entities within the design process, which hampers their versatility and limit their spectrum of application. Here, to address this limitation, we introduce CARBonAra (namely, Context-aware Amino acid Recovery from Backbone Atoms and heteroatoms), a new protein sequence generator model based on our recent Protein Structure Transformer (PeSTo[77]), a geometric transformer architecture that operates on atom point clouds. Representing molecules uniquely by element names and coordinates, PeSTo's transformer can be applied to and predict protein interfaces with virtually any kind of molecules, either other proteins, nucleic acids, lipids, ions, small ligands, or cofactors. Based on the same architecture, trained uniquely on structural data available on the PDB, CARBonAra predicts the amino acid confidence per position from a backbone scaffold alone or complexed by any kind of non-protein molecules. The model uses geometrical transformers to encode the local neighbourhood of the atomic point cloud using the geometry and atomic elements. It encodes the interactions of the nearest neighbours and employs a transformer to decode and update the state of each atom. By pooling the atom states from the atomic to the residue level and decoding them, the model predicts multi-class residue-wise amino acid confidences (Figure 4.1a and

Methods 2.4.3). CARBonAra thus provides a potential sequence space that can be refined through the incorporation of specific constraints, such as a molecular context critical to the protein's function, a particular objective, or provided allowed conformations. CARBonAra offers a novel level of flexibility in protein design by recognizing and incorporating any molecular context into its sequence predictions. This distinctive capability of our method expands therefore the scope of applications in the field of protein design.

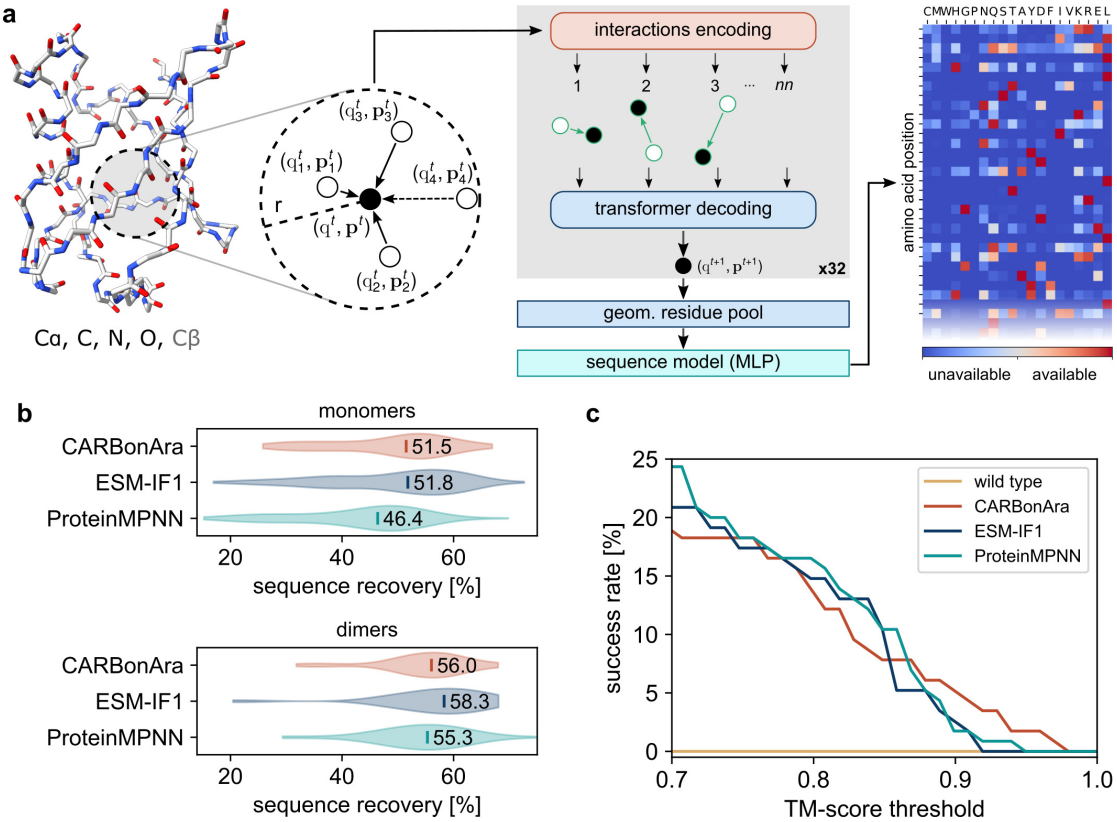

Figure 4.1: CARBonAra architecture and comparison with other state-of-the-art methods. (**a**) The model applies multiple geometric transformer operations to the coordinates and atom element of a backbone scaffold with added virtual Cβ to predict the amino acid confidence at each position in the sequence. (**b**) Comparison of the sequence recovery of different methods for monomers and dimers with indicated median sequence recovery. (**c**) Percentage of AlphaFold predicted structures, in single sequence mode, above a TM-score threshold.

4.2 Results

4.2.1 Inverse folding benchmark

CARBonAra performs on par with state-of-the-art methods like ProteinMPNN and ESM-IF1 for sequence prediction of isolated proteins or protein complexes (Figure 4.1b), while having a similar computational cost taking only a few seconds per run (~3 seconds). Our method achieves a median sequence recovery rate of 51.3% for protein monomer design and 56.0%

for dimer design when reconstructing protein sequences from backbone structures. Moreover, the success rate of the generated sequences using AlphaFold in single-sequence mode is commendable, especially in generating structures with a TM-score above 0.9 (Figure 4.1c).

In order to better characterize the model, we analyzed the interpretability of the prediction's confidence. The Pearson correlation of 0.88 between prediction confidence and the sequence recovery rate suggests that model's confidence can be a reliable indicator of prediction quality (Figure 4.2a). Next, we analyzed the relationship between prediction confidence and the likelihood of the prediction being accurate across all amino acids (Figure 4.2b). Based on these insights, we derived a score from the prediction confidence. This score effectively quantifies the accuracy of the prediction of the model. Our analysis revealed a linear relationship between this interpolated score and the recovery rate (Figure 4.2c). Thus, the score offers an estimate on the quality of a generated sequence.

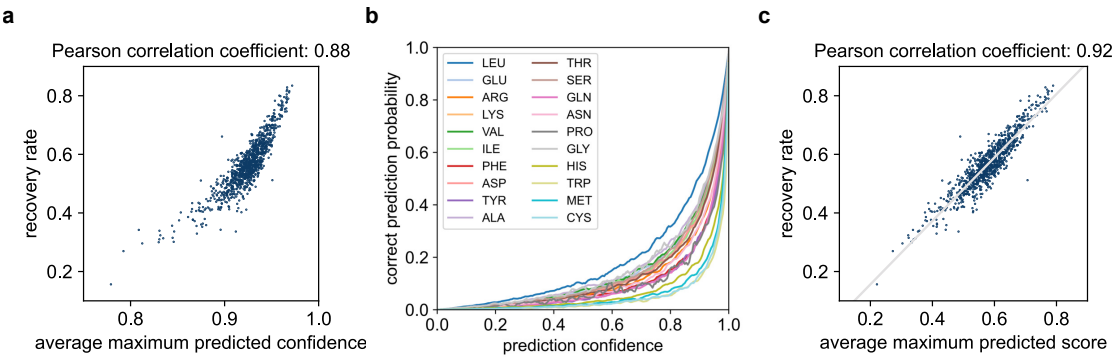

Figure 4.2: Prediction confidence analysis. (a) Recovery rate as a function of the average maximum prediction score (943 structures from the testing dataset). (b) Relationship between prediction confidence and the prediction accuracy for each amino acid type (4096 subunits from the training dataset). (c) Rescaling prediction score into a prediction confidence correlated with the probability to be correct (943 structures from the testing dataset).

4.2.2 Flexible sequence sampling strategies

In contrast to other methods, CARBonAra uses multi-class amino acid predictions that generate a space of potential sequences, opening various possibilities for sequence sampling. For example, one can tailor sequences to meet specific objectives, such as achieving maximal or minimal sequence identity, or low sequence similarity in order to design unique sequences with a specific fold (Figure 4.3, see also Methods 2.4.3).

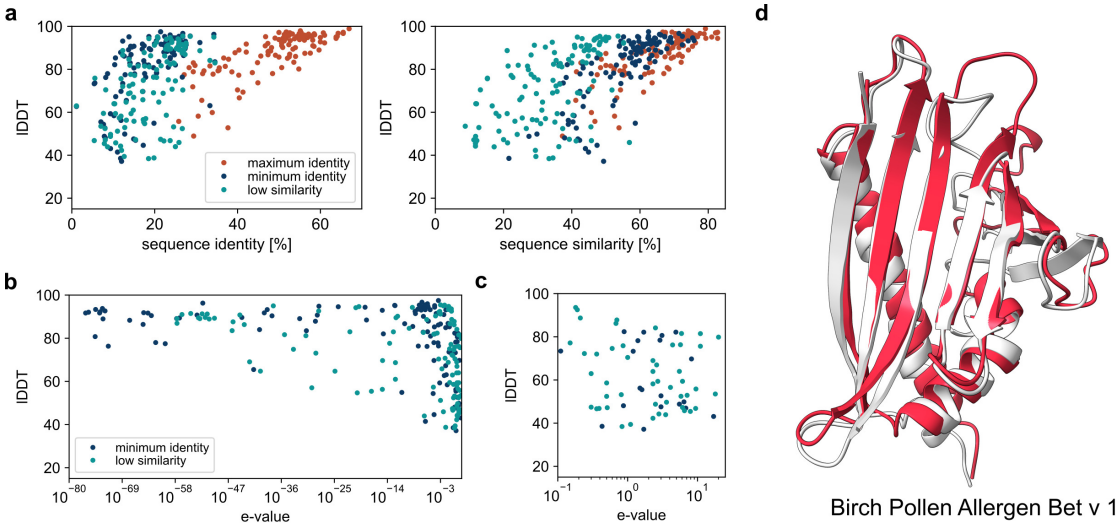

Figure 4.3: Analysis of different sequence sampling approaches using AlphaFold with MSA. (**a**) Local Distance Difference Test (lDDT) of AlphaFold predicted structures against scaffold monomers from sequences generated using CARBonAra with, as objective, maximum sequence identity, minimum sequence identity, and low sequence similarity. (**b**) lDDT of the AlphaFold predicted structures as a function of the expect value (E-value) of the generated sequences. (**c**) Close up on the generated sequences with a high E-value. (**d**) Using the birch pollen allergen Bet v 1 protein (PDB: 6R3C) as a scaffold, in white, a new sequence was generated with a low sequence similarity as objective. The AlphaFold predicted structure, in red, has a lDDT of 70 with the reference. The generated sequence has a 7% identity and 13% similarity with the original scaffold protein. When compared to the reference, the predicted structure has a lDDT score of 70.

We observed that the model is able to learn the tighter amino acid packing at protein cores thus resulting in higher recovery rates and fewer amino acid possibilities for buried amino acids (Figure 4.4a-c). As such, CARBonAra confidently recovers core amino acids while demonstrating greater flexibility on the protein's surface, unless additional functional or structural constraints are provided.

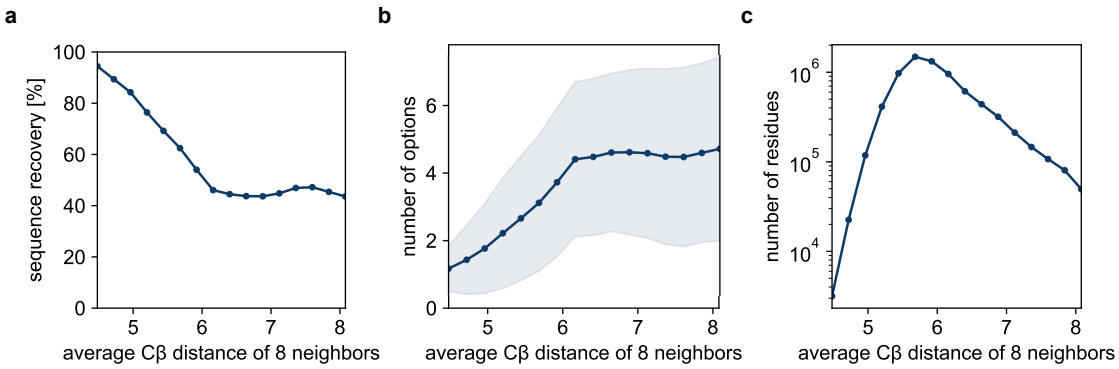

Figure 4.4: Analysis of buried against surface amino acids. (a) Sequence recovery, (b) number of predicted options per position and (c) number of residues as a function of the average Cβ distance of the 8 nearest neighbours (18866 structures from the testing dataset).

An informative way to refine the sequence space uses dynamics as a constraint. By applying CARBonAra to structural trajectories from molecular dynamics (MD) simulations, we were able to improve sequence recovery, especially in cases that previously showed low recovery rates (Figure 4.5). Simultaneously, we observed a reduction in the number of possible amino acids predicted per position. This further limit the sequence space and could enable the design of targeted structural conformations.

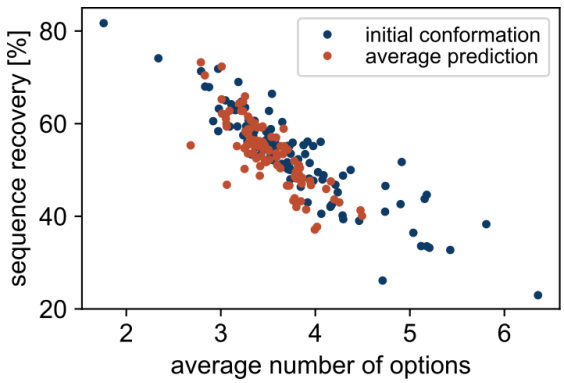

Figure 4.5: Effect of conformations changes on recovery rate. Comparison of the sequence recovery between the predicted sequence on crystal structures and the consensus sequence predictions derived from 500 frames sampled from 1 μs molecular dynamics simulations for 80 monomers.

4.2.3 Context-aware sequence generation

More importantly, leveraging PeSTo's architecture, this model has the new ability to perform protein sequence prediction conditioned by a specific non-protein molecular context. On a test set similar to the one used for PeSTo, we show that the overall structure median sequence recovery increased from 54% to 58% (Figure 4.7) when an additional molecular context is provided. In particular, CARBonAra achieves median sequence recovery rates at the interface of 56% when protein interacting partners are considered and 55% when nucleic acids are used as interfacial restraints, providing a significant improvement over predictions without context (Figure 4.6a). Similarly, the recovery rate at the interface improved significantly if small-molecule entities such as ions (67%), lipids (57%), and ligands (60%) are included (Figure 4.6a). Including these molecules boosts sequence recovery in their surroundings, and reduces the number of amino acid possibilities to sample from (Figure 4.6b). This shows CARBonAra's power to properly craft the residue types required for the binding of specific molecules.

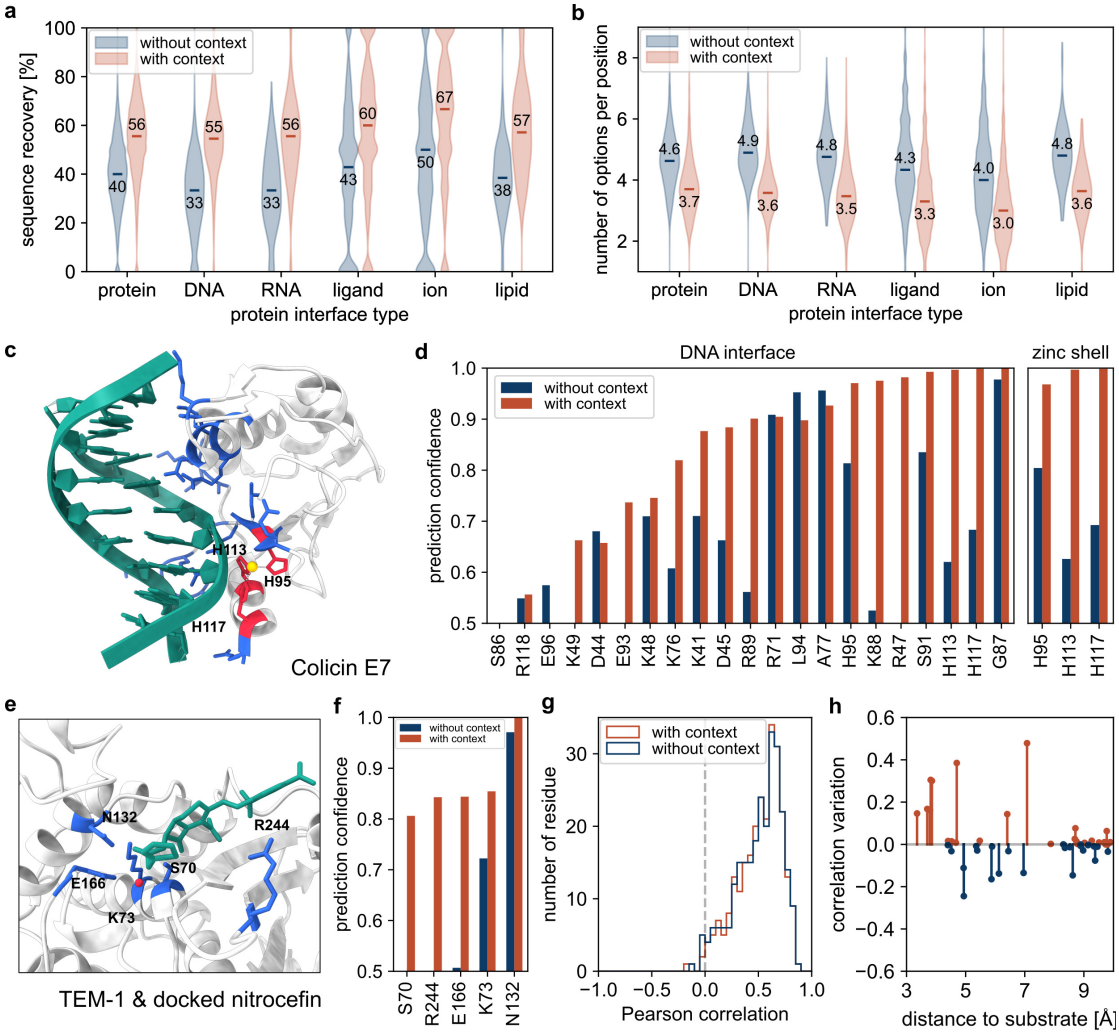

Figure 4.6: Context-aware amino acid recovery extends to various biomolecules. (**a**) Sequence recovery at the interface (residues within 5 Å) without and with proteins, nucleic acids, ligands, ions, and lipids binders. (**b**) Number of predicted possible amino acids per position at the interface (residues within 5 Å) without and with proteins, nucleic acids, ligands, ions, and lipids binders (considering a confidence prediction threshold of 0.5). (**c**) Colicin E7 endonuclease domain in complex with DNA and a zinc ion (PDB: 1ZNS). The protein-DNA interface (residues within 4 Å) is highlighted in blue. The protein-zinc shell is highlighted in red (residues within 3 Å). (**d**) Estimated accurate prediction probability for the scaffold amino acids at the protein-DNA interface and the protein-zinc shell with and without the presence of DNA and zinc. (**e**) Nitrocefin docked in the active site of the β-lactamase TEM-1 (PDB: 1BT5). Relevant residues for substrate recognition and hydrolysis are shown in blue, nitrocefin in green, and the catalytic water molecule in red. (**f**) Prediction confidence with and without the substrate for the relevant amino acids for binding. (**g**) Correlation of the predictions with deep sequencing analysis of TEM-1. (**h**) Correlation variation by adding the context (nitrocefin and catalytic water) for the amino acids close (in Cβ distance) to the substrate.

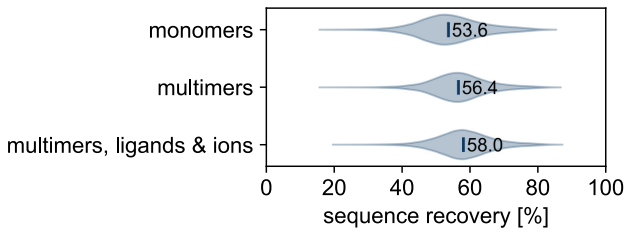

Figure 4.7: Benchmark of different use cases. Sequence recovery distribution for systems of monomers, multimers and any biomolecules (18866 structures from the testing dataset). The median sequence recovery is indicated for each case.

An exemplary case to illustrate the power of this approach is the endonuclease domain of ColE7, which interacts with duplex DNA in a zinc-dependent manner[116]. The sequence recovery rate obtained by CARBonAra showed a significant increase from 29% to 52% at the metal and DNA interfaces when the zinc ion or the 12-bp DNA duplex was included as resolved in the native structure (Figure 4.6d). Thus, imposing the presence of non-protein interacting interfaces can enhance the sequence recovery rate significantly, also with respect to predictions done by ProteinMPNN (24%) and ESM-IF1 (43%). Interestingly, when a non-native molecular context is provided such as a larger ion (e.g., calcium) the sequence recovery rate decreased (Figure 4.8). Thus, the predicted amino acid confidence of an ion pocket is widely dependent on the given context, as illustrated also for the metallo β-lactamase BJP-1 (Figure 4.9).

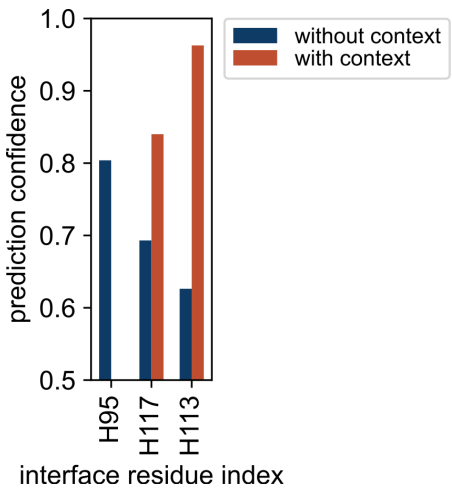

Figure 4.8: Effect of changing the ion type on the prediction. The prediction confidence for the three most important amino acids for ion binding in the case where the zinc ion of Colicin E7 is replaced with a calcium ion.

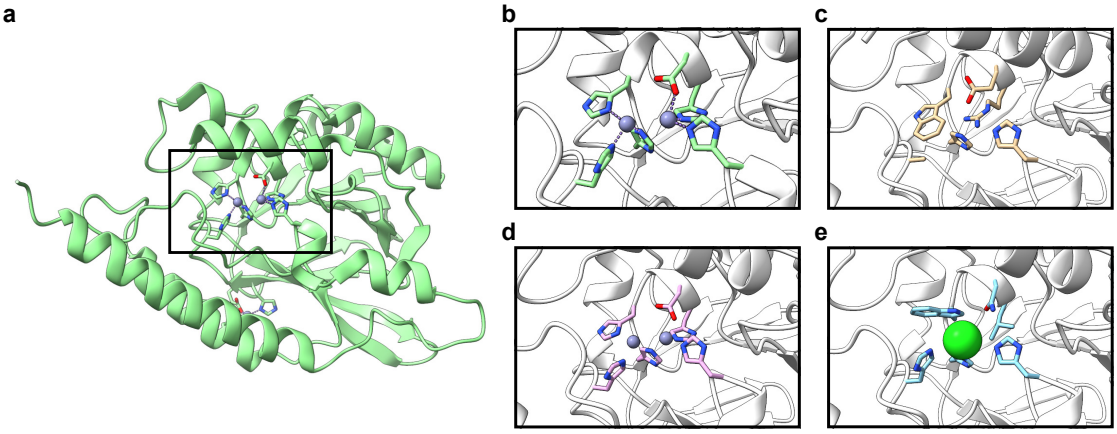

Figure 4.9: Ion binding pocket design. Effect of the ion context on the optimal predicted sequence in the case of a metallo β-lactamase zinc binding pocket. (a) Metallo β-lactamase structure with a pocket containing two zinc ions (PDB ID: 3LVZ). (b) WT pocket of the metallo β-lactamase. Pocket of an AlphaFold predicted structure with a designed sequence applied to the scaffold structure without zinc ions (c), containing the original zinc ions (d) and containing a manually placed chloride ion (e).

Relevant for enzyme design is the possibility to design sequences under the restraints provided by a desired substrate or high-affinity ligand. To test this case, we explored CARBonAra's ability to predict the sequence of a TEM-1 β-lactamase-like enzyme when the native context at the active site is provided (Figure 4.6e). Without context, the catalytic S70 and substrate binding R244 are never predicted positively (confidence of 0.39 and 0.11 respectively, Figure 4.6f), however, when the prediction is done with a β-lactam (here nitrocefin) docked at the catalytic pocket, the catalytic triad S70, K73, and E166, along with key residues necessary to β-lactam binding (i.e., N132, R244) all have a high prediction confidence (> 0.8) and low ranking (top 2) (Figure 4.10). Importantly, in this case, the sequence recovery is maximal when also the catalytic water is considered, hinting at a very high sensitivity for the molecular context.

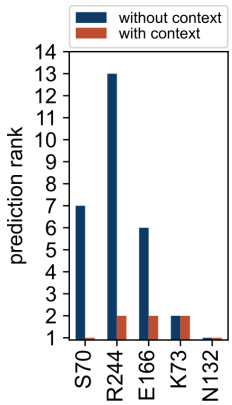

Figure 4.10: Effect of the docked nitrocefin and catalytic water in TEM-1 on the prediction ranking. Rank of the prediction from maximum to minimum confidence for the 5 important amino acids at the pocket without and with the docked nitrocefin and catalytic water.

Given that TEM-1 β-lactamase has been widely studied, we took the occasion to probe what information CARBonAra's residue-wise amino acid probabilities provide when compared to experimental data. We correlated the estimated probabilities to the residue-wise amino acid probabilities measured experimentally through deep sequencing of a saturated mutagenesis library of the TEM-1 β-lactamase[81] (Figure 4.6g). We observed an average correlation of 0.51 ± 0.21 for CARBonARa with deep sequencing data, which is similar to the correlation between the deep sequencing data with the multiple sequence alignment of this enzyme's family (0.52 ± 0.22). This shows that CARBonARa's estimated probabilities can capture functional sequence variability, a central topic in the realm of protein evolution[117, 118]. Moreover, we observed that adding the context to the active site of TEM-1 (i.e. docked nitrocefin and the catalytic water) improved the correlation locally (i.e. for amino acids within 5 Å) but also affects the predictions of amino acids further away (up to 10 Å). These results hint at the possibility to use CARBonAra for the study of the effect of a specific context locally as well as their long-range influences (Figure 4.6h).

4.2.4 In silico de novo structure design

We then applied our sequence prediction method, CARBonAra, to de novo protein design with a specific aim to create a non-natural protein fold. Our target was a triangular structure made out of alpha helices, featuring a holo core rather than the typical hydrophobic core found in most proteins. A rough template of the desired structure was initially prepared manually and subsequently refined using Foldit[119]. Once refined, CARBonAra was used to predict the sequence for this structure. We validated this sequence by generating its corresponding structure in-silico using AlphaFold.

The design process was iterative, involving multiple cycles of prediction and refinement. Each cycle used CARBonAra for sequence prediction, AlphaFold for in-silico structure validation, and Foldit for structural refinement. We continued this iterative process until the structure reached a high predicted lDDT score with AlphaFold in single sequence mode. The final structure, as shown in Figure 4.11, successfully captures the intended non-natural shape, showing, in-silico, the effectiveness of our approach.

We used CARBonAra to sample sequences and subsequently folded them using AlphaFold in single sequence mode. This process was iteratively performed until we obtained five sequences that exhibited a pLDDT score above 80 and conformed to the target triangular shape. The five best sequences were then subjected to experimental testing to assess their viability. The results confirmed that all five sequences were expressible and soluble. Further validation was conducted using circular dichroism spectroscopy. The data from the spectroscopy showed that all five constructs are predominantly helical in their secondary structure. This aligns well with the in silico results, however more experimental testing should be performed to very that the fold corresponds to the computational prediction.

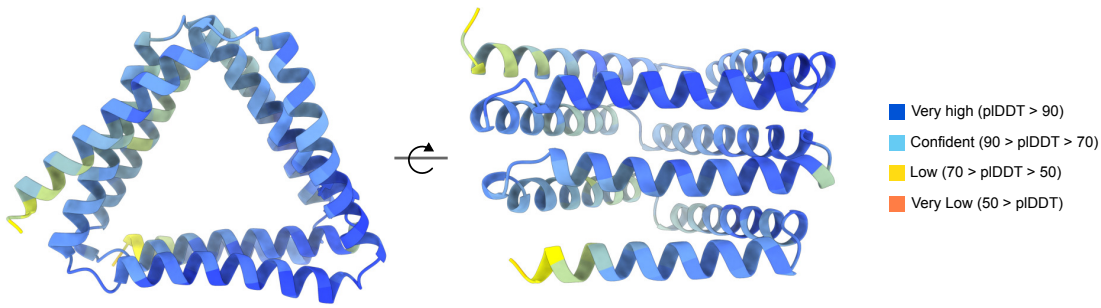

Figure 4.11: AlphaFold predicted structure of a de novo designed triangle protein. Fernando Meireles built the template triangle structure manually and using Foldit. From this initial template, we generated an optimized sequence using CARBonAra. We show here the AlphaFold predicted structure in single sequence mode (average predicted lDDT of 88.9).

4.3 Discussion

CARBonAra offers a new approach to predicting protein sequences based on their backbone geometry. By using a geometric transformer architecture, it can operate in a structure-centric manner across different molecular contexts.

The method is flexible in its sampling strategies. It can operate through direct sampling methods for straightforward applications. Additionally, it is context-aware and can adjust predictions based on interactions with not only proteins but also other types of biomolecules like nucleic acids, lipids, ions, and small molecules. This added layer of context can improve the accuracy of sequence predictions. There's also potential for using dynamics as a constraint in sampling, offering a more realistic representation of molecular behavior.

Pairing the capabilities of CARBonAra with modern diffusion models for backbone conformation sampling opens new opportunities for designing protein-based materials and therapeutics. Overall, CARBonAra broadens the scope of computational approaches in the field, offering both enhanced predictive accuracy and broader applicability.

5 Conclusion and perspectives

5.1 Summary

In summary, this work represents significant advancements in three main domains: new method development in deep learning, binding interface prediction, and sequence prediction from backbone scaffolds.

Method Development

We have successfully defined, developed, and implemented a novel neural network operation known as the Geometric Transformer. This operation harnesses the properties of global translation and rotation symmetry, ensuring that the Geometric Transformer is both translation invariant and rotation equivariant. This unique characteristic offers robustness in processing any structures. Further enhancing the capabilities of our method, we introduced a new self-attention operation, the Geometric Pooling. This operation plays a crucial role in reducing and embedding the point cloud state, making it possible to transition from an atomic to a residue point cloud representation.

Building upon these foundational operations, we then established an architecture named PeSTo, an acronym for Protein Structure Transformer. One of the standout qualities of PeSTo is its inherent simplicity. It operates directly on structural data without necessitating any preprocessing and effectively processes any point cloud of atoms. This direct approach not only streamlines the prediction process but also minimizes potential sources of error introduced during preprocessing. Lastly, the modularity of PeSTo is one of its significant strength. Its design ensures that it can be easily adapted and applied across a variety of tasks, highlighting its general applicability in structural biology.

Binding interfaces prediction

PeSTo has proven itself as a formidable tool in predicting the binding interfaces of proteins with a wide range of biomolecules. Notably, this includes interactions with other proteins, nucleic acids, ions, ligands, and lipids. Comparative analysis revealed that PeSTo surpasses existing methods, especially in predicting protein-protein interfaces, placing it at the forefront in this domain. We also demonstrated that PeSTo is able to predict binding interfaces with limited data. This capability was highlighted through its accurate predictions involving lipids, carbohydrates, and notably, cyclodextrins.

In terms of scalability and applicability, we challenged PeSTo by employing it to predict interfaces for a vast number of structures produced by AlphaFold. The results of these predictions is what we have termed the "interfaceome", a comprehensive database of interfaces. The creation of the interfaceome not only attests the efficiency of PeSTo efficiency but also positions it as a powerful tool for extensive protein interaction analyses.

Inverse folding problem

In our studies, we also demonstrated that PeSTo can be used for more than just protein interface prediction. We applied it to the inverse folding problem and developed a method named CARBonAra for this purpose. CARBonAra was shown to perform as well as other leading methods in predicting sequences from backbone scaffolds. We evaluated, in-silico, the method using AlphaFold and AlphaFold-multimers: by predicting the structure based on the sequences generated with CARBonAra. One key feature of CARBonAra is its ability to work with both protein and non-protein molecules. This is because PeSTo can handle any type of atomic point cloud. When given information about the structural context, CARBonAra can adjust and improve its predictions. Finally, we found that CARBonAra can work with a variety of structural contexts, ranging from proteins and nucleic acids to ligands, ions, lipids, and water molecules.

5.2 Importance and Implications

Conceptual significance

One important conceptual significance of this work is the strategic choices made in structure representation and model architecture. One primary consideration was to keep the structural description as general as possible. The aim behind this was twofold. First, a general framework has the intrinsic capacity to be applied across a broad array of applications. This breadth is advantageous as it increases the method's utility in diverse research areas. Second, having a flexible framework allows for adaptability. In situations where a particular problem proves too challenging to solve with the available dataset, the model can be easily repurposed for a different task. This provides a level of robustness and ensures that the work remains rel-

evant and applicable, even as new challenges emerge in the field of structural biology.

To demonstrate this flexibility and broad applicability, we successfully applied the architecture to two distinct but equally challenging problems: predicting binding interfaces and predicting sequences from a protein backbone. This not only validated the approach but also showcased its versatility, substantiating the concept that a well-designed, general model can indeed serve multiple purposes effectively.

It is worth emphasizing the type of representation chosen for atoms, as it plays a central role in the effectiveness and versatility of the model. The selected atomic description is intuitively aligned with physics. Each atom is described by a scalar state, which can encapsulate various properties like charge and mass. Additionally, every atom is associated with a vector state that can contain information about velocity, momentum, spin, and so on.

This choice in representation does not just stop at basic atomic properties; it also captures geometrical specifics. Bond angles and dihedral angles, for instance, often have energetically more favorable conformations. The vectors are designed to hold this geometric information without overall additional complexity.

This atomic and geometric representation can also be exploited for further research and application. One intriguing avenue is to use these vector states as inputs for the geometric transformer. For example, incorporating velocity information into the model could be particularly useful for accelerated molecular dynamics. Moreover, the output vector states of the geometric transformer could be useful for sampling different conformations. The output vector state could also be applied for the fitting or refining of atom coordinates in a Cryo-EM density map. For this, we could adapt the geometric transformer operation to integrate the density based information, adding another track to the architecture for predicting the coordinates of the atoms in the Cryo-EM map. The vector input and output open up new possibilities for the application of the model in more complex and nuanced tasks in structural biology.

Implications of the general interface prediction

While we have generalized the prediction to include a variety of biomolecules, there is room to push the boundaries further. For protein-protein binding interfaces, our preliminary experiments indicate promising results in identifying specific amino acids that come into contact at the interface. This application could shed light onto the nature of protein-protein interactions. Specifically, it could be employed to analyse the specificity or non-specificity of certain amino acids within an interface and identify key residues that play a crucial role in the binding mechanism.

Building on this concept, the framework can also be adapted to predict detailed protein binding interfaces with nucleic acids, such as DNA and RNA. This extension could revolutionize our understanding of protein-DNA and protein-RNA interactions. For instance, we could foreseeably predict specific binding motifs, thereby adding another layer of granularity to the

existing body of research. Such detailed predictions open new avenues for the exploration and understanding of complex biological systems.

Broader impact of this work

The conceptual reach of this work is not limited to proteins; it invites a more comprehensive view of biomolecules. For example, using CARBonAra, the sequence prediction model, we can in theory design protein binders that interact specifically with DNA or RNA. These binders can be fine-tuned to optimize specificity or non-specificity of the binding motif. Additionally, the problem can be easily reversed to predict DNA or RNA sequences based on a specific backbone structure of DNA or RNA in the context of a known protein.

A key strength of the architecture is its agnosticism toward the type of biomolecule. It only requires atomic elements and coordinates for its calculations, making it broadly applicable. More specifically, the model can be applied to non-standard amino acids and understand the nuances in chemical differences of post-translational modifications. This feature allows for the approach to be extended to more complex biomolecules like glycans.

Furthermore, the agnostic nature of the architecture of the model allows its application to extend beyond the realm of biomolecules. The same principles could be employed in various disciplines including chemistry, material science, and physics. This versatile approach offers a powerful tool for studying a wide array of molecular interactions, paving the way for future advancements in multiple scientific fields.

Future directions in drug discovery

The potential applications of this work in drug discovery are particularly promising. Although drug discovery is a multifaceted and highly competitive field, there are more straightforward problems where the architecture can make immediate contributions. The first important question, this application can help solve is the identification of cryptic pockets in proteins. Training the model for this specific task could provide a valuable tool for screening new targets for drug interaction, a pivotal step in drug development.

Second, the architecture can be extended to predict interfaces with specific fragments of ligands. By training the model using a loss inspired from the Contrastive Language–Image Pretraining (CLIP), it becomes possible to scan a large library of compounds to identify potential drug candidates. This could result in methods that are more efficient than currently existing ones, offering a more robust toolset for drug design. Overall, the technology has the capacity to advance not only the field of structural biology but also to make significant strides in the ever-important domain of drug discovery.

Addressing more complex issues in drug design requires identifying the current weaknesses of existing methods. In my view, the central bottleneck is the accurate modeling of interac-

tions between molecules. Protein structures are inherently dynamic and function within a specific biological context. This adds layers of complexity when searching for targets; if the optimal conformation space of a specific target is not known or available, the accurate computation of binding affinity becomes challenging.

Another key issue lies in the docking of ligands and scoring their likelihood of binding in a given conformation. Here, an intriguing solution could be the computational co-folding of proteins with ligands, aimed at achieving accurate protein-ligand docking. The PeSTo architecture, relying on atomic elements, coordinates, and the molecular topology, could potentially be adapted for this purpose. Such an algorithm could engage in first-principles structure determination both folding the protein and resolving other molecules without the need for external information such as Multiple Sequence Alignment (MSA). The details of the proposed approach will be elaborated on in a subsequent section.

Potential of this foundational method

The geometric pooling operation we introduced, which reduces information from the atomic to the residue level, is highly adaptable and can be generalized for varying degrees of compression. This opens the door for more extensive structure embedding, beyond just the residue level. In practice, the geometric transformer and geometric pooling operations can be interwoven in series. By doing so, the architecture is capable of incrementally processing and compressing structural information down to a single geometric point.

This architecture essentially forms the encoder module of a full autoencoder setup. The utility of such an encoder extends beyond our initial applications; it could be particularly useful for predicting global properties of a structure. Examples include thermal stability, biological function, or even the subcellular localization of a given protein. Thus, the extension to an autoencoder architecture amplifies the range of problems that can be addressed in structural biology.

Next, the decoder module can be build using geometric unpooling operations that allows for the progressive decompression of the encoded information to recover the original structure. More specifically, I suggest that the unpooling operation is dependent on the geometric information encoded within the vector states and can be structured in a U-net style architecture for effective decoding. I propose to employ an attention mechanism within this decoding phase. In this setup, the scalar states of the compressed state serve as the keys, the vector states of the compressed state act as the values, and the scalar states from the prior uncompressed state serves as the queries.

Combining the encoder and decoder modules form the full autoencoder architecture. It enables the embedding of structures in a latent space that can be subsequently decoded to recover the structure. The autoencoder architecture offers versatility for different applications. One such application is the embedding and sampling of molecular conformations. By en-

coding conformations into and sampling conformations from a latent space, the model can efficiently generate new, plausible states.

As mention previously, another promising application lies in first-principles protein folding. A straightforward way to achieve this would be to feed the model atomic elements, topology, and random coordinates. The architecture could then iteratively refine these initial conditions to approach the target structure. An enhancement to this would be to incorporate techniques inspired from the stable diffusion approach, allowing for diffusion in the latent space to more efficiently and accurately recover the target structure.

In drug docking scenarios, the complete folding of the protein might not be necessary if we already have the undocked structure of the protein. In such cases, the model could take the undocked protein and ligand structures as inputs and predict the docked ligand with the protein, allowing the conformation to adapt to the ligand if necessary. In a broader perspective, first-principles structure prediction could model the interactions between any type of molecules such as protein-lipids for perpheral membrane proteins or to study the implications of non-natural amino acids such as post-translational modifications.

In conclusion, the significance of this work extends beyond the developed examples. Given that the model is agnostic to the type of biomolecules and the composition of the system, it offers the potential to predict and model interactions between any types of molecules at the structural level. This universality could have profound implications across various scientific domains.

A Generalizable transport signal processing and deep learning method for the classification of single events

Krapp L.F., Cao C., Dal Peraro M.
Generalizable transport signal processing and deep learning method for the classification of single events

Disclaimer

The following section is adapted from the unpublished work.

Contributions

L.F.K., C.C. and M.D.P. designed the research project. L.F.K. implemented the transport processing pipeline. C.C. conducted the experiments. L.F.K., C.C. and M.D.P. analyzed the results.

A.1 Introduction

Nanopores are pores on the nanometer scale allowing the transport of small molecules through thin surface material. There are two families of nanopores: biological or organic nanopores and solid-state or inorganic nanopores. Biological nanopores are usually toxins produced by bacteria in order to create a hole in the membrane of a cell, leading to an unregulated flow of ions and small molecules and the death of the cell. Solid-state nanopores are man-made pores in materials with a thickness of several nanometers.

It has been observed that the translocation of small molecules through a nanopore generates current blockades that can be used as a signature to identify single molecules [120]. Moreover, nanopores can also be applied to polymers such as DNA, enabling nanopore-based sequencing [121]. Using nanopores as a sensing method opens the door for various applications in single-molecule characterization, as well as other ways to store digital information [122]. Bi-

ological nanopores have multiple significant advantages over their solid-state counterparts. First, pore-forming toxins are self-assembling and will automatically insert themselves into a lipid bilayer. For most nanopore, the self-assembly consistently creates pores with the same structure and a hole of the same size. Second, the structure of the nanopore can be finely tuned with mutations to fit a specific application [123][124].

Analyzing transport signal is a complex task due to the low signal-to-noise ratio (SNR). This low SNR makes it challenging to accurately detect and analyze events in the data, as the actual signals are often obscured by noise. Various methods have been developed to process these signals and to detect and analyze events more effectively [125, 126]. Traditional approaches might include filtering techniques and statistical methods for step detection. However, these methods often fall short when dealing with the complexities and ambiguities presented by the low SNR, particularly for short and fast events.

Deep learning techniques have shown promise in tackling this problem. Specifically, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been applied to the classification of translocation events in transport signals[127]. CNNs are particularly useful for spatial pattern recognition and can extract hierarchical features from the data, while RNNs can capture temporal dependencies, which are often important in the context of transport events.

In this work, we introduce a data analysis pipeline designed specifically for processing and analyzing large volumes of transport signals. The signals are measured with the experimental setup explained in Figure A.1. The pipeline is constructed to manage the complexities and challenges inherent in signal data. We have applied this pipeline to two distinct applications: the classification of tailor-made polymers for data storage[122] and the identification of post-translational modifications (PTM) on disordered peptides.

The tailor-made polymers were designed to encode ’0’ and ’1’ using monomers with very different sterical volumes, as described in our previous work[122]. They are terminated by two adenosine and a padding ’0’: ’AA0...0AA’. The ’0’ and ’1’ in the middle of the chain defines the information stored. For instace, we have 1-bit (’AA0x0AA’), 2-bits (’AA0xx0AA’), 3-bits (’AA0xxx0AA’) and 4-bits (’AA0xxxx0AA’).

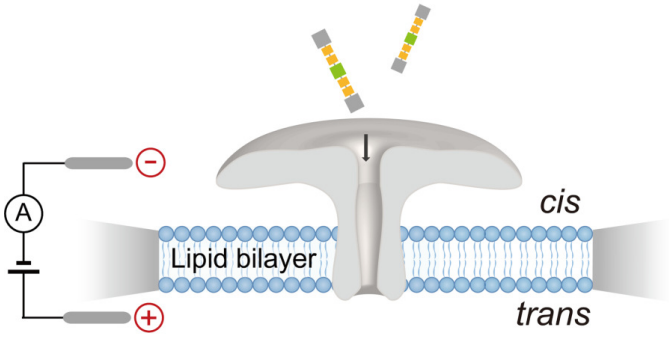

Figure A.1: A single nanopore, in this case aerolysin, is inserted in a lipid bilayer. A voltage is applied to the conductive buffer between the cis and trans chambers. The charged linear analyte is capture by the pore and translocated through the pore resulting in a blocked current: an event. The chemical properties of components of the polymer can displace a varying amount of current, enabling the identification of different type of analytes with a unique event fingerprint.

A.2 Methods

We describe here the transport signal processing (TSP) pipeline to detect and process translocation events for the classification of polymers. The first stage semi-automatically processes raw measurements by segmenting the signal, detecting the open pore current, detecting the events, and processing the events. The second stage is used to manually select segments and events by filtering out outliers. The processed selected events can then be used for further analysis or deep learning applications.

A.2.1 Signal processing

We assume that all the measurements are done at a constant voltage. First, the signal is divided into multiple segments in order to remove unwanted measures, see Figure A.2a,b. The segmentation is done using voltage discontinuities that can indicate that the pore is blocked. It is also segmented by scanning on an 8s window for large current variations from the mean. For each segment, the open pore current is automatically detected and the current distribution is fitted with a Gaussian function to extract the average and standard deviation (σ) of the open pore current.

Events processing

For each segment, the events are detected and processed, see Figure A.2c. Events are detected using a detection threshold at 3σ from the average open pore current. We then process each event by extracting the core of the event using an adaptive cut-off within the event to remove the tails of the event. The local extrema are extracted. The relative current is defined as the

current divided by the average open pore current. It is used to normalize small variations in open-pore current between measurements.

Segments and events filtering

Segments of the signal with no current or multiple pores are filtered using an upper and lower threshold on the average open pore current. Moreover, segments with a high open pore current noise level ($\sigma > 4.2$ pA) that can indicate an issue with the pore or the experimental setup are filtered. Events with a dwell time between 0.4 and 30ms are kept while discarding short spike events and rare long events. Moreover, outliers are discarded by keeping events with an average relative current between 10 and 60%. On average, ~90% of non-spike events detected are kept. Spike events are defined as purely convex events without any distinct local maxima within the event.

Data analysis

In our data analysis, we employed several statistical methods to assess the signal of events. Specifically, we calculated the mean, standard deviation, kurtosis, and skewness of the current within the event. These statistics provide a comprehensive understanding of the distribution characteristics of the signal, helping us identify patterns or anomalies. In addition to statistical analysis, we also used k-means clustering on interpolated events. This technique allows us to group similar events together, making it easier to observe trends or differences within the dataset. The interpolation before clustering ensured that the events were comparable on a similar scale.

We implemented a consistency check using the Kullback-Leibler (KL) divergence between measurements. This helped us quantify the similarity or divergence between different sets of statistics, thereby allowing us to validate the reliability of our data analysis procedures. Lastly, we developed an algorithm for levels analysis, employing a multi-Gaussian fit on the local extrema distribution. This advanced method helps us identify multiple states or levels within the signal, which is critical for understanding more complex event structures.

Overall, our data analysis involved a mix of statistical metrics, clustering, consistency checks, and advanced fitting techniques to provide a comprehensive evaluation of the event signals.

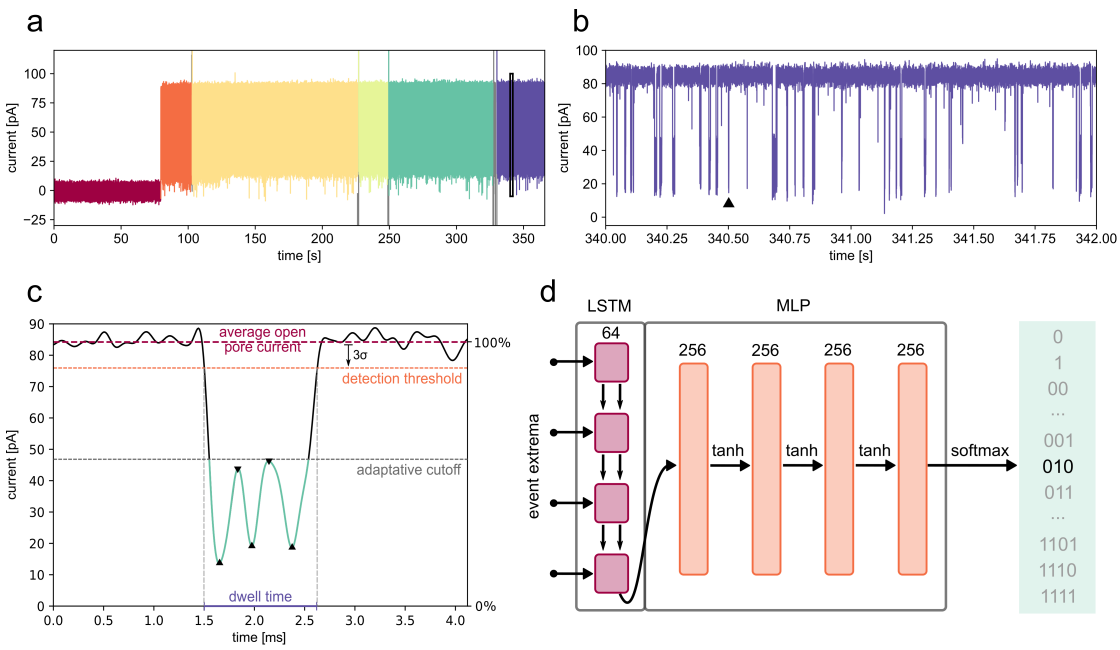

Figure A.2: (a) Example of raw signal segmentation using voltage or current discontinuities from a measurement of AA00100AA at 25°C and 100 mV. (b) Part of a segment showing the open pore current signal and events. (c) Example of event detection and processing using an adaptive cutoff and local extrema extraction. (d) Deep recurrent neural network for events classification using the local extrema as input features composed of a long short-term memory (LSTM) layer with state size 64 and a multilayer perceptron (MLP) with 4 hidden layers of size 256. The output of the model illustrates the classification task for the 1 to 4-bits polymers.

Deep learning

The local extrema position in relative current and time within the event are used as input features. The events can have different lengths, so we used a long short-term memory (LSTM) recurrent neural network, which is well suited for the variable sizes of inputs. The deep learning model is composed of the blocks of operations, see Figure A.2d. First, a single pass LSTM layer with state size 64 reads and encodes the variable length input features. Second, a multilayer perceptron (MLP) with 4 hidden layers of size 256 decodes the output from the LSTM layer and identifies the event. The model is trained to identify single events using the cross-entropy loss as a criterion. The number of classes depends on the classification task. The architecture can be adapted to different classification problems based on the data quality and quantity. An optional, filtering RNN can be trained in parallel. It is a scaled-down version of the classification RNN trained to assess the quality of the predictions. It is used in conjunction with the classification confidence to filter out events with low confidence predictions.

Datasets

We applied our method to two different classes of polymers: tailor-made polymers and peptides with PTMs. The task of polymers identification was divided into five tasks: 1 bit (71'000 events, 2 classes), 2 bits (440'000 events, 4 classes), 3 bits (550'00 events, 8 classes), 4 bits (2'850'000 events, 16 classes) and 1 to 4 bits (3'910'000 events, 30 classes) polymers. The task of PTM identification was divided into 3 tasks: Localized PTM (WT, nY125, pY125) (64'000 events), single PTM (WT, nY125, nY136, pS129, pY125) (100'000 events), multi-PTMs (WT, nY125, nY125nY133nY136, nY136, pS129, pY125, pY125pS129) (120'000 events).

To create a proper train, evaluation, and test dataset, we grouped events based on their segment of origin. This approach ensures that events in each dataset come from different blocks of measurements. By doing so, we mitigate the risk of overfitting and introduce a level of generalization. The advantage of this segmentation strategy is that it makes the events unbiased. This is crucial for the practical application of our model, as it ensures that the trained model can be effectively applied to new, measured segments in real-world scenarios.

A.3 Results

We defined and developed a method for the classification of tailor-made polymers and then applied the same protocol to the identification of PTMs from peptide translocation events.

A.3.1 Events processing and features selection

We observed that the trained models were not performing as well when evaluating them on events measured from another instance of the pore. Mainly, the experimental conditions are not exactly the same, and the open pore current can be different between measurement instances. We hypothesized that the first current drop when the polymer enters the pore and the current increase when the polymer is leaving is dependent on the instance of the pore but not relevant to the identification of the polymers. Using our adaptive cutoff, we obtain an accuracy of 93±1% on events core instead of 81±2% for raw events on the validation dataset for the same training accuracy of ~95%. Extracting the core of the event forces the model to recognize patterns within the event instead of using the current changes when the event enters and leaves the pore to recognize the pore instance. Therefore, we show that our trained models are generally applicable to subsequent instances of the pore.

Using normalized events (z-score), we obtain an accuracy of 72±1% against 93±1% for raw events core on the validation dataset. Further analysis revealed that the rescaling of the amplitude of the signal (dividing by the standard deviation) is the major contributor to the decrease in performance with an accuracy of ~75% against ~85% for current centered events only on the validation dataset. Therefore, rescaling the events distort some information important for the identification of the polymers.

As a baseline, we trained and evaluated a simple fully connected neural network (NN) on handpicked features. The selected features are the dwell time, the relative current average, standard deviation, skewness and kurtosis, the number of local extrema, and the peak to peak relative current. For 1-bit, 2-bits, and 3-bits polymers, we obtain an accuracy of 92±1%, 79±2%, and 76±1% for the RNN model respectively and 89±1%, 91±1%, and 71±2% for the NN model respectively. We observe that the hand-picked features do not scale well with the increasing number of classes.

A.3.2 Evaluation of the deep learning model

In evaluating our deep learning model, we found that it performs robustly in both polymer classification and PTM identification tasks, see Figure A.3a,d. The model achieved a consistent ROC AUC score of approximately 0.9 across all classification tasks, indicating a high level of performance. It is noteworthy that the performance of the model scales well with the number of classes. While identifying events becomes theoretically more challenging as the number of different polymers increases, our model maintains a strong ROC AUC score. For instance, in a 1-bit case with only two combinations, the model performs just as well as in a 4-bit case with 16 combinations.

Another key finding is that the accuracy of event classification is directly correlated with the dwell time of the event, as shown in Figure A.3b,e. Short events, which contain less information and more noise, are harder to classify accurately. Conversely, the longer the event dwell time, the more information is available for accurate identification.

We also introduced a fine-tuning step based on confidence levels to ignore events of poor quality (Figure A.3c,f). With a selection rate of 1 in 2 detected events, the model already shows high accuracy across all classifications. However, when we tightened the selection rate to 1 in 4 events, the predictive accuracy of our model was maximized.

In summary, our deep learning model not only performs well in classifying polymers and identifying PTMs but also shows scalability with the number of classes. We show that there is a direct relationship between event dwell time and classification accuracy. Furthermore, the performance of the model can be fine-tuned by using confidence levels to filter out low-quality events.

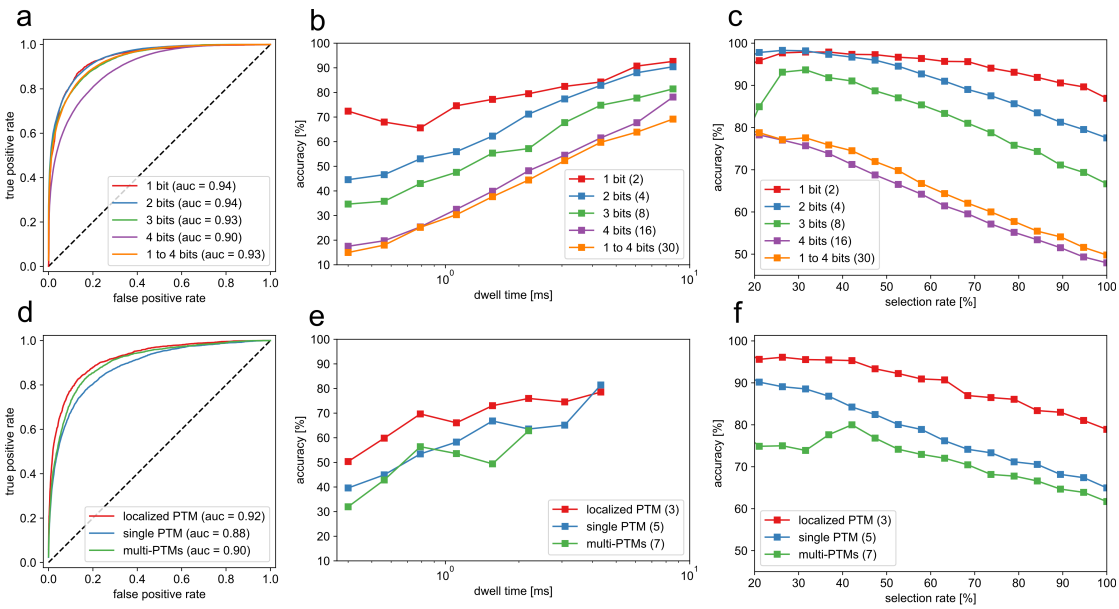

Figure A.3: Assessment of our deep learning model on polymers (a,b,c) and post-translational modifications (PTMs) (d,e,f) classification tasks. (a,d) Receiver operating characteristic (ROC) curve and area under the curve (AUC) scores. (b,e) Model accuracy as a function of the dwell time of the events. (c,f) Model accuracy for different confidence thresholds is expressed as the selection rate of the events.

A.4 Discussion

These applications showcase the versatility and broad applicability of our methodology. Our results demonstrate that the pipeline performs well in single-event analysis across different types of analytes. This underscores the utility of our approach, highlighting its potential for diverse applications in the field of transport signal analysis.

Our method can be extended to general sequence-to-sequence prediction using the connectionist temporal classification loss [128]. It would allow us to generalize our method to the prediction of the sequences of polymers with arbitrary lengths. Moreover, events interpretability using approaches such as integrated gradient [129] can be used to analyze and understand translocation events. Transformers [26, 27] are the current leading deep learning approach for natural language processing. Unlike recurrent neural networks such as LSTM or GRU, the computational complexity of transformers scales with the input length. Due to the length of the events, recurrent neural networks are more competitive than transformers due to the scaling computational complexity. Improving the classification and reading of analytes could be achieved using linear time and space complexity with linformer [130] or performers [131]. As an added benefit, the integrated attention mechanisms in transformers can be used to interpret events. This information can be used to improve pore designs and experimental conditions.

Bibliography

[1] D. L. D. L. Nelson, 1942, *Lehninger principles of biochemistry*. Fourth edition. New York : W.H. Freeman, 2005., 2005.

[2] O. Carugo and K. Djinović-Carugo, "Structural biology: A golden era," *PLOS Biology*, vol. 21, p. e3002187, June 2023.

[3] W. Kühlbrandt, "The Resolution Revolution," *Science*, vol. 343, pp. 1443–1444, Mar. 2014.

[4] C. B. Anfinsen, "Principles that Govern the Folding of Protein Chains," *Science*, vol. 181, pp. 223–230, July 1973.

[5] V. Muñoz, ed., *Protein Folding: Methods and Protocols*, vol. 2376 of *Methods in Molecular Biology*. New York, NY: Springer US, 2022.

[6] J. Maynard Smith, "Natural Selection and the Concept of a Protein Space," *Nature*, vol. 225, pp. 563–564, Feb. 1970.

[7] J. Koehler Leman, P. Szczerbiak, P. D. Renfrew, V. Gligorijevic, D. Berenberg, T. Vatanen, B. C. Taylor, C. Chandler, S. Janssen, A. Pataki, N. Carriero, I. Fisk, R. J. Xavier, R. Knight, R. Bonneau, and T. Kosciolek, "Sequence-structure-function relationships in the microbial protein universe," *Nature Communications*, vol. 14, p. 2351, Apr. 2023.

[8] I. Roterman-Konieczna, ed., *Identification of Ligand Binding Site and Protein-Protein Interaction Area*, vol. 8 of *Focus on Structural Biology*. Dordrecht: Springer Netherlands, 2013.

[9] H. Lu, Q. Zhou, J. He, Z. Jiang, C. Peng, R. Tong, and J. Shi, "Recent advances in the development of protein–protein interactions modulators: mechanisms and clinical trials," *Signal Transduction and Targeted Therapy*, vol. 5, p. 213, Sept. 2020.

[10] S. A. Hollingsworth and R. O. Dror, "Molecular Dynamics Simulation for All," *Neuron*, vol. 99, pp. 1129–1143, Sept. 2018.

[11] D. S. Ziemianowicz and J. Kosinski, "New opportunities in integrative structural modeling," *Current Opinion in Structural Biology*, vol. 77, p. 102488, Dec. 2022.

[12] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, pp. 583–589, Aug. 2021.

[13] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker, "Robust deep learning–based protein sequence design using ProteinMPNN," *Science*, vol. 378, pp. 49–56, Oct. 2022.

[14] P. Gainza, F. Sverrisson, F. Monti, E. Rodolà, D. Boscaini, M. M. Bronstein, and B. E. Correia, "Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning," *Nature Methods*, vol. 17, pp. 184–192, Feb. 2020.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning*. Information Science and Statistics, New York, NY: Springer New York, softcover reprint of the original 1st edition 2006 (corrected at 8th printing 2009) ed., 2016.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Dec. 2019. arXiv:1912.01703 [cs, stat].

[18] TensorFlow Developers, "TensorFlow," Aug. 2023.

[19] S. Mittal and S. Vaishay, "A survey of techniques for optimizing deep learning on GPUs," *Journal of Systems Architecture*, vol. 99, p. 101635, Oct. 2019.

[20] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 2017. arXiv:1412.6980 [cs].

[21] M. Minsky and S. A. Papert, *Perceptrons: an introduction to computational geometry*. Cambridge/Mass.: The MIT Press, 2. print. with corr ed., 1972.

[22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, Nov. 1998.

[23] P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. Van Hoesel, H. Schopmans, T. Sommer, and P. Friederich, "Graph neural networks for materials science and chemistry," *Communications Materials*, vol. 3, p. 93, Nov. 2022.

[24] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997.

[25] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 2014. arXiv:1412.3555 [cs].

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," Tech. Rep. arXiv:1706.03762, arXiv, Dec. 2017. arXiv: 1706.03762 [cs].

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Tech. Rep. arXiv:1810.04805, arXiv, May 2019. arXiv: 1810.04805 [cs].

[28] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, "Emergent Abilities of Large Language Models," Oct. 2022. arXiv:2206.07682 [cs].

[29] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank," *Nucleic Acids Research*, vol. 28, pp. 235–242, Jan. 2000.

[30] The UniProt Consortium, "UniProt: the universal protein knowledgebase in 2021," *Nucleic Acids Research*, vol. 49, pp. D480–D489, Jan. 2021.

[31] E. Laine, S. Eismann, A. Elofsson, and S. Grudinin, "Protein sequence-to-structure learning: Is this the end(-to-end revolution)?," *Proteins: Structure, Function, and Bioinformatics*, vol. 89, no. 12, pp. 1770–1786, 2021.

[32] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges," May 2021. arXiv:2104.13478 [cs, stat].

[33] G. Derevyanko, S. Grudinin, Y. Bengio, and G. Lamoureux, "Deep convolutional networks for quality assessment of protein folds," *Bioinformatics*, vol. 34, pp. 4046–4053, Dec. 2018.

[34] G. Derevyanko and G. Lamoureux, "Protein-protein docking using learned three-dimensional representations," tech. rep., bioRxiv, Aug. 2019.

[35] J. S. Smith, O. Isayev, and A. E. Roitberg, "ANI-1: An extensible neural network potential with DFT accuracy at force field computational cost," *Chemical Science*, vol. 8, no. 4, pp. 3192–3203, 2017.

[36] K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller, "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions," *arXiv:1706.08566 [physics, stat]*, Dec. 2017.

[37] F. Baldassarre, D. Menéndez Hurtado, A. Elofsson, and H. Azizpour, "GraphQA: protein model quality assessment using graph convolutional networks," *Bioinformatics*, vol. 37, pp. 360–366, Feb. 2021.

[38] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, "Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds," Tech. Rep. arXiv:1802.08219, arXiv, May 2018. arXiv: 1802.08219 [cs].

[39] B. Anderson, T.-S. Hy, and R. Kondor, "Cormorant: Covariant Molecular Neural Networks," Tech. Rep. arXiv:1906.04015, arXiv, Nov. 2019. arXiv: 1906.04015 [physics, stat].

[40] S. Eismann, R. J. Townshend, N. Thomas, M. Jagota, B. Jing, and R. O. Dror, "Hierarchical, rotation-equivariant neural networks to select structural models of protein complexes," *Proteins: Structure, Function, and Bioinformatics*, vol. 89, no. 5, pp. 493–501, 2021.

[41] B. Jing, S. Eismann, P. Suriana, R. J. L. Townshend, and R. Dror, "Learning from Protein Structure with Geometric Vector Perceptrons," Tech. Rep. arXiv:2009.01411, arXiv, May 2021. arXiv: 2009.01411 [cs, q-bio, stat].

[42] B. Jing, S. Eismann, P. N. Soni, and R. O. Dror, "Equivariant Graph Neural Networks for 3D Macromolecular Structure," Tech. Rep. arXiv:2106.03843, arXiv, July 2021. arXiv: 2106.03843 [cs, q-bio].

[43] V. G. Satorras, E. Hoogeboom, and M. Welling, "E(n) Equivariant Graph Neural Networks," *arXiv:2102.09844 [cs, stat]*, Feb. 2021.

[44] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," Tech. Rep. arXiv:2005.14165, arXiv, July 2020. arXiv: 2005.14165 [cs].

[45] F. B. Fuchs, D. E. Worrall, V. Fischer, and M. Welling, "SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks," Tech. Rep. arXiv:2006.10503, arXiv, Nov. 2020. arXiv: 2006.10503 [cs, stat].

[46] M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, C. Millán, H. Park, C. Adams, C. R. Glassman, A. DeGiovanni, J. H. Pereira, A. V. Rodrigues, A. A. van Dijk, A. C. Ebrecht, D. J. Opperman, T. Sagmeister, C. Buhlheller, T. Pavkov-Keller, M. K. Rathinaswamy, U. Dalwadi, C. K. Yip, J. E. Burke, K. C. Garcia, N. V. Grishin, P. D. Adams, R. J. Read, and D. Baker, "Accurate prediction of protein structures and interactions using a three-track neural network," *Science*, vol. 373, pp. 871–876, Aug. 2021.

[47] R. Chowdhury, N. Bouatta, S. Biswas, C. Rochereau, G. M. Church, P. K. Sorger, and M. AlQuraishi, "Single-sequence protein structure prediction using language models from deep learning," tech. rep., bioRxiv, Aug. 2021.

[48] R. Evans, M. O'Neill, A. Pritzel, N. Antropova, A. Senior, T. Green, A. Žídek, R. Bates, S. Blackwell, J. Yim, O. Ronneberger, S. Bodenstein, M. Zielinski, A. Bridgland, A. Potapenko, A. Cowie, K. Tunyasuvunakool, R. Jain, E. Clancy, P. Kohli, J. Jumper, and D. Hassabis, "Protein complex prediction with AlphaFold-Multimer," preprint, Bioinformatics, Oct. 2021.

[49] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives, "Evolutionary-scale prediction of atomic-level protein structure with a language model," *Science*, vol. 379, pp. 1123–1130, Mar. 2023.

[50] A. Porollo and J. Meller, "Prediction-based fingerprints of protein–protein interactions," *Proteins: Structure, Function, and Bioinformatics*, vol. 66, no. 3, pp. 630–645, 2007.

[51] Y. Murakami and K. Mizuguchi, "Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites," *Bioinformatics*, vol. 26, pp. 1841–1848, Aug. 2010.

[52] T. C. Northey, A. Barešić, and A. C. R. Martin, "IntPred: a structure-based predictor of protein–protein interaction sites," *Bioinformatics*, vol. 34, pp. 223–229, Jan. 2018.

[53] F. Sverrisson, J. Feydy, B. E. Correia, and M. M. Bronstein, "Fast end-to-end learning on protein surfaces," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15267–15276, June 2021. ISSN: 2575-7075.

[54] N. Ferruz, M. Heinzinger, M. Akdel, A. Goncearenco, L. Naef, and C. Dallago, "From sequence to function through structure: Deep learning for protein design," *Computational and Structural Biotechnology Journal*, vol. 21, pp. 238–250, 2023.

[55] J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, N. Hanikel, S. J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh, I. Sappington, S. V. Torres, A. Lauko, V. De Bortoli, E. Mathieu, S. Ovchinnikov, R. Barzilay, T. S. Jaakkola, F. DiMaio, M. Baek, and D. Baker, "De novo design of protein structure and function with RFdiffusion," *Nature*, pp. 1–3, July 2023. Publisher: Nature Publishing Group.

[56] C. Hsu, R. Verkuil, J. Liu, Z. Lin, B. Hie, T. Sercu, A. Lerer, and A. Rives, "Learning inverse folding from millions of predicted structures," Sept. 2022.

[57] P. Gainza, S. Wehrle, A. Van Hall-Beauvais, A. Marchand, A. Scheck, Z. Harteveld, S. Buckley, D. Ni, S. Tan, F. Sverrisson, C. Goverde, P. Turelli, C. Raclot, A. Teslenko,

M. Pacesa, S. Rosset, S. Georgeon, J. Marsden, A. Petruzzella, K. Liu, Z. Xu, Y. Chai, P. Han, G. F. Gao, E. Oricchio, B. Fierz, D. Trono, H. Stahlberg, M. Bronstein, and B. E. Correia, "De novo design of protein interactions with learned surface fingerprints," *Nature*, vol. 617, pp. 176–184, May 2023.

[58] S. R. Hall, F. H. Allen, and I. D. Brown, "The crystallographic information file (CIF): a new standard archive file for crystallography," *Acta Crystallographica Section A Foundations of Crystallography*, vol. 47, pp. 655–685, Nov. 1991.

[59] P. E. Bourne, H. M. Berman, B. McMahon, K. D. Watenpaugh, J. D. Westbrook, and P. M. Fitzgerald, "[30] Macromolecular crystallographic information file," in *Methods in Enzymology*, vol. 277, pp. 571–590, Elsevier, 1997.

[60] M. Wojdyr, "GEMMI: A library for structural biology," *Journal of Open Source Software*, vol. 7, p. 4200, May 2022.

[61] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020. Number: 7825 Publisher: Nature Publishing Group.

[62] N. The HDF Group and Q. Koziol, "HDF5-Version 1.12.0," 2020. Language: en.

[63] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 89, pp. 10915–10919, Nov. 1992.

[64] T. Vreven, I. H. Moal, A. Vangone, B. G. Pierce, P. L. Kastritis, M. Torchala, R. Chaleil, B. Jiménez-García, P. A. Bates, J. Fernandez-Recio, A. M. J. J. Bonvin, and Z. Weng, "Updates to the Integrated Protein–Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2," *Journal of Molecular Biology*, vol. 427, pp. 3031–3041, Sept. 2015.

[65] J. Tubiana, D. Schneidman-Duhovny, and H. J. Wolfson, "ScanNet: an interpretable geometric deep learning model for structure-based protein binding site prediction," *Nature Methods*, vol. 19, pp. 730–739, June 2022.

[66] M. Mirdita, K. Schütze, Y. Moriwaki, L. Heo, S. Ovchinnikov, and M. Steinegger, "ColabFold: making protein folding accessible to all," *Nature Methods*, vol. 19, pp. 679–682, June 2022.

[67] M. Steinegger and J. Söding, "MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets," *Nature Biotechnology*, vol. 35, pp. 1026–1028, Nov. 2017. Number: 11 Publisher: Nature Publishing Group.

[68] L. A. Abriata and M. Dal Peraro, "Assessment of transferable forcefields for protein simulations attests improved description of disordered states and secondary structure propensities, and hints at multi-protein systems as the next challenge for optimization," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 2626–2636, Jan. 2021.

[69] J. Huang, S. Rauscher, G. Nawrocki, T. Ran, M. Feig, B. L. de Groot, H. Grubmüller, and A. D. MacKerell, "CHARMM36m: an improved force field for folded and intrinsically disordered proteins," *Nature Methods*, vol. 14, pp. 71–73, Jan. 2017.

[70] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. C. Berendsen, "GROMACS: Fast, flexible, and free," *Journal of Computational Chemistry*, vol. 26, no. 16, pp. 1701–1718, 2005.

[71] S. Träger, G. Tamò, D. Aydin, G. Fonti, M. Audagnotto, and M. Dal Peraro, "CLoNe: automated clustering based on local density neighborhoods for application to biomolecular structural ensembles," *Bioinformatics*, vol. 37, pp. 921–928, Aug. 2020.

[72] M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Žídek, T. Green, K. Tunyasuvunakool, S. Petersen, J. Jumper, E. Clancy, R. Green, A. Vora, M. Lutfi, M. Figurnov, A. Cowie, N. Hobbs, P. Kohli, G. Kleywegt, E. Birney, D. Hassabis, and S. Velankar, "AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models," *Nucleic Acids Research*, vol. 50, pp. D439–D444, Jan. 2022.

[73] A. Shrake and J. A. Rupley, "Environment and exposure to solvent of protein atoms. Lysozyme and insulin," *Journal of Molecular Biology*, vol. 79, pp. 351–371, Sept. 1973.

[74] R. T. McGibbon, K. A. Beauchamp, M. P. Harrigan, C. Klein, J. M. Swails, C. X. Hernández, C. R. Schwantes, L.-P. Wang, T. J. Lane, and V. S. Pande, "MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories," *Biophysical Journal*, vol. 109, pp. 1528–1532, Oct. 2015.

[75] S. T. Sherry, M.-H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski, and K. Sirotkin, "dbSNP: the NCBI database of genetic variation," *Nucleic Acids Research*, vol. 29, pp. 308–311, Jan. 2001.

[76] I. R. Humphreys, J. Pei, M. Baek, A. Krishnakumar, I. Anishchenko, S. Ovchinnikov, J. Zhang, T. J. Ness, S. Banjade, S. R. Bagde, V. G. Stancheva, X.-H. Li, K. Liu, Z. Zheng, D. J. Barrero, U. Roy, J. Kuper, I. S. Fernández, B. Szakal, D. Branzei, J. Rizo, C. Kisker, E. C. Greene, S. Biggins, S. Keeney, E. A. Miller, J. C. Fromme, T. L. Hendrickson, Q. Cong, and D. Baker, "Computed structures of core eukaryotic protein complexes," *Science*, vol. 374, p. eabm4805, Nov. 2021.

[77] L. F. Krapp, L. A. Abriata, F. Cortés Rodriguez, and M. Dal Peraro, "PeSTo: parameter-free geometric deep learning for accurate prediction of protein binding interfaces," *Nature Communications*, vol. 14, p. 2175, Apr. 2023.

[78] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. L. Madden, "BLAST+: architecture and applications," *BMC Bioinformatics*, vol. 10, p. 421, Dec. 2009.

[79] Y. Zhang and J. Skolnick, "Scoring function for automated assessment of protein structure template quality," *Proteins: Structure, Function, and Bioinformatics*, vol. 57, no. 4, pp. 702–710, 2004.

[80] V. Mariani, M. Biasini, A. Barbato, and T. Schwede, "lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests," *Bioinformatics*, vol. 29, pp. 2722–2728, Nov. 2013.

[81] Z. Deng, W. Huang, E. Bakkalbasi, N. G. Brown, C. J. Adamski, K. Rice, D. Muzny, R. A. Gibbs, and T. Palzkill, "Deep sequencing of systematic combinatorial libraries reveals β-lactamase sequence constraints at high resolution," *Journal of Molecular Biology*, vol. 424, pp. 150–167, Dec. 2012.

[82] O. Trott and A. J. Olson, "AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading," *Journal of computational chemistry*, vol. 31, pp. 455–461, Jan. 2010.

[83] C. V. Robinson, A. Sali, and W. Baumeister, "The molecular sociology of the cell," *Nature*, vol. 450, pp. 973–982, Dec. 2007.

[84] M. Vidal, M. E. Cusick, and A.-L. Barabási, "Interactome Networks and Human Disease," *Cell*, vol. 144, pp. 986–998, Mar. 2011.

[85] T. Rolland, M. Taşan, B. Charloteaux, S. J. Pevzner, Q. Zhong, N. Sahni, S. Yi, I. Lemmens, C. Fontanillo, R. Mosca, A. Kamburov, S. D. Ghiassian, X. Yang, L. Ghamsari, D. Balcha, B. E. Begg, P. Braun, M. Brehme, M. P. Broly, A.-R. Carvunis, D. Convery-Zupan, R. Corominas, J. Coulombe-Huntington, E. Dann, M. Dreze, A. Dricot, C. Fan, E. Franzosa, F. Gebreab, B. J. Gutierrez, M. F. Hardy, M. Jin, S. Kang, R. Kiros, G. N. Lin, K. Luck, A. MacWilliams, J. Menche, R. R. Murray, A. Palagi, M. M. Poulin, X. Rambout, J. Rasla, P. Reichert, V. Romero, E. Ruyssinck, J. M. Sahalie, A. Scholz, A. A. Shah, A. Sharma, Y. Shen, K. Spirohn, S. Tam, A. O. Tejeda, S. A. Trigg, J.-C. Twizere, K. Vega, J. Walsh, M. E. Cusick, Y. Xia, A.-L. Barabási, L. M. Iakoucheva, P. Aloy, J. De Las Rivas, J. Tavernier, M. A. Calderwood, D. E. Hill, T. Hao, F. P. Roth, and M. Vidal, "A Proteome-Scale Map of the Human Interactome Network," *Cell*, vol. 159, pp. 1212–1226, Nov. 2014.

[86] R. Esmaielbeiki, K. Krawczyk, B. Knapp, J.-C. Nebel, and C. M. Deane, "Progress and challenges in predicting protein interfaces," *Briefings in Bioinformatics*, vol. 17, pp. 117–131, Jan. 2016.

[87] Q. C. Zhang, D. Petrey, L. Deng, L. Qiang, Y. Shi, C. A. Thu, B. Bisikirska, C. Lefebvre, D. Accili, T. Hunter, T. Maniatis, A. Califano, and B. Honig, "Structure-based prediction

of protein–protein interactions on a genome-wide scale," *Nature*, vol. 490, pp. 556–560, Oct. 2012.

[88] D. E. Scott, A. R. Bayly, C. Abell, and J. Skidmore, "Small molecules, big targets: drug discovery faces the protein–protein interaction challenge," *Nature Reviews Drug Discovery*, vol. 15, pp. 533–550, Aug. 2016.

[89] A. G. Green, H. Elhabashy, K. P. Brock, R. Maddamsetti, O. Kohlbacher, and D. S. Marks, "Large-scale discovery of protein interactions at residue resolution using co-evolution calculated from genomic sequences," *Nature Communications*, vol. 12, p. 1396, Mar. 2021.

[90] G. Croce, T. Gueudré, M. V. R. Cuevas, V. Keidel, M. Figliuzzi, H. Szurmant, and M. Weigt, "A multi-scale coevolutionary approach to predict interactions between protein domains," *PLOS Computational Biology*, vol. 15, p. e1006891, Oct. 2019.

[91] S. Ovchinnikov, H. Kamisetty, and D. Baker, "Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information," *eLife*, vol. 3, p. e02030, May 2014.

[92] Q. Cong, I. Anishchenko, S. Ovchinnikov, and D. Baker, "Protein interaction networks revealed by proteome coevolution," *Science*, vol. 365, pp. 185–189, July 2019.

[93] B. Dai and C. Bailey-Kellogg, "Protein interaction interface region prediction by geometric deep learning," *Bioinformatics*, vol. 37, pp. 2580–2588, Sept. 2021.

[94] B. Ozden, A. Kryshtafovych, and E. Karaca, "Assessment of the CASP14 assembly predictions," *Proteins: Structure, Function, and Bioinformatics*, vol. 89, no. 12, pp. 1787–1799, 2021.

[95] M. F. Lensink, G. Brysbaert, T. Mauri, N. Nadzirin, S. Velankar, R. A. G. Chaleil, T. Clarence, P. A. Bates, R. Kong, B. Liu, G. Yang, M. Liu, H. Shi, X. Lu, S. Chang, R. S. Roy, F. Quadir, J. Liu, J. Cheng, A. Antoniak, C. Czaplewski, A. Giełdoń, M. Kogut, A. G. Lipska, A. Liwo, E. A. Lubecka, M. Maszota-Zieleniak, A. K. Sieradzan, R. Ślusarz, P. A. Wesołowski, K. Zięba, C. A. Del Carpio Muñoz, E. Ichiishi, A. Harmalkar, J. J. Gray, A. M. J. J. Bonvin, F. Ambrosetti, R. Vargas Honorato, Z. Jandova, B. Jiménez-García, P. I. Koukos, S. Van Keulen, C. W. Van Noort, M. Réau, J. Roel-Touris, S. Kotelnikov, D. Padhorny, K. A. Porter, A. Alekseenko, M. Ignatov, I. Desta, R. Ashizawa, Z. Sun, U. Ghani, N. Hashemi, S. Vajda, D. Kozakov, M. Rosell, L. A. Rodríguez-Lumbreras, J. Fernandez-Recio, A. Karczynska, S. Grudinin, Y. Yan, H. Li, P. Lin, S.-Y. Huang, C. Christoffer, G. Terashi, J. Verburgt, D. Sarkar, T. Aderinwale, X. Wang, D. Kihara, T. Nakamura, Y. Hanazono, R. Gowthaman, J. D. Guest, R. Yin, G. Taherzadeh, B. G. Pierce, D. Barradas-Bautista, Z. Cao, L. Cavallo, R. Oliva, Y. Sun, S. Zhu, Y. Shen, T. Park, H. Woo, J. Yang, S. Kwon, J. Won, C. Seok, Y. Kiyota, S. Kobayashi, Y. Harada, M. Takeda-Shitaka, P. J. Kundrotas, A. Singh, I. A. Vakser, J. Dapkunas, K. Olechnovic, C. Venclovas, R. Duan, L. Qiu, X. Xu, S. Zhang, X. Zou, and S. J. Wodak, "Prediction of protein assemblies, the next frontier:

The CASP14- CAPRI experiment," *Proteins: Structure, Function, and Bioinformatics*, vol. 89, no. 12, pp. 1800–1823, 2021.

[96] F. Comitani and F. L. Gervasio, "Exploring Cryptic Pockets Formation in Targets of Pharmaceutical Interest with SWISH," *Journal of Chemical Theory and Computation*, vol. 14, pp. 3321–3331, June 2018.

[97] A. Kuzmanic, G. R. Bowman, J. Juarez-Jimenez, J. Michel, and F. L. Gervasio, "Investigating Cryptic Binding Sites by Molecular Dynamics Simulations," *Accounts of Chemical Research*, vol. 53, pp. 654–661, Mar. 2020.

[98] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. Cohen, "3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data," *arXiv:1807.02547 [cs, stat]*, Oct. 2018.

[99] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, "E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials," *Nature Communications*, vol. 13, p. 2453, May 2022.

[100] N. O. G. Jørgensen, "Carbohydrates," in *Encyclopedia of Inland Waters* (G. E. Likens, ed.), pp. 727–742, Oxford: Academic Press, Jan. 2009.

[101] J. P. Swiercz, T. Nanji, M. Gloyd, A. Guarné, and M. A. Elliot, "A novel nucleoid-associated protein specific to the actinobacteria," *Nucleic Acids Research*, vol. 41, pp. 4171–4184, Apr. 2013.

[102] N. T. Odermatt, M. Lelli, T. Herrmann, L. A. Abriata, A. Japaridze, H. Voilquin, R. Singh, J. Piton, L. Emsley, G. Dietler, and S. T. Cole, "Structural and DNA binding properties of mycobacterial integration host factor mIHF," *Journal of Structural Biology*, vol. 209, p. 107434, Mar. 2020.

[103] D. C. Berry, S. M. O'Byrne, A. C. Vreeland, W. S. Blaner, and N. Noy, "Cross Talk between Signaling and Vitamin A Transport by the Retinol-Binding Protein Receptor STRA6," *Molecular and Cellular Biology*, vol. 32, pp. 3164–3175, Aug. 2012.

[104] S. Ng, E. Lin, P. I. Kitov, K. F. Tjhung, O. O. Gerlits, L. Deng, B. Kasper, A. Sood, B. M. Paschal, P. Zhang, C.-C. Ling, J. S. Klassen, C. J. Noren, L. K. Mahal, R. J. Woods, L. Coates, and R. Derda, "Genetically Encoded Fragment-Based Discovery of Glycopeptide Ligands for Carbohydrate-Binding Proteins," *Journal of the American Chemical Society*, vol. 137, pp. 5248–5251, Apr. 2015.

[105] A. Malik, A. Firoz, V. Jha, and S. Ahmad, "PROCARB: A Database of Known and Modelled Carbohydrate-Binding Protein Structures with Sequence-Based Prediction Tools," *Advances in Bioinformatics*, vol. 2010, p. 436036, 2010.

[106] T. Serno, R. Geidobler, and G. Winter, "Protein stabilization by cyclodextrins in the liquid and dried state," *Advanced Drug Delivery Reviews*, vol. 63, pp. 1086–1106, Oct. 2011.

[107] A. Gu and N. J. Wheate, "Macrocycles as drug-enhancing excipients in pharmaceutical formulations," *Journal of Inclusion Phenomena and Macrocyclic Chemistry*, vol. 100, pp. 55–69, June 2021.

[108] R. Challa, A. Ahuja, J. Ali, and R. K. Khar, "Cyclodextrins in drug delivery: An updated review," *AAPS PharmSciTech*, vol. 6, pp. E329–E357, June 2005.

[109] B. I. M. Wicky, L. F. Milles, A. Courbet, R. J. Ragotte, J. Dauparas, E. Kinfu, S. Tipps, R. D. Kibler, M. Baek, F. DiMaio, X. Li, L. Carter, A. Kang, H. Nguyen, A. K. Bera, and D. Baker, "Hallucinating symmetric protein assemblies," *Science*, vol. 378, pp. 56–61, Oct. 2022.

[110] J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, N. Hanikel, S. J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh, I. Sappington, S. V. Torres, A. Lauko, V. De Bortoli, E. Mathieu, R. Barzilay, T. S. Jaakkola, F. DiMaio, M. Baek, and D. Baker, "Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models," preprint, Biochemistry, Dec. 2022.

[111] R. Verkuil, O. Kabeli, Y. Du, B. I. M. Wicky, L. F. Milles, J. Dauparas, D. Baker, S. Ovchinnikov, T. Sercu, and A. Rives, "Language models generalize beyond natural proteins," Dec. 2022.

[112] J. Ingraham, V. Garg, R. Barzilay, and T. Jaakkola, "Generative Models for Graph-Based Protein Design," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.

[113] D. Sgarbossa, U. Lupo, and A.-F. Bitbol, "Generative power of a protein language model trained on multiple sequence alignments," *eLife*, vol. 12, p. e79854, Feb. 2023.

[114] X. Zhou, G. Chen, J. Ye, E. Wang, J. Zhang, C. Mao, Z. Li, J. Hao, X. Huang, J. Tang, and P. Ann Heng, "Protein Sequence Design by Entropy-based Iterative Refinement," preprint, Bioinformatics, Feb. 2023.

[115] S. L. Lisanza, J. M. Gershon, S. Tipps, L. Arnoldt, S. Hendel, J. N. Sims, X. Li, and D. Baker, "Joint Generation of Protein Sequence and Structure with RoseTTAFold Sequence Space Diffusion," preprint, Biochemistry, May 2023.

[116] L. G. Doudeva, H. Huang, K.-C. Hsia, Z. Shi, C.-L. Li, Y. Shen, Y.-S. Cheng, and H. S. Yuan, "Crystal structural analysis and metal-dependent stability and activity studies of the ColE7 endonuclease domain in complex with DNA/Zn2+ or inhibitor/Ni2+," *Protein Science*, vol. 15, no. 2, pp. 269–280, 2006.

[117] L. A. Abriata, T. Palzkill, and M. Dal Peraro, "How structural and physicochemical determinants shape sequence constraints in a functional enzyme," *PloS One*, vol. 10, no. 2, p. e0118684, 2015.

[118] A. Mayorov, M. Dal Peraro, and L. A. Abriata, "Active Site-Induced Evolutionary Constraints Follow Fold Polarity Principles in Soluble Globular Enzymes," *Molecular Biology and Evolution*, vol. 36, pp. 1728–1733, Aug. 2019.

[119] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, and Z. Popović, "Predicting protein structures with a multiplayer online game," *Nature*, vol. 466, pp. 756–760, Aug. 2010.

[120] M. Akeson, D. Branton, J. J. Kasianowicz, E. Brandin, and D. W. Deamer, "Microsecond Time-Scale Discrimination Among Polycytidylic Acid, Polyadenylic Acid, and Polyuridylic Acid as Homopolymers or as Segments Within Single RNA Molecules," *Biophysical Journal*, vol. 77, pp. 3227–3233, Dec. 1999.

[121] J. Clarke, H.-C. Wu, L. Jayasinghe, A. Patel, S. Reid, and H. Bayley, "Continuous base identification for single-molecule nanopore DNA sequencing," *Nature Nanotechnology*, vol. 4, pp. 265–270, Apr. 2009.

[122] C. Cao, L. F. Krapp, A. Al Ouahabi, N. F. König, N. Cirauqui, A. Radenovic, J.-F. Lutz, and M. D. Peraro, "Aerolysin nanopores decode digital information stored in tailored macromolecular analytes," *Science Advances*, vol. 6, p. eabc2661, Dec. 2020.

[123] C. Cao, N. Cirauqui, M. J. Marcaida, E. Buglakova, A. Duperrex, A. Radenovic, and M. Dal Peraro, "Single-molecule sensing of peptides and nucleic acids by engineered aerolysin nanopores," *Nature Communications*, vol. 10, p. 4918, Oct. 2019.

[124] S. F. Mayer, C. Cao, and M. Dal Peraro, "Biological nanopores for single-molecule sensing," *iScience*, vol. 25, p. 104145, Apr. 2022.

[125] C. Plesa and C. Dekker, "Data analysis methods for solid-state nanopores," *Nanotechnology*, vol. 26, p. 084003, Feb. 2015. Publisher: IOP Publishing.

[126] F. L. R. Lucas, K. Willems, M. J. Tadema, K. M. Tych, G. Maglia, and C. Wloka, "Unbiased Data Analysis for the Parameterization of Fast Translocation Events through Nanopores," *ACS Omega*, vol. 7, pp. 26040–26046, Aug. 2022.

[127] K. Misiunas, N. Ermann, and U. F. Keyser, "QuipuNet: Convolutional Neural Network for Single-Molecule Nanopore Sensing," *Nano Letters*, vol. 18, pp. 4040–4045, June 2018. Publisher: American Chemical Society.

[128] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, ICML '06, (New York, NY, USA), pp. 369–376, Association for Computing Machinery, June 2006.

[129] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks," Tech. Rep. arXiv:1703.01365, arXiv, June 2017. arXiv: 1703.01365 [cs].

[130] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-Attention with Linear Complexity," Tech. Rep. arXiv:2006.04768, arXiv, June 2020. arXiv: 2006.04768 [cs, stat].

[131] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, and A. Weller, "Rethinking Attention with Performers," Tech. Rep. arXiv:2009.14794, arXiv, Mar. 2021. arXiv: 2009.14794 [cs, stat].

Lucien Krapp

Rue des fontaines 4
CH-1058 Villars-Tiercelin
✉ *lucienkrapp@gmail.com*
github.com/lfkrapp

Employment

2019 – **Ph.D. candidate in Physics**, *EPFL*, Lausanne.
Applied machine learning in nanopore sequencing, Laboratory for Biomolecular Modeling
Supervisor: Prof. Dal Peraro

Education

2018 **Master of Science MSc in Physics**, *EPFL*, Lausanne.

2016 – 2017 **Minor in Computational Science and Engineering**, *EPFL*, Lausanne.

2015 – 2018 **Master in Physics**, *EPFL*, Lausanne.

2012 – 2015 **Bachelor in Physics**, *EPFL*, Lausanne.

2008 – 2012 **Swiss Federal Maturity**, *Gymnase de Beaulieu*, Lausanne.
Advanced mathematics, specific option : mathematics and physics

Experience

Apr 2018 – Jun 2018 **Internship**, *EPFL*, Lausanne.
Internship in the Laboratory for Biomolecular Modeling at EPFL

Feb 2016 – Jun 2016 **Assistant**, *EPFL*, Lausanne.
Assistantship in Object-oriented programming and design, section Electrical Engineering

Sep 2015 – Dec 2015 **Assistant**, *EPFL*, Lausanne.
Assistantship in General Physics I, section Microtechnique

Computer skills

Languages Python, Rust, C++, Shell script, Latex

Utilities PyTorch, TensorFlow, Matlab

OS Linux

Projects

2016–2017 *Correlation between the fission yeast transcriptome and proteome*,
Supervisors: Prof. V. Simanis, Dr. M. Catasta

2017–2018 *Using artificial neural network to enable protein-protein docking*,
Supervisors: Giorgio Tamo, Prof. Dal Peraro & Prof. De Los Rios

Publications

- **L. Krapp**, F. Meireles, L. Abriata, M. Dal Peraro, *Context-aware geometric deep learning for protein sequence design*, **bioRxiv**, 2023

- **L.F. Krapp**, L.A. Abriata, F. Cortés Rodriguez, M. Dal Peraro, *PeSTo: parameter-free geometric deep learning for accurate prediction of protein binding interfaces*, **Nature Communications**, 2023
- F. Cortés Rodríguez, **L.F. Krapp**, M. Dal Peraro, L.A. Abriata, *Visualization, Interactive Handling and Simulation of Molecules in Commodity Augmented Reality in Web Browsers Using moleculARweb's Virtual Modeling Kits*, **Chimia**, 2022
- A. Chiki, Z. Zhang, K. Rajasekhar, L.A. Abriata, I. Rostami, **L.F. Krapp**, D. Boudeffa, M. Dal Peraro, H.A. Lashuel, *Investigating Crosstalk Among PTMs Provides Novel Insight Into the Structural Basis Underlying the Differential Effects of Nt17 PTMs on Mutant Httex1 Aggregation*, **Frontiers in Molecular Biosciences**, 2021
- F. Cortés Rodríguez, G. Frattini, **L.F. Krapp**, H. Martinez-Hung, D.M. Moreno, M. Roldán, J. Salomón, L. Stemkoski, S. Traeger, M. Dal Peraro, L.A. Abriata, *MoleculARweb: A Web Site for Chemistry and Structural Biology Education through Interactive Augmented Reality out of the Box in Commodity Devices*, **Journal of Chemical Education**, 2021
- C. Cao, **L.F. Krapp**, A. N.F. König, A. Radenovic, J.-F. Lutz and M. Dal Peraro, *Aerolysin nanopores decode digital information stored in tailored macromolecular analytes*, **Science Advances**, 2020
- A. Krapp, R. Hamelin, F. Armand, D. Chiappe, **L.F. Krapp**, E. Cano, M. Moniatte and V. Simanis. *Analysis of the S. pombe meiotic proteome reveals a switch from anabolic to catabolic processes and extensive post-transcriptional regulation*, **Cell Report**, 2019

Conferences attended

Oct 2022	CIS SV Retreat (Talk)
Jun 2022	AI4Science (Talk)
Jun 2022	From Solid State To Biophysics X (Poster)
Jan 2022	Pacific Symposium on Biocomputing (Poster)
Sep 2019	Bioinformatics EPFL (Poster)
May 2019	Novartis Leadership Forum Hackathon
Jan 2017	Applied Machine Learning Days (Poster)