APPLICATIONS OF DEEP NEURAL NETWORKS IN COMPUTER-AIDED DRUG DESIGN

by

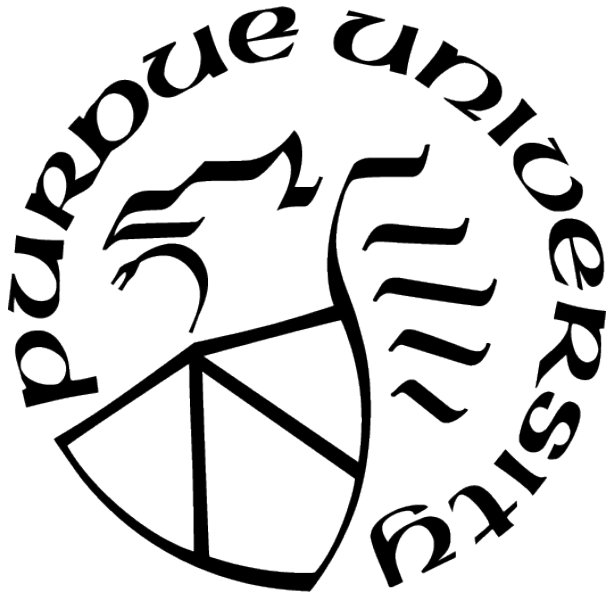
Ahmadreza Ghanbarpour Ghouchani

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



Department of Medicinal Chemistry and Molecular Pharmacology

West Lafayette, Indiana

May 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Markus Lill, Chair

Department of Medicinal Chemistry and Molecular Pharmacology

Dr. Chiwook Park

Department of Medicinal Chemistry and Molecular Pharmacology

Dr. Daisuke Kihara

Department of Biological Sciences and Department of Computer Science

Dr. Elizabeth Topp

Department of Industrial and Physical Pharmacy

Approved by:

Dr. Andy Hudmon

In memory of my father, Dr. Alireza Ghanbarpour,

who left us before his time,

but his presence in my heart, empowered me along the way.

To my mother, Shahrzad,

whose unconditional love continues to nurture my soul.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Markus Lill for his patience, supervision, guidance and support and also granting me the freedom to pursue my own topic of interest.

I am deeply indebted to Dr. Andy Hudmon for his support through my final year of study and I greatly appreciate his mentoring and wisdom. This would not be possible without him.

I also wish to thank my graduate advisory committee members Dr. Chiwook Park, Dr. Daisuke Kihara, Dr. Elizabeth Topp for their feedback and advice.

Finally, many thanks to my friends who were like family to me when I was away from home as an international student, and were with me during sad and happy times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

Å	Angstrom
AAE	Adversarial Autoencoder
AE	Autoencoder
ANN	Artificial Neural Network
ATP	Adenosine Triphosphate
BERT	Bidirectional Encoder Representations from Transformers
CADD	Computer-Aided Drug Design
CNN	Convolutional Neural Network
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DNN	Deep Neural Network
ELMo	Embeddings from Language Models
ELU	Exponential Linear Unit
GAN	Generative Adversarial Network
GCMC	Grand Canonical Monte Carlo
GDL	Generalized Dice Loss
GPCR	G Protein-Coupled Receptor
GRU	Gated Recurrent Unit
HSP90	Heat Shock Protein 90
KDE	kernel Density Estimation
KL	Kullback-Leibler
LSTM	Long Short-Term Memory
MD	Molecular Dynamics
MAE	Mean Absolute Error
MSE	Mean Squared Error
NLP	Natural Language Processing
NMR	Nuclear Magnetic Resonance
NN	Neural Network
PDB	Protein Data Bank

QED	Quantitative Estimate of Druglikeness
QT	Quality Threshold
RF	Random Forests
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SAR	Structure-Activity Relationship
SMILES	Simplified Molecular-Input Line-Entry System
SSP	Secondary Structure Propensity
SVM	Support Vector Machines
VAE	Variational Autoencoder

ABSTRACT

Ghanbarpour Ghouchani, Ahmadreza. Ph.D., Purdue University, May 2021. Applications of Deep Neural Networks in Computer-aided Drug Design. Major Professor: Markus A. Lill.

Deep neural networks (DNNs) have gained tremendous attention over the recent years due to their outstanding performance in solving many problems in different fields of science and technology. Currently, this field is of interest to many researchers and growing rapidly. The ability of DNNs to learn new concepts with minimal instructions facilitates applying current DNN-based methods to new problems. Here in this dissertation, three methods based on DNNs are discussed, tackling different problems in the field of computer-aided drug design.

The first method described addresses the problem of prediction of hydration properties from 3D structures of proteins without requiring molecular dynamics simulations. Water plays a major role in protein-ligand interactions and identifying (de)solvation contributions of water molecules can assist drug design. Two different model architectures are presented for the prediction the hydration information of proteins. The performance of the methods are compared with other conventional methods and experimental data. In addition, their applications in ligand optimization and pose prediction is shown.

The design of de novo molecules has always been of interest in the field of drug discovery. The second method describes a generative model that learns to derive features from protein sequences to design de novo compounds. We show how the model can be used to generate molecules similar to the known for the targets the model have not seen before and compare with benchmark generative models.

Finally, it is demonstrated how DNNs can learn to predict secondary structure propensity values derived from NMR ensembles. Secondary structure propensities are important in identifying flexible regions in proteins. Protein flexibility has a major role in drug-protein binding, and identifying such regions can assist in development of methods for ligand binding prediction. The prediction performance of the method is shown for several proteins with two or more known secondary structure conformations.

PUBLICATION(S)

Chapter 2 is a reprint of the following publication:

Ghanbarpour, A., Mahmoud, A. H., Lill, M. A. (2020). Instantaneous generation of protein hydration properties from static structures. Communications Chemistry, 3(1), 1-19.

Chapter 3 is a reprint of the following preprint:

Ghanbarpour, A., Lill, M. A. (2020). Seq2Mol: Automatic design of de novo molecules conditioned by the target protein sequences through deep neural networks. arXiv preprint arXiv:2010.15900.

1. INTRODUCTION

1.1 Machine learning in computer-aided drug design (CADD)

Machine learning and its subcategories are algorithms that are automatically learned aiming to find patterns in data. The idea of finding patterns in data in the field of drug discovery and rational drug design is not something new, and goes back to early regression methods that were used by Hammet to find associations in reaction rates and equilibrium constants in benzene derivatives [1]. Since then, more complex algorithms have been developed and applied in the field of drug design to find non-linear relationships as well as linear ones, notably, support vector machines (SVM) [2], random forests (RF) [3] and artificial neural networks [4]. All these methods, have their own pros and cons and have their own appropriate uses for suitable problems. In recent years, with the improvement in technology and the availability of high-performance hardware, more complex variants of neural networks have been developed. Especially deep neural networks – neural networks with multiple layers between input and output layers, have gained tremendous amount of attention due to the significant performance gain to solve many problems in science and technology [5].

1.2 Artificial neural networks (ANNs)

ANNs are a collection of interconnected processing units dubbed as artificial neurons (Figure 1.1). Each artificial neuron can be considered as a function which maps input vector $X = [x_1 ... x_n]$ to output y, by multiplying each element x_i of the input vector by weights w_i, adding the bias b and finally passing the output through an activation function f(z). In mathematical terms:

$$y = f\left(\sum_{\mathrm{i}=1}^{n} w_\mathrm{i} x_\mathrm{i} + b\right) \tag{1.1}$$

Neurons can be stacked as layers and each layer receives the output of the previous layer as input. In the most common case, every output from each neuron from a layer is passed to every neuron in the next layer, forming a *fully connected* neural network (see Section: Common neural network types). The networks ultimately will output a label (classification) or one or more scalar properties (regression), (Figure 1.2).

As the name implies, in machine learning methods an algorithm is learned to carry out some specific tasks. In neural networks, the weights in neurons are tuned by optimizing a loss function, which usually measures the error between predicted and actual values of the desired output of the network. In backpropagation the gradient of the loss function with respect to the weights is computed and the weights are changed according to this gradient. Specific optimizing algorithm, e.g. stochastic gradient decent or Adam [6], are utilized in training the ANN. The goal is to tune the weights so that the loss function over the samples during training is minimized.

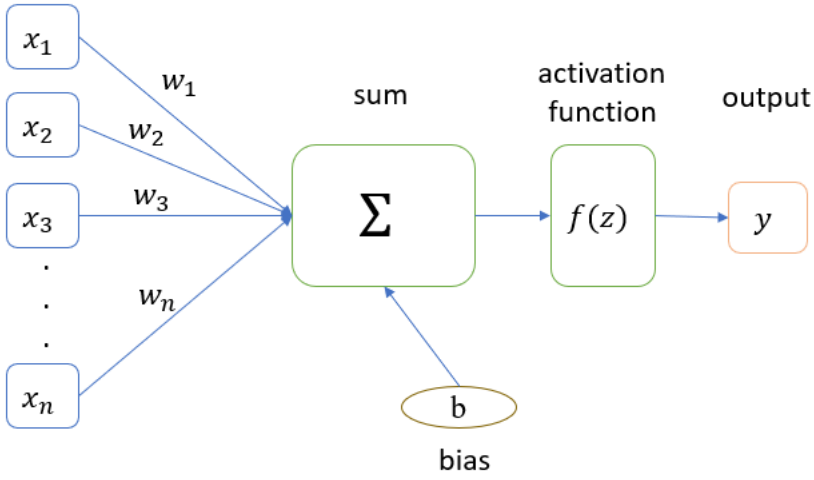


Figure 1.1. Structure of an artificial neuron. Each input is multiplied by a weight, the weighted inputs are summed, and a bias term is usually added. The sum is inputted to the activation function to yield the output.

1.2.1 Common types of neural networks layers

Fully-connected layer

As mentioned before, in fully-connected layers each neuron is connected to each neuron in the next layer (Figure 1.2). These type of layers can be used as general purpose layers in different architectures. However, since every connection has its own weight and every neuron is connected to every other neuron in the next layer, the number of trainable weights can

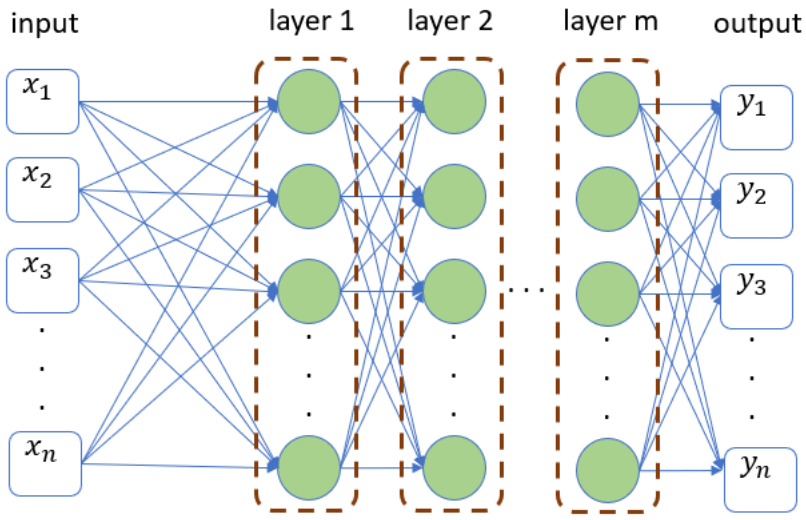


Figure 1.2. A deep neural network with n inputs and m hidden layers. There is a connection from each neuron in each layer to each neuron in the next layer, but not among neurons in the same layer.

rapidly become large, making the training computationally very expensive and the model prone to overfitting. This makes such layers inefficient for processing inputs such as images.

Convolutional layer

Convolutional layers are usually used to process image or image-like data representations [7] (Figure 1.3). The convolution operation $(I \otimes K)$ where I is a single-channel 2D image and K is a convolution kernel with dimensions (k_1, k_2) is linear and can be written for the image pixel position (i, j) as:

$$(I \otimes K)_{\mathrm{ij}} = \sum_{m=1}^{k_1} \sum_{n=1}^{k_2} K_{m,n} I_{\mathrm{i}+m,\mathrm{j}+n} \tag{1.2}$$

A 2D convolutional layer has length, width and depth parameters, where length and width describe the dimensions of its receptive field (kernel) and depth relating to the number of different input channels. For example a depth of three is used for images with RGB color values. In 3D convolution layers kernels with 4-dimensional weight tensors are required. The use of kernels with small number of weights in convolutional layers greatly reduce the number of weight parameters that need to be optimized compared to fully connected layers. This

approach reduces training time and improves convergence speed. Furthermore, translational and to a certain extent rotational invariance is achieved by the convolution mechanism, i.e. feature detection does not rely on where in the input image the feature appears [8].

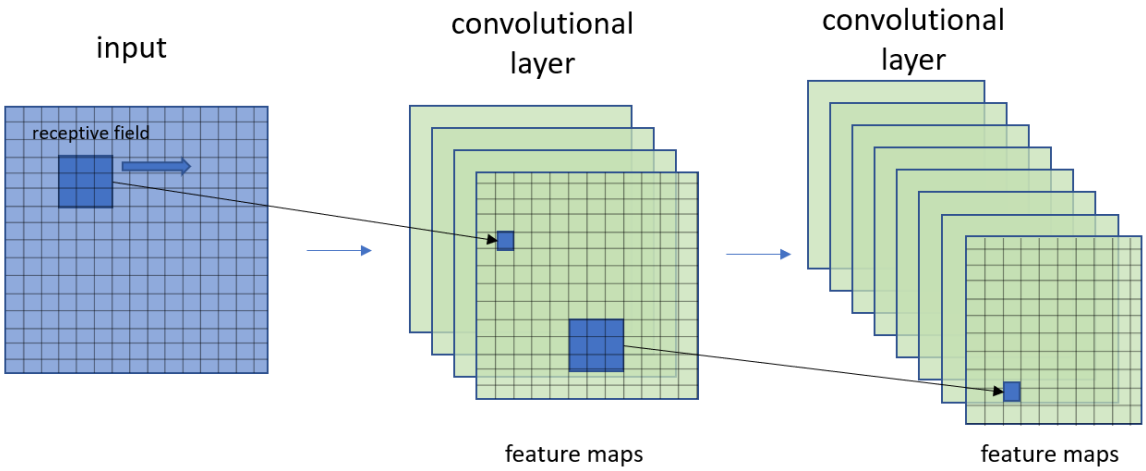


Figure 1.3. Convolutional layers. The convolution operation generates feature maps by sliding a kernel over the input. Feature maps can vary in number depending on the number of kernels used.

Recurrent layers

Recurrent layers are designed to process sequential data, such as time-series, text, protein sequence, simplified molecular input line entry specification (SMILES) sequence, etc. While most common neural networks work in a feed-forward fashion, that is the direction of data flow is always from input to output, recurrent layers enable the flow of data in both directions, by introducing loops (Figure 1.4), and hence, making the output not only dependent on the input of the directly connected layer but also layers before that. In other words, they enable the network to have a "memory" of past inputs. An recurrent layer is composed of multiple cells for each time step of the sequence of inputs being processed. A cell at time step t receives

information from the previous cell state $cell^{t-1}$ in addition to the input x^t to calculate an output. In other words, cell state $cell^t$ is computed as:

$$\text{cell}^{\text{t}} = f\left(Wx^t + U\text{cell}^{\text{t}-1}\right) \tag{1.3}$$

where f is the activation function, W is the weight matrix for the input and U is the weight matrix parameterizing hidden-state to hidden-state connections. RNNs may be unable to handle long-term dependencies in data [9]. To address the issue, architectures such as long short-term memory (LSTM) [10] and gated recurrent unit (GRU) [11] were developed.

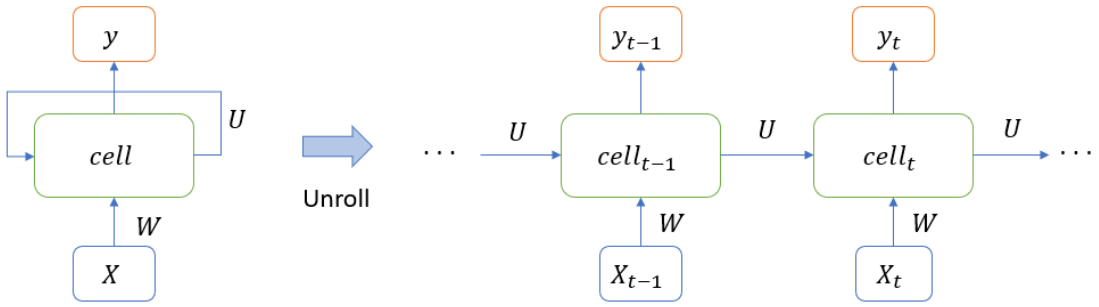


Figure 1.4. A RNN cell shown in both rolled and unrolled form. W is the weight matrix of the input, while U is the weight matrix for the hidden-state to hidden-state connections.

1.2.2 Building models

The first step in building models is to clearly explain the problem of interest. Next steps are usually preparing the data, designing the neural network architecture, training and finally, evaluation. Designing architectures should be done not only with consideration of the input data representation, but also the desired output in mind. In terms of output, the model is aimed to either make predictions about the input, which is done by a discriminative/predictive model, or to generate new data similar to the input data which is carried out by using a generative model (see Section: Generative models). In discriminative/predictive tasks, the goal is usually to classify or to predict one or more values for each input sam-

ple. The output in these cases is usually a one-dimensional vector containing class labels or/and predicted values. However, in generative models, the output can vary. In the field of drug discovery, the generated output as well as the input can be image-like, text-like (e.g. sequence of characters), or graph-like.

In modern deep neural networks approaches, the goal is to automatically extract features by the model, rather than explicitly define features before model building. This allows the the model to detect hidden patterns in data, not obvious to the human eye. The advantages of such models is the automatic feature extraction and data abstraction [12]. Using this approach requires one to convert the data to commonly used data representations for neural networks. Some common types of representations are discussed below.

1.2.3 Building models based on data representations

Visual representations

Structural information is of high value in tasks related to structure-based drug design. Macromolecular and small molecule 3D structures can be described using voxels, so that essential and useful structural information is extracted during learning, without much human interaction. Convolutional neural networks (CNNs) are usually used for processing image-like data. CNNs can be fully convolutional or a combination of convolutional and fully-connected layers. To convert structures to 3D image-like data, initially the 3D structure of molecules, that is, the coordinates of the atoms has to be voxelized. In order to reduce the sparsity of 3D grids, sometimes Gaussian smearing is used, which applies a Gaussian function with the atom coordinate as the center [13]. Multiple grids may be generated for multiple atom types, and may be provided to the neural networks as different input channels. The drawbacks of this approach are: First, the grids can only be generated in a fixed predefined image size, which is sub-optimal for proteins or ligands that can adapt largely variable size. Thus, a whole molecule may not fit into the grid. Second, the grids are not transformation and rotation invariant, so the orientation of the molecule affects the training and prediction. To mitigate this problem, data augmentation is used, i.e. providing the same structure in a variety of orientations and transformations during the training.

Many works have used CNNs to process protein and molecular structures. For example, 3D convolutions have been used for protein's binding site similarity prediction [14], protein-ligand binding affinity prediction [15], protein's binding site detection [16] and classification [17], ligand's pose prediction in binding [13].

Sequential representations

Proteins and genes have long been represented as sequence of their building blocks namely, amino acids and base pairs. Similarly, small molecules structures can be written as a sequence of characters describing the atoms and the bonding among them, commonly using SMILES language. Representing data as a sequence of characters enables application of methods originally developed for natural language processing (NLP) tasks to to biological and chemical data. Recurrent neural networks (RNNs) are commonly used to process this type of data (see Section: Generative models: Recurrent neural networks).

Graph representations

Graph representations are natural to chemicals but have also been previously used to model and analyze proteins [18]. In recent years neural network methods have been developed to process graph data. 2D or 3D structures of small molecules and proteins are encoded as graphs. Typically, the atoms or amino acids are the nodes and edges represent the bonds or distances. Atom and bond types can be passed on to the network as node and edge features. Here, the data represented is rotation and transformation invariant due to the data being a graph instead of images. Graph convolutions combined with fully-connected layers may be used to process graphs and extract features. Works such as [19] generate graph-based fingerprints of small molecules to be used in downstream tasks, graphs have also been used in protein design [20] and generative models (See Section: Generative models).

1.2.4 Generative models

Unlike predictive models which aim to predict some target value based on input data, the generative models seek to learn the true distribution of the data they are trained on,

so to generate new data with features learned from the training set. In other words, generative models aim to learn a function that estimates the true distribution of the data [21]. Currently, in the field of drug design and discovery, generative models have been used to generate de novo molecules (see sections on model types), where usually molecules are represented as SMILES strings, or molecular graphs. Also, generative models have been used for protein modelling and design, where proteins were represented as their distance matrices [22]. Different variants of generative models are as follows:

Recurrent neural networks

Inspired by their successful applications in NLP tasks [23], RNNs have been used to generate text that represent molecules as SMILES sequences, for de novo molecule generation [24]–[26]. The ability of RNN to hold internal states enables such networks to maintain a memory of previous characters, hence making them suitable for processing sequential data. Therefore, in order to train such networks data must be represented as a sequence. For this reason, molecules are usually represented using the SMILES language, which uses a specific grammar to represent 2D structures of molecules. Newer variants of RNNs, namely LSTMs and GRUs has shown improved performance over simple RNNs [27]. Once a network is trained on a series of molecules represented as SMILES strings, it is able to generate new and valid molecules that did not appear in the training data. However, given that SMILES strings are very sensitive to errors, there can be a large fraction of invalid strings that do not produce a valid structure.

Autoencoder-based models

Autoencoders are a type of neural network architecture which learn a mapping from input data to latent representation and back to a reconstruction of the input data, using an encoder and decoder network. Autoencoders are not generally used as generative models. However, their variants, namely, variational autoencoders (VAE) and adversarial autoencoders (AAE) [28] (Figures 1.5 and 1.6, respectively) are able to generate new data. Autoencoders are composed of two compartments, an encoder and decoder. The encoder

reduces the high-dimensional input data to a low dimensional representation or embedding z, while the decoder reconstructs the data from the embedding. With $P(z)$ as the prior distribution of z, let $Q(z|X)$ and $P(X|z)$ be probabilities of encoding and decoding distributions. $Q(z|X)$ and $P(X|z)$ can be estimated by training the VAE [29]. In AAE, an additional discriminator network is introduced to add additional constraints on the training. Both architectures have been used mainly for compound generation [28].

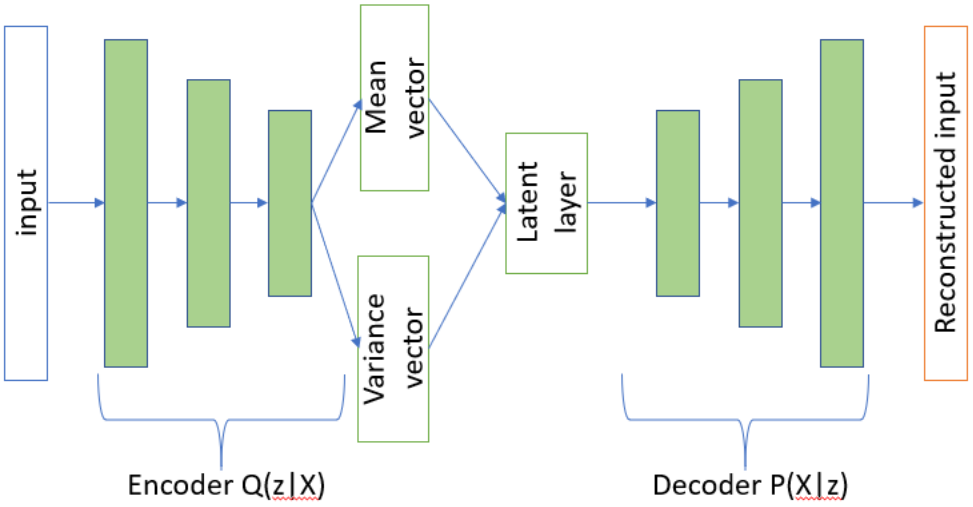


Figure 1.5. Architecture of a VAE.

Generative adversarial networks (GANs)

A GAN model is composed of two neural networks: The generator and the discriminator that compete with each other in an adversarial fashion (Figure 1.7). Both networks are trained in parallel. In its simplest form, the generator learns the mapping of an embedding to some output which resembles the training data, and the discriminator evaluates the generator's performance by deciding whether the generated samples are real or fake. The discriminator itself is trained by learning to distinguish between the real data in the training set and the generated (fake) data sampled from the generator. Finally, the trained generator is used to generate new data after the training is completed, usually when the loss function of generator and the discriminator do no longer decreases in value [30]. GANs have been used broadly in image generation due to their impressive performance, however, training

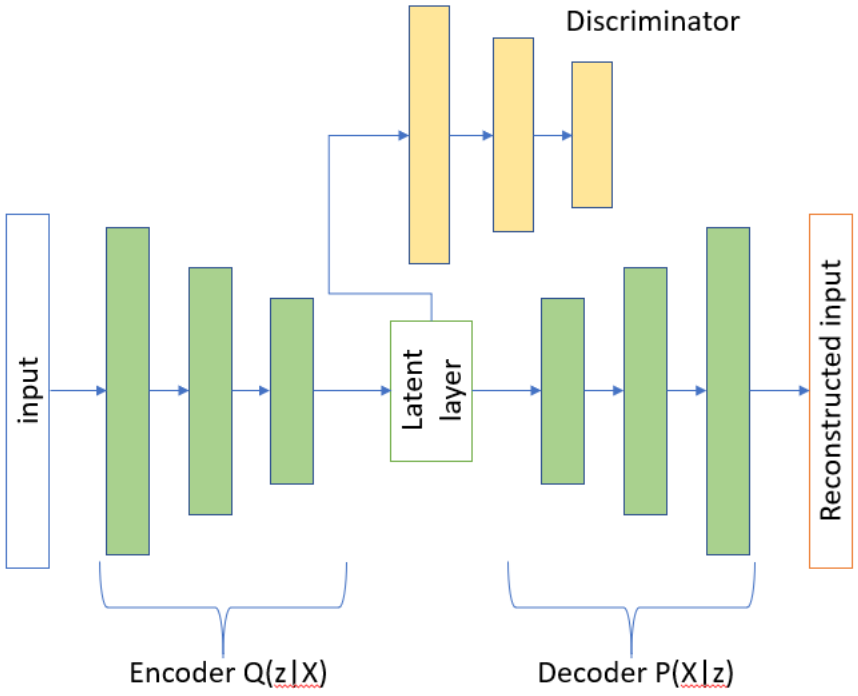


Figure 1.6. Architecture of an AAE. In this design, a discriminator is added to the autoencoder.

GANs is more challenging than other models since the adversarial balance should always be maintained between the generator and the discriminator during the training, otherwise problems such as *mode collapse* can occur [31]. If *mode collapse* happens, the generator only generates more or less the same output which has poor diversity making the model not useful for generating new data. Methods such as MolGan [32] and ORGANIC[33] employ GAN models for de novo molecule generation.

Reinforcement learning (RL)

Generative models can be further tuned using reinforcement learning. Initially developed to make decision making possible in uncertain environments, reinforcement learning can guide the molecular generation toward molecules with desired properties. In reinforcement learning, external scoring functions can be used in order to reward the model to find a policy toward maximizing the reward, i.e. by finding the best actions through trial and error.

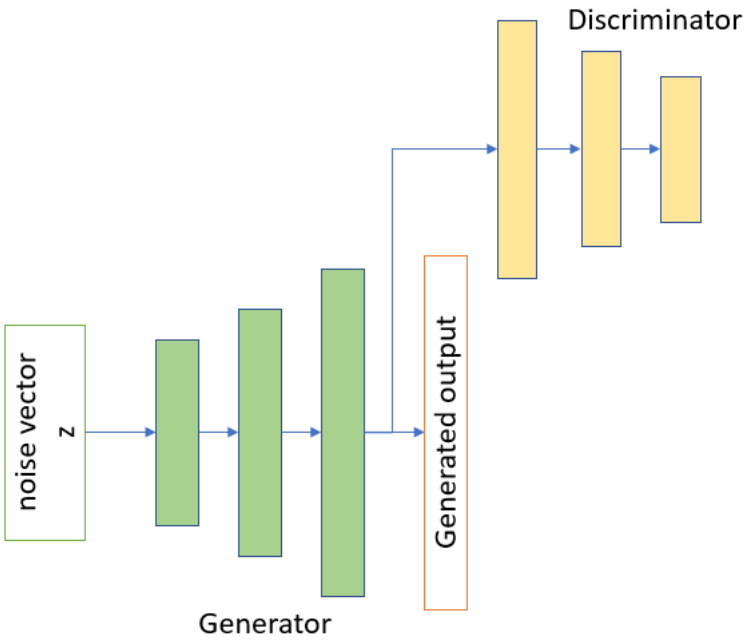


Figure 1.7. Structure of a GAN. The network is composed of two networks, the generator and the discriminator. Both networks are trained in an adversarial fashion simultaneously.

Reinforcement learning has been used in conjugation with other generative approaches to guide the generative process [32], [34]–[36].

1.3 Promises and strengths

Automatic feature extraction

A very useful property of deep neural networks is automatic feature extraction. Feature engineering is a time-consuming process and requires expert knowledge of the field. Automatic feature extraction enables the model to extract essential, sometimes complex or hidden features from the data for the task at hand without being informed about them beforehand. Such advantage enables building machine learning models faster and more efficient. Therefore, the data can be provided as is, images, text, etc with minimal preprocessing required.

Generative power

Another important advantage of deep neural networks which is not so common in other machine learning approaches is their ability to generate new data. Generative models can be useful in protein and peptide design, de novo design of small molecules, etc.

Reinforcement learning

A very useful characteristic of deep neural networks is their ability to be combined with reinforcement learning algorithms which gives rise to a new set of methods known as deep reinforcement learning. Reinforcement learning algorithms are useful in finding policy in uncertain environments, that is, when the goal is not to predict a specific target value, but rather to identify an overall optimal policy. These algorithms may be used in situations where the scoring function is unknown or there is no differentialable loss function that can be used for backpropagation. For example, in the case of generating compound in a specific Log P range, the target compounds are not known. However, the model output can be scored by a scoring function to direct the network in finding a policy which generates compounds within the desired Log P range.

1.4 Challenges and caveats

The black box and interpretability problem

Over the past years many machine learning models have emerged that were able to solve complex problems in a variety of fields [5]. However, these models have typically been a black box to the users and even developers and the exact inner workings of them are unknown. The lack of "interpretability" may makes these models less reliable, especially in sensitive tasks, since unknown errors may occur and the reliability of predictions is not exactly known. This is particular true when they are unintentionally trained on erroneous data, or data not representative of the real-world data. Without the knowledge of the inner working of the black box, it is difficult to reliably assess if the model has learned a concept or the training

occurred by chance. Therefore, to address such issues, there should be mechanisms to assess the model's learning and making the model more interpretable.

Overfitting and related issues

All machine learning models are susceptible to overfitting and deep neural networks are no different. Overfitting can occur due to small training data size or poor design of the model. Being a black box, it may be hard to distinguish if a model can generalize, if validation and test sets are not chosen carefully. For a model to generalize, the training, validation and test data should be diverse and representative of the real world data. Lacking diversity can cause the model have bias and may be only performing well on a small set of data. It's also important to prevent data leakage from poor preprocessing or train/test-set splits. There are measures used to mitigate the problem of overfitting, namely data augmentation methods, which increases the dataset size by slightly modifying copies of the samples in the dataset, or transfer learning, which uses a model already trained on a larger, similar dataset and subsequently continue the training of the model on the specific dataset of the problem at hand; often only a subset of parameters are optimized throughout this final training process.

Finding causal relations

The well-known phrase: "Correlation does not imply causation" has been used in the field of statistics to refer to the inability to interpret a cause-and-effect relationship between variables on the basis of their association. Machine learning methods tend to find correlations and associations in data, rather than causal relations, therefore the interpretation of the results should be done with caution. Some features may just happen to co-occur in the data, which may cause the model to associate them while it does not mean that those features explain the concept the model is trying to learn. Subsequently, the model may make a wrong decision or prediction by the observed association in the training data. In recent years, there have been studies to develop methods to find causal relations between features and predictions [37].

1.5 Future directions

Deep neural networks have greatly impacted research in drug discovery. However, challenges still remain. As mentioned previously, many methods deploying deep neural networks for drug design were originally designed for image and text processing. While the methods can be applied with minimal changes to solve the problems in drug discovery, their corresponding input data representations may pose some issues. Proteins represented as voxels do not convey any information regarding bonding. Image transformations and rotations may alter model performance and image sizes are fixed in convolutional networks, posing problems in representing proteins with different sizes. Text representations sometimes suffer from typing errors that cause the model's output to be invalid molecular structures. On the other hand, graph representations can be used to represent bonds and atoms, and the lengths of the graphs can be variable, hence there is no such concerned of fixed input size. Graphs have been used to represent proteins as well as small molecules [19], [20], and seems to be the better approach compared to images or text.

Another challenge, is the availability of problem-specific data. Generating experimental data used for machine learning tasks is expensive. Labeling data is another expensive process that is not always available. Without high-quality data, even the best algorithms can fail. One solution for such problems are to move toward models that can perform with small dataset sizes, or models that can auto-label data or perform in unsupervised manner, that is being able to make predictions using unlabeled data. Recently, there have been works in auto-labeling data, and also self-supervised and unsupervised learning in text and image-classification [38]–[40]. Similar approaches may be used in biological and chemical data to develop models.

The community should also move forward to address the black box problem in machine learning models. There have been approaches known as *saliency methods* that use different techniques to find regions in image data that contain the prominent features the model performs based on. However, the explanations derived using such methods can sometimes be unreliable if they are sensitive to variables that do not affect the model's prediction [41]. Explainable models can not only do the task, but also may improve our understanding of

the hidden biology and chemistry in data. Such models will be more robust, reliable and less prone to errors. Explainable models can minimize the adverse effects of faulty data and human error in the process and may also be used as knowledge discovery tools, while performing for the task they are designed to carry out.

1.6 Scope of the present study

In this dissertation, I present three applications of deep neural networks in drug discovery, each addressing a specific problem, using different approaches and data representations. I show how automatic feature extraction and data abstraction in deep neural networks enables the models to yield reasonable results, both in predictive and generative types. Chapter 2 describes two deep learning methods to predict protein hydration data, one of which is used to also predict thermodynamic information of water molecules. Each model uses its own data representation: The first model uses image-like input data generated from proteins, while the second uses spherical harmonics expansion to describe the protein environment. In chapter 3 a generative model is described, which designs de novo molecules conditioned by protein sequences, using the features derived by unsupervised learning via another model. The model is then modified using reinforcement learning to increase diversity and novelty. Finally, chapter 4 shows how a network can learn to predict protein disorder solely from sequences, by directly being provided the sequences as text and the disorder values.

2. INSTANTANEOUS GENERATION OF PROTEIN HYDRATION PROPERTIES FROM STATIC STRUCTURES

2.1 Introduction

The prediction of thermodynamic properties of biochemical systems such as Gibbs free energies is critical in understanding and quantifying essential biological process such as protein folding, protein-ligand and protein-protein binding. Resource intensive molecular simulations are routinely used to sample atomistic configurations of the dynamic biochemical system in order to calculate thermodynamic properties. Recently, machine learning methods have been explored to accelerate and improve configurational sampling of protein systems in comparison to molecular dynamics (MD) simulations [42]–[49]. This acceleration is achieved by machine learning concepts that learn collective variables from MD trajectories [44], [45], [48], [49] or that generate new atomistic configurations in a statistically independent manner [42], [46], [47]. The focus of these methods lies in the thermodynamic characterization for structural studies of proteins. Application of these machine learning approaches to investigate the thermodynamic properties of biochemical processes such as protein-ligand or protein-protein binding is still to be explored.

(De)Solvation of protein and ligand is typically a driving force for such association processes. The thermodynamic properties of water molecules around protein moieties depend strongly on the formation and dynamics of hydrogen-bond networks in a heterogeneous protein environment. Several methods [50] have been devised to identify water molecules adjacent to proteins' surfaces which includes knowledge-based methods such as WaterScore [51] or AcquaAlta [52], statistical and molecular mechanics approaches such as 3D-RISM [53] or SZMAP [54], Monte-Carlo methods such as grand-canonical Monte Carlo (GCMC) simulations [55], and MD methods such as WATCLUST [56], WaterMap [57], [58] or WATsite [59]–[61]. GCMC- and MD-based hydration-site prediction is accurate and widely accepted as gold-standard to compute the likely water-positions in the binding sites of proteins, and the enthalpy and entropy contribution of a replaced water molecule to binding free energies. This statement was confirmed in a recent analysis on the structure-activity relationships for different target systems which demonstrated the superiority of simulation-based water pre-

diction compared to other commercial methods such as SZMAP, WaterFLAP and 3D-RISM [62].

Hydration information can be used to estimate the desolvation free energy contributions to a ligand's binding affinity or the potential for water-mediated interactions [58], [63], [64]. Grid-based adaptations of the inhomogeneous solvation theory (IST) [65], for example GIST [66], have been developed for direct inclusion of the hydration information in docking algorithms.

In addition to water replacement and reorganization, ligand binding typically also involves conformational changes of the protein [67]. Recently, we demonstrated the influence of conformational changes of the protein on hydration site positions and thermodynamics [68], [69]. These studies concluded that hydration site prediction on flexible proteins needs to be performed on alternative protein states. Furthermore, we recently demonstrated the general importance of water networks around the bound ligand for forming enthalpically favorable complexes [70]. Thus, it is indispensable to re-calculate hydration information in an efficient manner for each bound ligand or even binding pose during docking.

Hydration-site prediction based on GCMC- and MD-simulations is accurate but also rather time-consuming. Utilization of these concepts in a real-world compound-design project on flexible proteins and large sets of ligands with alternative binding poses is therefore difficult to attain with current computer hardware and therefore currently impractical. A significantly more efficient method for hydration profiling is necessary, that would allow its incorporation in virtual screening to dynamic and flexible protein entities. In this study, we provide evidence that modern machine learning approaches may present a realistic solution for obtaining thermodynamic hydration information in an efficient manner; we present the first deep learning methods that instantaneously predict the thermodynamics of hydration data (Figure 2.1).

First, we demonstrate that simple machine learning methods based on local descriptors that characterize the direct interaction between protein and a potential water molecule at a specific position in the binding site are insufficient to predict hydration information. The reason for this observation is that interactions among water molecules are critical for stabilizing the hydration pattern in binding sites, forming energetically favorable water networks

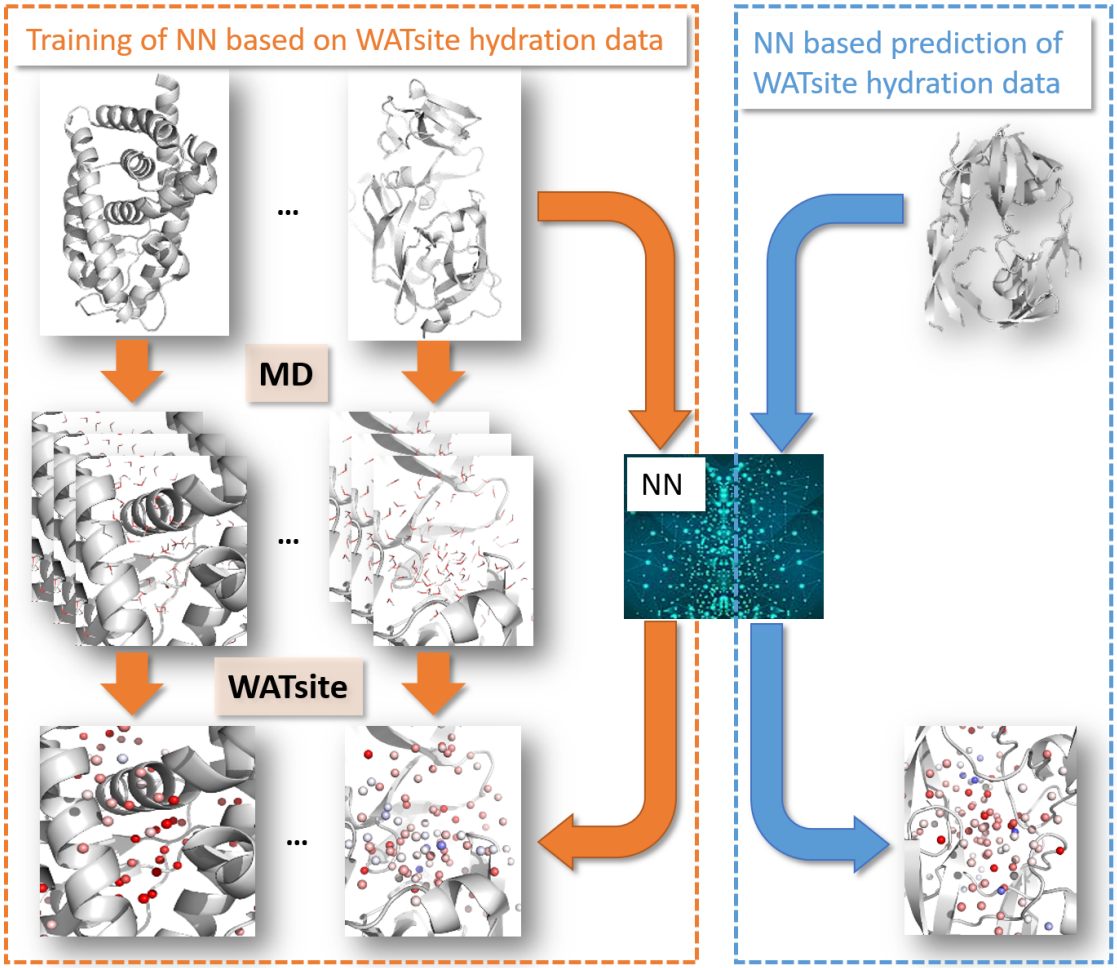


Figure 2.1. Overall idea of WATsiteOnTheFly. A neural network is trained to generate thermodynamic hydration data based on static protein structure. This allows efficient calculation of (de)solvation data without performing MD simulations.

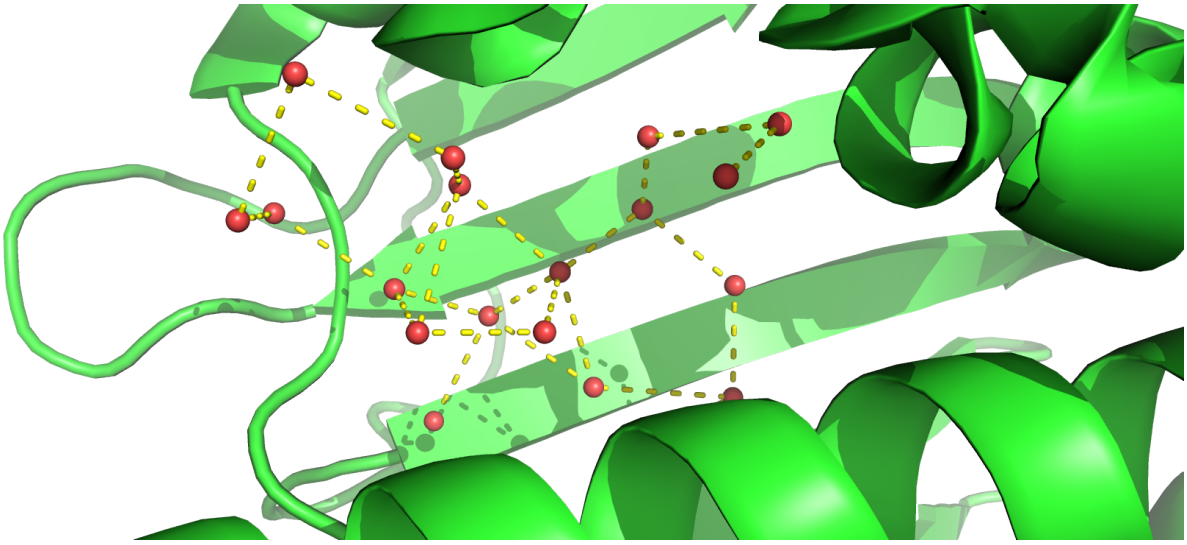


Figure 2.2. Network of water molecules in binding sites. Example of crystallographic water molecules in the binding site of the apo structure of HSP90 (PDB (Protein Data Bank)-id: 1uyl). As water molecules in the binding site are stabilized by hydrogen-bond interactions, models that rely purely on protein-water interactions fail to represent the thermodynamic state and therefore to predict position, enthalpy and entropy of water molecules.

(Figure 2.2). The importance of multi-body effects for the prediction of thermodynamic properties of hydration was also emphasized in previous studies [71]. To correctly model and predict hydration data, more complex machine learning methods need to be designed that include potential water interactions. We have designed two different machine learning concepts based on deep neural networks that include those multi-body effects which are critical determining the positions and thermodynamic properties of water networks (Figure 2.3).

Based on convolutional neutral networks (CNN), the first approach aims to predict hydration information of all grid points in the binding site in a single calculation. First, interactions are computed between protein and multiple atomistic probes placed on a 3D grid encompassing the binding site. Those interaction grids, called molecular interaction fingerprints (MIF), are then used as input to the CNN to predict hydration occupancy. Due to the use of spatial kernels in CNN, correlations between neighboring grid points are incorporated. This allows to implicitly include water-water interactions in the machine learning model.

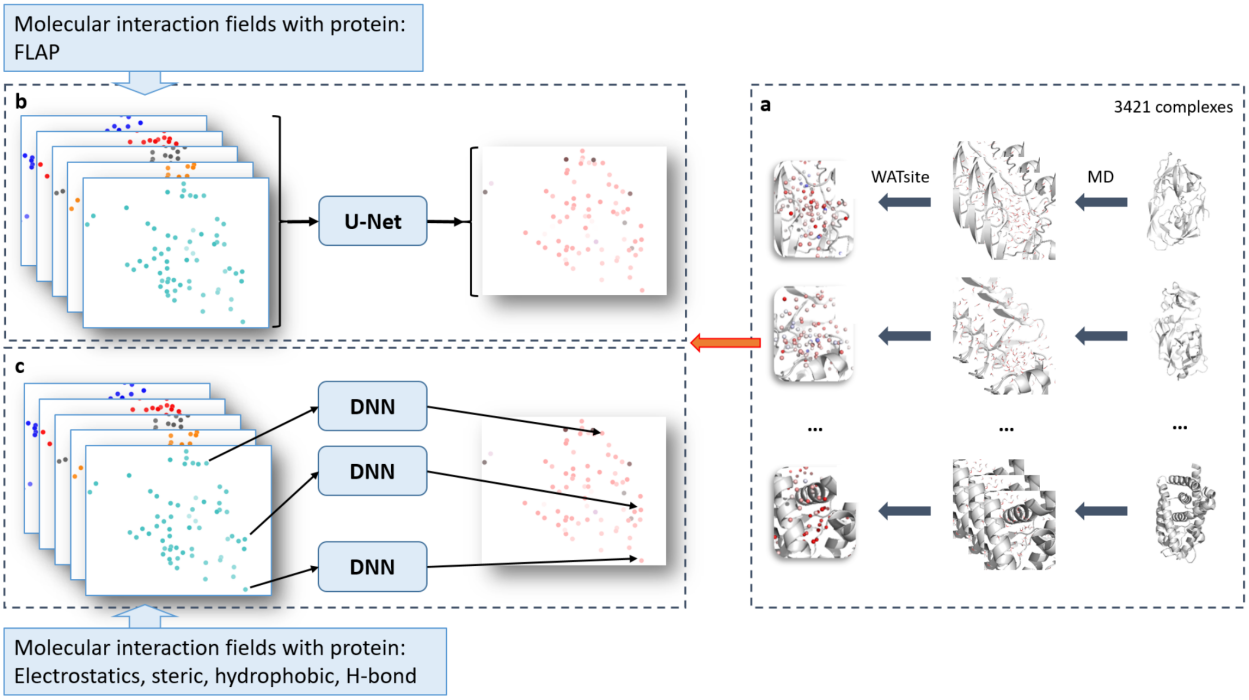


Figure 2.3. Overall procedure of prediction of WATsite data using neural networks. (a) WATsite simulation are used to generate hydration data. Data is used as output layer for training of neural networks. (b) Direct prediction of complete 3D hydration image using U-Net approach. (c) Point-wise prediction using simple fully-connected neural network.

In contrast, the second model predicts hydration information for each grid point separately using spherical-harmonics local descriptors. Again, interactions between protein and atomistic probes are mapped on a 3D grid. Spherical-harmonics expansions of those interaction maps around each grid point then encode the local environment of a potential water molecule which includes protein-water and water-water interactions.

Both models are trained on a large data set of thousands of protein structures. For each protein structure, MD simulation is performed. Subsequent WATsite analysis predicts hydration density and thermodynamic profiles on a 3D grid. This hydration data on the grid functions as ground truth throughout the training and validation of the neural network (NN) models. After the model has been trained it can be applied to any static protein structure without the need to prepare and run any MD simulations.

2.2 Methods

2.2.1 Water prediction on proteins

Here, hydration site data was generated for several thousand protein systems using WATsite (Figure 2.4). The recently published protocol combining 3D-RISM, GAsol and WATsite (Figure 2.4) was used to achieve convergence for hydration site occupancy and thermodynamics predictions for solvent-exposed and occluded binding sites [61]. Using 3D-RISM site-distribution function [72]–[74] and GAsol [75] for initial placement of water molecules, WATsite then performs explicit water MD simulations of each protein. Finally, explicit water occupancy and free energy profiles of each hydration site (i.e. high water-occupancy spot) in the binding site are computed. This hydration data is distributed on a 3D grid that encompasses the binding site and is used as output layers for the neural networks to be trained on. Details on WATsite simulations and analysis can be found in the Supplementary Methods section.

2.2.2 Neural networks for WATsite prediction

Two different types of neural networks have been designed to predict hydration information (Figure 2.3 a). In both approaches, input descriptors were generated for each grid point

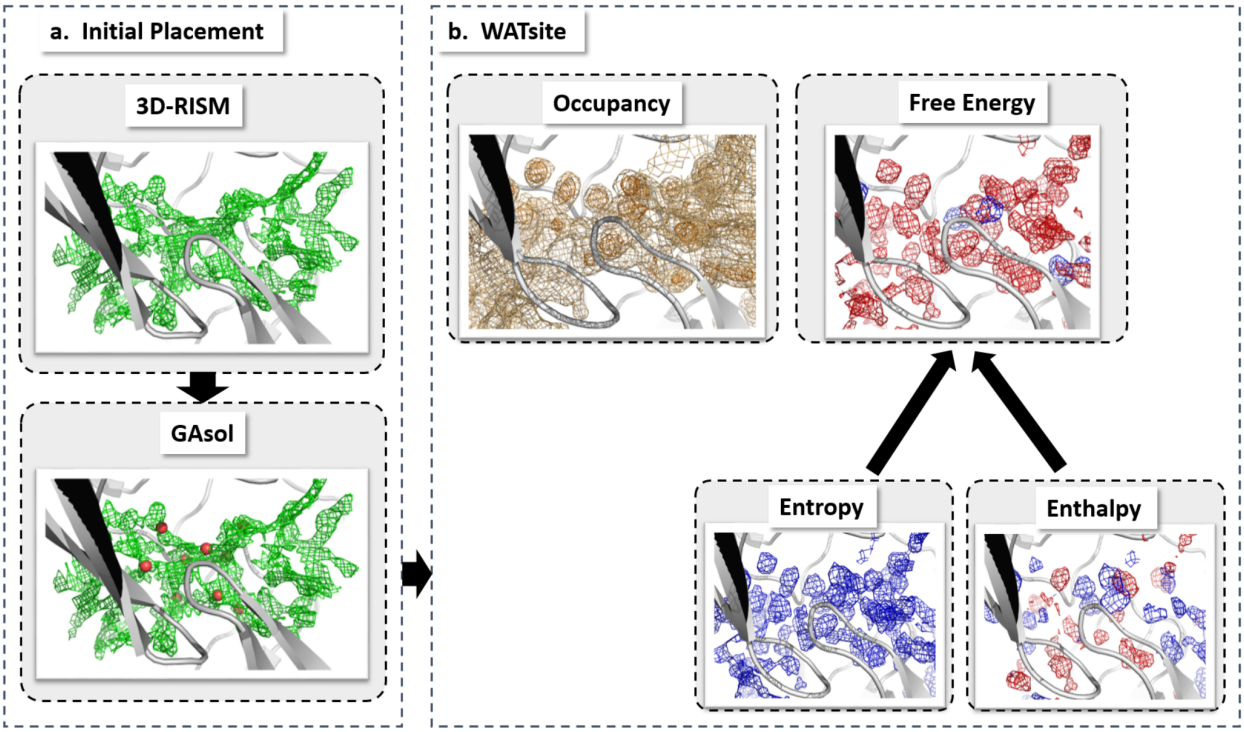


Figure 2.4. Overall procedure of WATsite. Overall procedure of WATsite combining (a) initial placement of water molecules using 3D-RISM and GAsol, and (b) subsequent MD simulation with explicit water molecules and WATsite analysis to generate water occupancy, enthalpy and entropy grids (adapted from [70]).

representing the spatial and physicochemical environment of that potential water location. In the first approach, the complete 3D input grid was translated into a 3D output grid representing the hydration information using a semantic segmentation approach (Figure 2.3 b). In the second approach, the hydration information of each individual point is predicted based on input descriptors (Figure 2.3 c).

Neural networks for semantic segmentation

In the first approach, to predict hydration data, we adapted deep neural network concepts commonly used in semantic image segmentation. Semantic image segmentation is the task to identify the pixels in an image that belong to a specific class or category, for example a specific object in an image. The great advantage of such networks is that they are able to be trained end-to-end by creating a mapping from the input layers to the output images. The resulting output is an image or a grid with the same dimensions as the input layers. Among the various architectures used for this task, U-Net has been demonstrated to often produce superior segmentation performance with smaller training sets compared to other methods [76]. Here, we used different forms of U-Nets but extended the segmentation task to multi-class segmentation. The multiple classes represent the occupancy of water molecules above various threshold values in different moieties along the protein surface.

Generation of descriptors

We used the "refined set v.2016" from the PDBind database [77], [78] consisting of 4057 protein-ligand complexes. Hydration site data was generated using WATsite as described in [70] (see also Supplementary Methods). The ligands were removed from their binding site for WATsite calculations but used to define the center of the hydration grids where the center of the grid is aligned to the ligand centroids in the X-ray structure.

All PDB files were processed by removing ions, water molecules, ligands and other heteroatoms. No proteins with cofactors in the binding site were used in this study. Preparation scripts available in WATsite's docker image bundle were used to further process the proteins: PROPKA [79], [80] was used for protonation state prediction and LEAP (part of the Amber-

tools package [81]) for assignment of Amber14 force field parameters. The prepared protein was used as input for WATsite and for the fully connected network (to generate features with spherical harmonics expansion method).

For the CNN-based approach molecular interaction fields (MIF) with different atomistic probes distributed on a 3D grid are used as input. MIFs are generated by first placing a fictitious probe molecule on each point of a 3D grid that encompasses the binding site. The interaction value between probe and protein is calculated at each grid point under the assumption of a rigid protein structure. Instead of providing an image of the protein, this approach rather generates a negative image of it and provides data for the binding site regions of the protein unoccupied by protein atoms but accessible to water molecules.

Molecular interaction fields (MIF) with different atomistic probes distributed on a 3D grid are computed using FLAP [82], [83] and are fed as input descriptors for the CNN. FLAP uses the GRID forcefield and its own atom types. The internal program GRIN [84], [85] is used to preprocess the protein. Additional details can be found in the Supplementary Methods section. The descriptor grids were aligned and interpolated to the WATsite grids by use of the MDAnalysis package [86], [87]. The process for selecting relevant chemical probes for FLAP is further explained in Section *Probe selection*. FLAP occasionally failed to generate output for one or two probes for some proteins due to an internal program issue. As this is a commercial software, it was not possible to correct this error. PDB files for which FLAP failed to generate an output were removed. Finally, 3421 PDBs were used for training and testing of the neural network models (Supplementary Data 2 and 3).

Probe selection

In FLAP, MIFs between protein and 78 different chemical probes are generated. To reduce the number of input layers for the CNN model, we performed k-means clustering of the FLAP grids of three randomly selected protein systems. The distance matrix used during clustering was based on Pearson correlation coefficients between the interaction values on the 3D FLAP grids of a pair of probes. In detail, the distance between two interaction probe types was defined as one minus the Pearson correlation coefficient. The number of clusters

was chosen to be 12. One representative probe type from each cluster was used to finally generate a set of 12 representative probes with largest diversity between their interaction grids, i.e. smallest Pearson correlation coefficient. These grids represent 12 input channels to the neural network. Increasing the number of channels (probe types) did not lead to significant improvement of the network and only increased the training time.

Processing of hydration occupancy data

Initially, the generated neural network models were designed to generate regression models to predict continuous occupancy values. These models, however, failed due to significant imbalance between low and high occupancy values (Supplementary Figure 1). Alternatively, we proceeded with a multi-class segmentation model with six output channels. Each of those channels represents the water occupancy above a chosen threshold. In detail, WATsite occupancy values were transformed into labels based on the threshold values that were selected for the network. The threshold values were 0, 0.02, 0.03, 0.045, 0.06 and 0.07. Input data grids from FLAP were clipped at -20 and 20 kcal mol^{-1} and scaled to be within -1 and 1, to remove the rare, extreme values. This range covers more than 99 % of all points (Supplementary Figure 2).

Network architecture and model building

Our neural network architecture was based on the work in [88], with the difference that in our implementation, the network contained six output channels. In detail, a modified version of a U-Net neural network was used which contains Residual connections and Inception blocks. Residual connections were first introduced in ResNets [89]. They have the advantage of preserving the gradient throughout a deep neural network addressing the vanishing gradient problem of those networks.

Another issue is the optimization of the kernel size of the convolutional filters. Suboptimal kernel sizes can lead to overfitting or underfitting of the network. Inception blocks have been designed to overcome this issue, whereby the Inception blocks contain convolutional layers with different kernel sizes running in parallel. Throughout the training process,

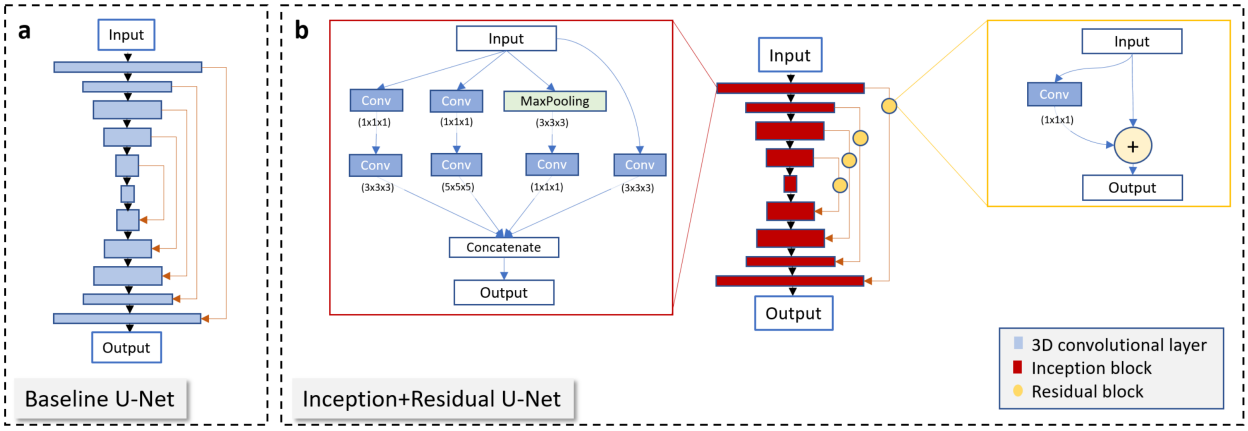


Figure 2.5. Network architectures. (a) Baseline U-Net and (b) Inception+Residual U-Net architecture used for multi-classification model for hydration density prediction.

the network learns to use the layers with convolutional kernel size that best fits the input data which results in better training process [90].

The U-Net that we used as a baseline model for our experiments consists of 6 encoder and 5 decoder layers (Figure 2.5 a and Supplementary Figure 3a). Each layer has a 3D convolutional layer with kernel size 2, stride size of 2 and zero padding. The number of filters for layers 1-6 is 32, 64, 128, 256, 512, and 512, respectively. Each convolutional layer was followed by a Batch Normalization layer, a Dropout layer and LeakyReLU activation. Each decoding layer consists of an Upsampling3D layer with size 2 followed by a convolutional layer, Batch Normalization layer, Dropout, and concatenation layer (which provided the skip connections in the U-Net) and ReLU activation. The number of filters for layers 7-10 is 512, 256, 128, and 64, respectively. The last layer consists of six filters (for the classification of 6 thresholds).

The Inception+Residual U-Net that we used resembles a U-Net, with the exception that each convolutional layer is replaced by an Inception block and the skip-connections contain a Residual block (Figure 2.5 b and Supplementary Figure 3b). Inception and Residual blocks and convolutional layers are followed by ReLU activation. The network has 5 encoder layers and 4 decoder layers. All Inception blocks are followed by a Dropout layer. Each

decoder layer has an Upsampling3D layer prior to the Inception block. The last layer is a convolutional layer with filter number of 6 and kernel size 1.

As discussed above, regions in the grid with high water occupancy are sparse by nature, resembling a significant imbalance between number of low-occupancy and high-occupancy grid points. This makes the prediction of higher occupancy grid points difficult, as commonly used loss functions such as mean squared error will not work properly for such imbalanced data. The sparsity of the dense regions causes the network to predict low or zero values for all grid points even for high occupancy points. This problem also occurs in image segmentation tasks, where the object of interest is small compared to the whole image being analyzed, for example in the detection of small tumors in brain images [91]. One of the loss functions that has been designed to train such imbalanced data is the Dice loss, which is a modified, differentiable form of the Dice coefficient [92]. We used the generalized form of the Dice loss (GDL) [91] which assigns higher weights to the sparser points:

$$GDL = 1 - 2\frac{\sum_{l=1}^{6} w_l \sum_n r_{ln} p_{ln}}{\sum_{l=1}^{6} w_l \sum_n (r_{ln} + p_{ln})}$$

with label weights $w_l = 1/(\sum_{n=1}^{N} r_{ln})^2$ proportional to the inverse of their populations squared. r_{ln} and p_{ln} are the reference and predicted label (l) values at a grid point n, respectively [93]. This loss function will strongly penalize sparse grid points, enforcing the learning algorithm to more precisely predict those values in addition to the large number of low-occupancy grid points.

Adam optimizer [6] with learning rate of 0.001 and a batch size of 16 was used for training the model. Learning was performed for 100 epochs using Keras [94] with Tensorflow [95] back-end. Once trained, the six output channels of the network are combined to obtain a grid with a range of values which represent the likeliness of hydration.

Neural networks for point-wise prediction using spherical harmonics expansion

In the second approach, the hydration information of each individual point is predicted based on the input descriptors specifying water-protein interactions at this location and the environment of this water location. The approach consists of two subsequent models,

a classifier to separate grid point with water occupancy from those without, and a second regression model only for grid points classified as "with occupancy" in the first model. In this regression model occupancy values and free energies of desolvation are computed. In classification and regression model, parameters for the protein atoms such as van der Waals radius and partial charge are directly taken from the coordinate and topology file prepared for WATsite simulations.

Classification model to identify grid points with water occupancy

For each grid point, the spatial environment and flexibility of surrounding atoms is computed. In detail, the distance from grid point k to all atoms i in the neighborhood of the grid point are computed and the van der Waals radius of the protein atom σ_{i} is subtracted:

$$\widetilde{r}_{\mathrm{i}k} = |R_{\mathrm{i}} - r_k| - \sigma_{\mathrm{i}}. \tag{2.1}$$

All $\widetilde{r}_{\mathrm{i}k}$ values up to 6 Å are distributed onto a continuous 25-dimensional vector using the Gaussian distribution function, where the value at bin i is

$$p_{k,\mathrm{i}} = \exp\left(-\left(\widetilde{r}_{\mathrm{i}k} - \left(\mathrm{i}\cdot w - 1\ \text{Å}\right)\right)^2/(2\cdot w^2)\right) \tag{2.2}$$

with $w = 7\ \text{Å}/25$. All values are finally scaled using $\tanh\left(p_{k,\mathrm{i}}/5\right)$ to limit values to the range [0;1].

Separate vectors are computed in the same manner for hydrogen-bond donor and acceptor atoms. The motivation for this additional descriptors are that shorter distances between water and hydrogen-bonding groups are observed compared to hydrophobic contacts.

Despite the applied harmonic restraints, dynamic fluctuations of the protein atoms are observed throughout the WATsite MD simulations. These fluctuations can have impact on the accessibility of water molecules to different locations in the binding site. To incorporate those atomic fluctuations in the neural network predictions of occupancy, we designed a simple flexibility descriptor for the side-chain atoms (backbone atoms are considered rigid in this analysis). The shortest topological distance t_{i} of a side-chain atom i to the corresponding

C_α atom is translated using $f_\mathrm{i} = 2 \cdot \tanh(t_\mathrm{i}/4)$. The distance between this atom and grid point k is then distributed to an additional 25-dimensional vector using a modified Gaussian distribution

$$q_{k,\mathrm{i}} = f_\mathrm{i} \cdot \exp\left(-\left(\widetilde{r}_{\mathrm{i}k} - \left(\mathrm{i} \cdot w - 1\ \text{Å}\right)\right)^2 / (2 \cdot w^2)\right) \tag{2.3}$$

Subtracting this vector $q_{k,\mathrm{i}}$ from the unmodified vector $p_{k,\mathrm{i}}$ generates a vector that measures the flexibility of the environmental atoms around grid point k.

All four vectors are concatenated which generates a 100-dimensional input vector to the neural network for classification.

In addition to the input layer, the neural network architecture consists of a fully-connected hidden layer with 1024 nodes with leaky-ReLU activation and dropout layer with dropout probability of 0.5, followed by a second fully-connected hidden layer with 512 nodes with leaky-ReLU activation and a final output layer with sigmoid activation to classify each grid point as either occupied (1) or unoccupied (0). A threshold occupancy value of 10^{-5} in the input was used to separate occupied from unoccupied grid points.

Adam optimizer [6] with learning rate of 0.001 and a batch size of 250 was used to train the model. Learning was performed for 50 epochs using Tensorflow [95].

Regression model

For each grid point, first the direct interactions between water probe and protein atoms is computed. In detail, electrostatic fields of the protein atoms i at location R_i with partial charge Q_i are computed on each grid point r_k

$$E_k^{elst} = \sum_\mathrm{i} \frac{Q_\mathrm{i}}{|R_\mathrm{i} - r_k|}. \tag{2.4}$$

Steric contacts of water probe with protein atoms i at location R_i with van der Waals radius σ_i and well-depth ϵ_i is computed using a soft alternative of the van der Waals equation

$$E_k^{sterics} = \sum_\mathrm{i} \sqrt{\epsilon_\mathrm{i}\epsilon_p} \left(\left(\frac{\sigma_{\mathrm{i}p}}{|R_\mathrm{i} - r_k|} \right)^4 - \left(\frac{\sigma_{\mathrm{i}p}}{|R_\mathrm{i} - r_k|} \right)^2 \right). \tag{2.5}$$

with $\sigma_{ip} = \sigma_i + \sigma_p$ (probe $\sigma_p = 1.6$ Å) and well-depth of probe $\epsilon_p = 0.012$ kcal mol^{-1}. Protein parameters from the Amber14 force field are used.

Hydrophobic contacts are computed [96] using

$$E_k^{hphob} = \sum_i \begin{cases} 1 & \text{if } s \leq -1 \\ 0.25 \cdot s^3 - 0.75 \cdot s + 0.5 & \text{if } -1 < s < 1 \\ 0 & \text{if } 1 \leq s. \end{cases} \tag{2.6}$$

with

$$s = 2.0 \cdot (|R_i - r_k| - \sigma_{ip} - 2.0) / 3.0. \tag{2.7}$$

Hydrogen-bond interactions between water probe and protein acceptor/donor heavy atoms i are computed using

$$E_k^{HBond-Acc} = \sum_i \exp\left(-|R_i - r_k - R^0|^2\right) \tag{2.8}$$

and

$$E_k^{HBond-Don} = \sum_i \begin{cases} -\exp\left(-|R_i - r_k - R^0|^2\right) \cdot \cos\left(\alpha_{iHk}\right) & \text{if } \cos\left(\alpha_{iHk}\right) < 0 \\ 0 & \text{if } \cos\left(\alpha_{iHk}\right) \geq 0 \end{cases} \tag{2.9}$$

respectively ($R^0 = 1.94$ Å).

Each interaction term is then scaled and transformed by a hyperbolic tangent function to the range $[0; 1]$

$$\widetilde{E}_k^{property} = \tanh(E_k^{property}) \tag{2.10}$$

with the exception of the electrostatic interaction term which is scaled to be within $[-1; 1]$ (small negative van der Waals interaction values are clipped off at zero). Each scaled

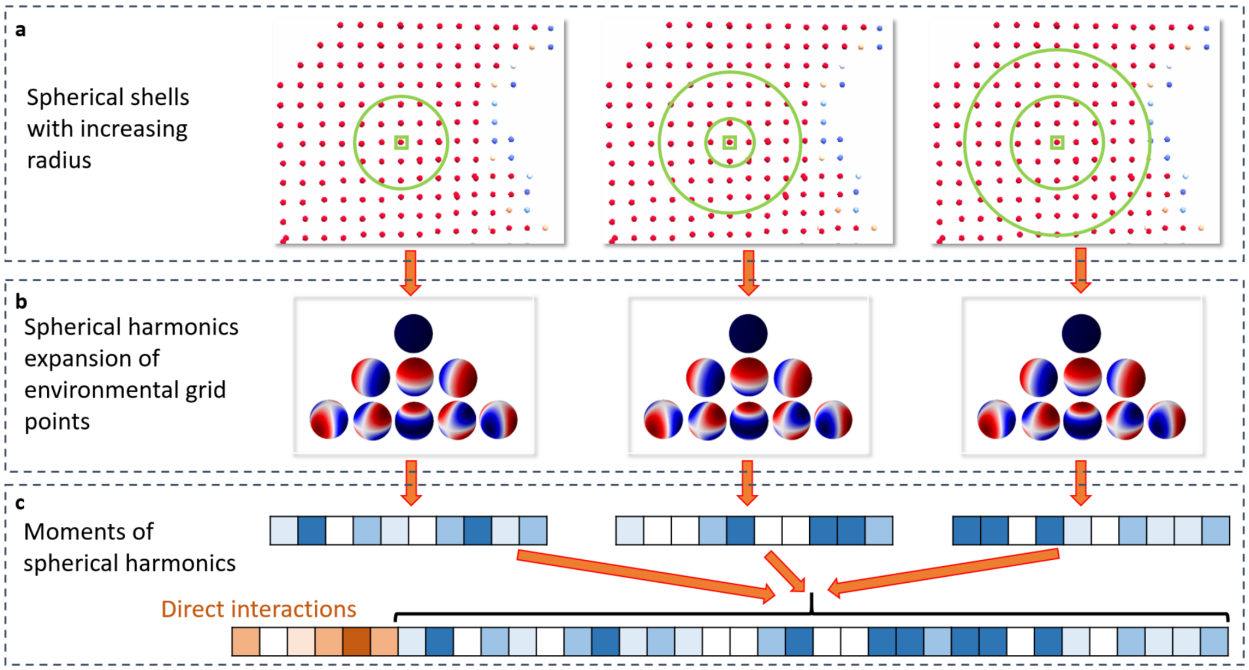


Figure 2.6. Input of neural network. Generation of input vector for neural network for point-wise prediction of hydration data. (a) For each grid point, the interaction fields from the protein are computed. Nearby grid points within a spherical shell around the grid point are identified. (b) The interaction field distribution of those grid points are represented by spherical harmonics expansion. (c) The moments of this expansion generate an environment vector. (d) The environment vectors of spherical shells with increasing radius are concatenated together with the direct interaction fields at this grid point. This final vector is used as input for the neural network.

interaction term is finally transformed into a continuous vector of size 20 using Gaussian distribution functions, where the value at each bin i is determined by

$$p_{k,\mathrm{i}}^{property} = \exp\left(-\left(\widetilde{E}_k^{property} - (\mathrm{i}\cdot w + \min\left(\widetilde{E}^{property}\right)\right)^2/(2\cdot w^2)\right) \tag{2.11}$$

(bin width of $w = 2/20$ and $w = 1/20$ for electrostatic interactions and all other interactions, respectively). The five 20-dimensional vectors are concatenated to generate a 100-dimensional input vector to the neural network.

The stability of water molecules not only depends on the protein environment but also on the surrounding network of additional water molecules. Thus, the environment of the

water probe needs to be quantified as well. Here, we use a spherical harmonics expansion of the interaction fields on surrounding grid point as additional descriptors. In detail, seven spherical shells with increasing radius are defined to identify neighboring grid points with increasing distance to probe location: $[-\epsilon;\ 1\ \text{Å}+\ \epsilon]$, $[0.5\ \text{Å}-\epsilon;\ 1.5\ \text{Å}+\ \epsilon]$, ..., $[3\ \text{Å}-\epsilon;\ 4\ \text{Å}+\ \epsilon]$ (ϵ is small value to include grid points with distance at the boundary of interval) (Figure 2.6). The grid points in each shell are projected onto a unit sphere and the interaction values of those grid points are used to compute the coefficient of the spherical harmonics up to a certain order l_{max}:

$$\widetilde{E}^{property}_{\text{neighbors of } k}(\theta,\phi) \approx \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} a_l^m Y_l^m(\theta,\phi) \tag{2.12}$$

The sum over the degrees of the L2-norm of the coefficients

$$\widetilde{a}_l = \sum_{m=-l}^{l} ||a_l^m|| \tag{2.13}$$

is computed, transformed using $\tanh(\widetilde{a}_l)$ and distributed onto continuous 5-dimensional vectors by a Gaussian distribution function (Equation 2.11). The vectors of direct interactions (Equation 2.11) are finally concatenated with the different coefficient vectors for the different l and different interaction types to generate the final input vector to the neural network.

The neural network architecture consists in addition to the input layer a fully-connected hidden layer with 2048 nodes with leaky-ReLU activation and dropout layer with dropout probability of 0.5, followed by a second fully-connected hidden layer with 1024 nodes with leaky-ReLU activation and a final output layer with occupancy and free energy values.

Adam optimizer [6] with learning rate of 0.001 and a batch size of 250 was used for training the model. Learning was performed for 125 epochs using Tensorflow [95].

2.2.3 Hydration site prediction

Clustering of occupancy grids to identify hydration sites

To compare hydration occupancy predictions with crystallographic water data and other hydration site prediction methods, occupancy grids obtained from the two neural network methods were clustered to predict hydration sites. Two different clustering method were

selected for this purpose. For the Inception+U-Net model, a modified DBSCAN clustering methods was utilized (see Supplementary Algorithm 1). For the point-wise prediction model using spherical harmonics, quality threshold (QT) clustering algorithm was used with the following parameters: Maximum cluster diameter: 1.9 Å; minimum number of grid points in a cluster: 5.

Evaluation of prediction performance: Comparison with experimental data and other hydration site prediction methods

To evaluate and compare the ability of our methods to reproduce water locations in X-ray data, we chose four apo systems from data from Rudling et al. [97]: Acetylcholinesterase, heat shock protein 90-alpha, trypsin I and fatty acid binding protein adipocyte with PDB-ids 1ea5, 1uyl, 1s0q, and 3q6l. All four systems are not part of our training set. The binding site center was defined by superposing the holo form of the same proteins (with ligand present) onto the apo form and using the centroid of the aligned ligand as the center of the grids. We compared the performance of our method with two other methods: WATsite [59] (MD-based method) and hydration site prediction generated from GAsol's clustering method on 3D-RISM grids [75] (grid-based method). All crystallographic water molecules and ions were removed as part of the protein preparation process. The proteins were prepared automatically by the scripts available in the WATsite 3.0 package for 3D-RISM and WATsite. Both methods were run using their default parameters. The spatial deviation of predicted hydration sites from crystallographic water locations observed in the PDB files was measured. The distance of each crystallographic water molecule to the closest predicted hydration site was measured. Only X-ray water molecules within 5 Å of any ligand and protein atom were considered.

2.3 Results

2.3.1 Neural network for semantic segmentation

Performance in prediction of water occupancy grids

To incorporate the context of a grid point in the neural network, we utilized CNNs based on the computed MIFs. This approach predicts the water occupancy on a grid point by incorporating spatial context from surrounding grid points during the convolutional feature abstraction process. The CNN network architecture (Supplementary Figure 3) down-samples the input layer identifying features important for the prediction of water occupancy. The final layers up-sample the grid to the desired occupancy grid. Similar architectures have been used for many applications such as semantic segmentation and generative models. More specifically, we use U-Net as the network architecture. U-Nets are commonly used for semantic segmentation tasks. For image segmentation tasks, a U-Net can rapidly learn to pass critical information such as the outlines of an object, which is similar between input and output layers. This process makes the learning more efficient. Similarly, for the task of water prediction, the surface of the protein is quickly captured by the U-Net from the input data. Our tests showed that without skip connections, it would be difficult for the network to capture the protein surface, or the solvent accessible surface with the same efficiency.

Initially, we attempted to generate regression models that aimed to predict the actual occupancy value of each pixel or grid point. The resulting models showed poor prediction performance, which can be largely attributed to the highly imbalanced nature of the water grids, i.e. most grid points in a water grid have low or zero occupancy. Alternatively, the water prediction task using 3D CNNs can be tackled as a segmentation problem, detecting dense areas where water is more likely to have high occupancy. We have formulated the problem of predicting water occupancy as a multi-class segmentation problem allowing to identify regions with different levels of water occupancy, here predicting occupancy levels with threshold values of 0, 0.02, 0.03, 0.045, 0.06 and 0.07 (see Supplementary Methods section for details on calculation of occupancy values). The threshold of >0 classifies regions that are generally accessible to water molecules. The threshold of 0.02 represents approximately bulk water density. Occupancy values above this threshold represent regions with

increased water density (= hydration sites). Most hydration sites are formed by densities with values between 0.045 to 0.06. Values above 0.07 are rather rare.

To evaluate the neural network's performance, 5-fold cross-validation was used. The set of proteins was first divided into five groups (Supplementary Data 1). Then, the network was trained on four groups and tested on the one group left out, generating a set of five models. Given the similarity among the proteins in the refined set, we chose not to use random assignment to the five groups. For proper validation of the procedure, we instead minimized the similarity among the different groups by clustering the whole set of proteins based on binding site similarity. This guarantees that during cross-validation, the test set is always the least similar to the training set. To equalize the size of the clusters, samples were removed from larger clusters, resulting in 223 protein systems contained in each cluster. The similarity was calculated using the FuzCav program [98] and the structures were clustered using the k-modes clustering algorithm [99], [100] on the feature vector generated by FuzCav. For the purpose of data augmentation, the training samples were rotated randomly on-the-fly along the coordinate axes.

Figure 2.7 shows visualization of the predicted water occupancy for two example proteins at different isovalues representing different thresholds of occupancy. At low thresholds, the quality of predicting occupancies is excellent; predicted and reference occupancy grids largely overlap. As the threshold is increased, the prediction quality drops due to the sparsity of the grid points with high occupancy, demonstrating that even with generalized form of the Dice loss (GDL; see Methods for details)[91] the problem of imbalance in the data set was not completely resolved. We further observed that the network fails to correctly predict the regions close to the boundaries of the grid. A possible explanation for this problem is that for these grid points the network does not receive the full context (MIFs of surrounding grid points) as those neighboring grid points would lie beyond the boundary of the grid box. This failure to correctly predict the occupancy of boundary grid point, however, does not create a serious issue for the purpose of predicting hydration information in the binding site, as the grid points on the boundary of the box lie outside of the binding pocket volume. A mitigation for this problem is to remove the prediction in the boundary regions of the grid box after model generation. Therefore, we focused our analysis on the relevant region in the

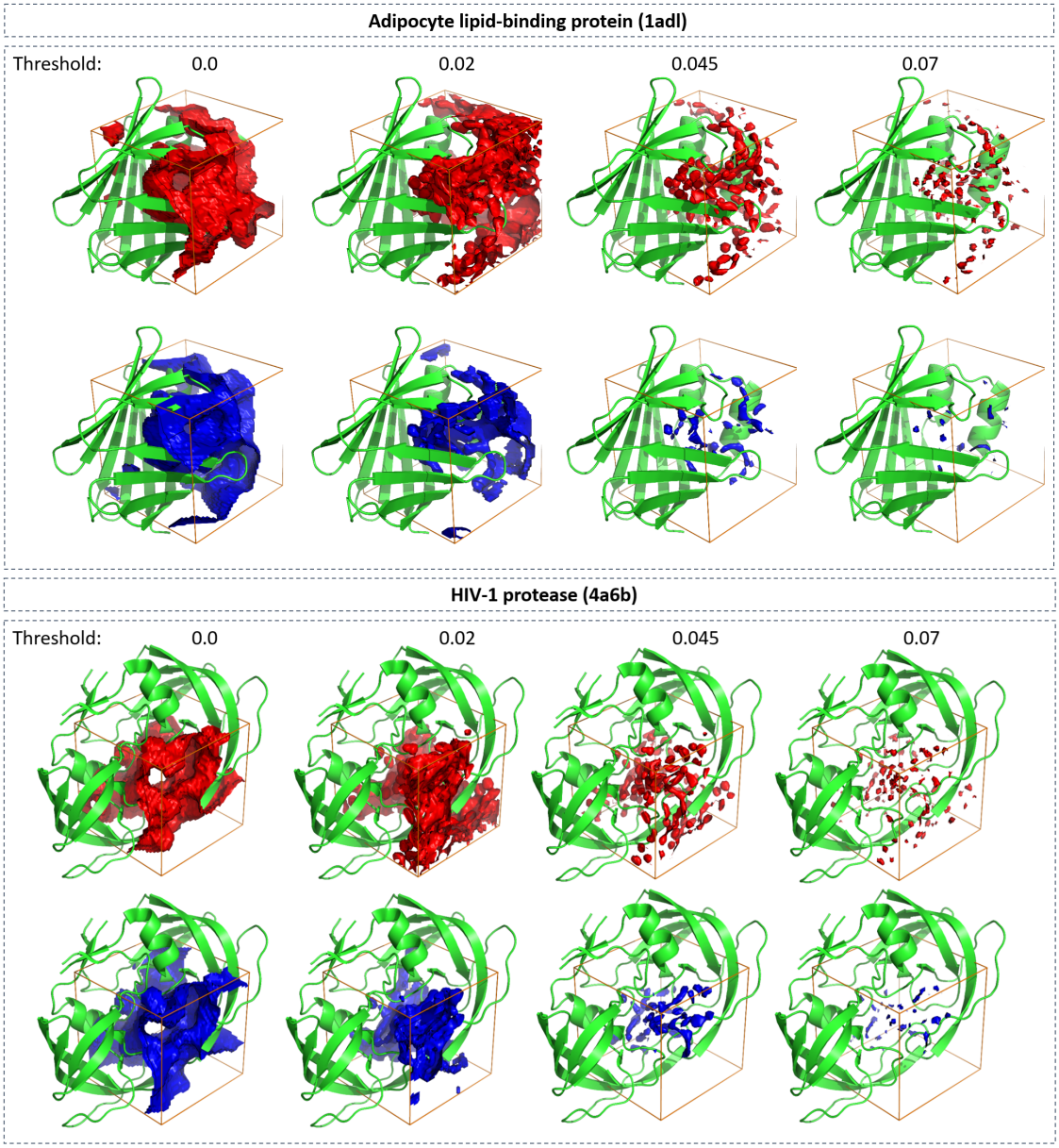


Figure 2.7. Accuracy of U-Net method. Visual comparison between ground truth (red) and neural-network predicted (blue) water occupancy for adipocyte lipid-binding protein (PDB-code: 1adl) and HIV-1 protease (4a6b). Predictions were performed using U-Net. Isosurfaces at four different threshold values (0.0, 0.02, 0.045, and 0.07) are shown. The task of predicting areas with higher occupancy becomes challenging for the network due to the sparsity of those points (at thresholds 0.045 and 0.07). The regions closer to the corners of the grid are more difficult to predict as information of the context of those grid points is missing.

Table 2.1. Performance of different U-Net architectures. Various metrics for the performance of a baseline U-Net and a U-Net using Inception and Residual blocks. Performance on the validation sets are displayed (shown as mean ± standard deviation of cross-validation trials). Metrics are shown for the grids covering the whole binding site and for the sub-grids focusing on the area within 5 Å of the ligand center. The results show that the Inception+Residual U-Net surpasses the baseline model's performance.

Network	Distance from Ligand	Generalized Dice Loss	Dice overlap (smoothed)
Baseline U-Net	Full grid	0.44 ± 0.08	0.40 ± 0.20
	<5 Å from center	0.35 ± 0.06	0.51 ± 0.17
Inception+ Residual U-Net	Full grid	0.29 ± 0.04	0.79 ± 0.04
	<5 Å from center	0.24 ± 0.02	0.84 ± 0.02

Table 2.2. Precision and recall of convolutional neural network. Precision and recall values for prediction of WATsite occupancy using fully convolutional neural network at five different levels of occupancy threshold values.

Occupancy threshold	Precision	Recall
0.02	0.86 ± 0.03	0.87 ± 0.03
0.03	0.79 ± 0.02	0.81 ± 0.03
0.045	0.73 ± 0.01	0.62 ± 0.03
0.06	0.72 ± 0.01	0.56 ± 0.01
0.07	0.70 ± 0.01	0.54 ± 0.00

vicinity of the bound ligand, i.e. all grid points with a maximum distance of 5 Å around the co-crystalized ligand.

Tables 2.1 and 2.2 show different metrics for the prediction quality of the model obtained from the cross-validation. Only data for the left-out systems are used in the statistical analysis. In Table 2.1 we used smoothed Dice overlap [92] to measure the overlap between the reference and the predicted grids. In this metric the confidence of prediction of a label is included. For each metric both the quality of the full grid and for the area within 5 Å from the ligand is displayed. Table 2.2 displays precision and recall values for the water occupancy in the area within 5 Å from the ligand.

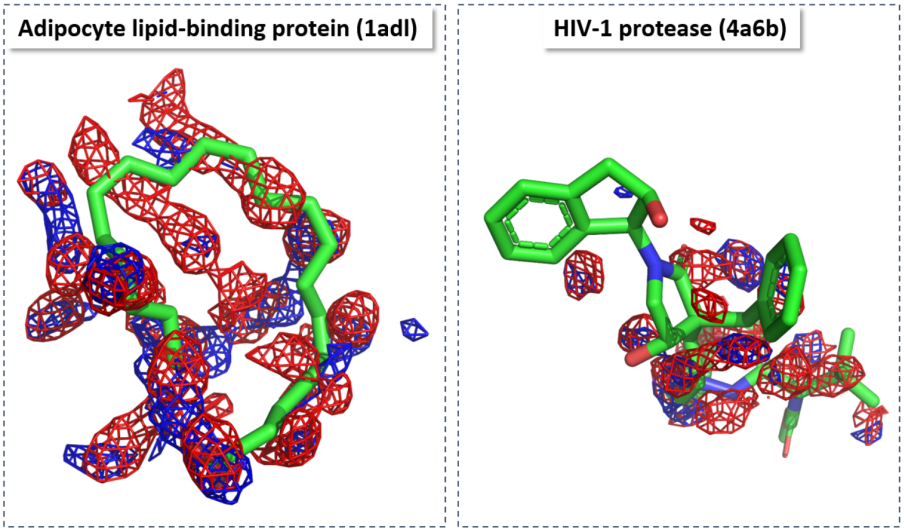


Figure 2.8. Accuracy of U-Net method focused on binding site. Visual comparison between group truth (red) and neural-network predicted (blue) water occupancy for adipocyte lipid-binding protein (PDB-code: 1adl) and HIV-1 protease (4a6b) within 5 Å of the co-crystallized ligand. Note that the ligands were not included either in the water simulations to produce the ground truth or in the generation of input MIF grids. They were added for visualization purpose only. Predictions were performed using U-net. Isosurfaces at a threshold value of 0.045 are shown.

Figure 2.8 shows an overlay of reference and predicted water occupancies within 5 Å of the co-crystalized ligand to demonstrate the prediction quality in the proximity of the ligand. For applications of the model to drug design, we are interested in this particular region to identify how hydration might enhance, diminish or interfere with ligand binding at the binding site.

Importance of probes

We further analyzed which input MIF grids contributed most to the prediction performance. To compute the feature importance we used the Mean Decrease Accuracy (MDA) or permutation importance method [101]. This method measures how the absence of a feature decreases the performance of a trained estimator. This method can be directly applied to the validation set without the need of retraining for each feature removal. A feature is replaced with random noise with the same distribution as the original input. One simple way is to shuffle the values of a grid randomly, so that it no longer contains useful information. As expected, the probes which are most influential for the prediction quality were either water probes (OH2) or probes which mediate hydrogen bonding. It should be noted that although water probes from Flap are designed to indicate the water affine areas, they do not linearly correlate with WATsite occupancy, namely, the Pearson correlation coefficient between those MIFs and WATsite occupancy is close to zero. Table 2.3 shows the performance drop with shuffling of each input grid on the validation sets (sorted by importance of probe).

2.3.2 Neural networks for point-wise prediction using spherical harmonics expansion

Classification model

In contrast to the segmentation model, in the point-wise model each individual grid point represents a sample that can be used for training and testing of the model. Thus, the size of the data set is significantly increased and allows to design a more aggressive testing protocol compared to the segmentation method. For the point-wise prediction, the same

Table 2.3. Importance of probe grids. Dice overlap value for the cross-validation sets after shuffling of grid point value for each of the 12 MIF grids. The larger the change in value, the more important the probe grid is for the prediction. Important probe grids are displayed in bold. The un-shuffled dice overlap values are shown in Table 1 for all grid points and grid points around ligand.

Probe	Dice overlap	Dice overlap (<5 Å from ligand)
C1=	0.56 ± 0.06	0.58 ± 0.04
OH2	0.51 ± 0.04	0.56 ± 0.03
CRY	0.62 ± 0.04	0.67 ± 0.04
I	0.63 ± 0.12	0.64 ± 0.09
O-	0.71 ± 0.04	0.73 ± 0.02
DRY	0.71 ± 0.07	0.79 ± 0.05
N+	0.77 ± 0.04	0.83 ± 0.02
H	0.75 ± 0.06	0.79 ± 0.02
F3	0.78 ± 0.05	0.83 ± 0.03
OC2	0.79 ± 0.05	0.84 ± 0.02
I-H	0.78 ± 0.05	0.81 ± 0.02
NA+	0.75 ± 0.03	0.81 ± 0.03

5-fold splitting procedure of the data set was used. In contrast to the segmentation model, only one-fifth was used for training and four-fifth for testing.

For the classification model, i.e. separating grid points between those with and without water occupancy, the normalized confusion matrix over the test set was computed (Figure 2.9). 94% of occupied grid points and 96% of unoccupied grid points were correctly classified. The precision values of 0.97/0.92 and recall values of 0.96/0.94 for occupied/unoccupied data signifies the accuracy of the classification model in identifying moieties in the binding site that have been observed to be occupied by water molecules throughout WATsite simulations.

Regression model

Whereas the classification model allows to identify regions with likely water occupancy with high accuracy, a rather small occupancy threshold of 10^{-5} was used. In practice it is desirable to identify regions in the binding site with high water densities and occupancy peaks that resemble hydration sites. Therefore, a regression model was designed to identify those high density among low density regions. Using descriptors encoding only the direct

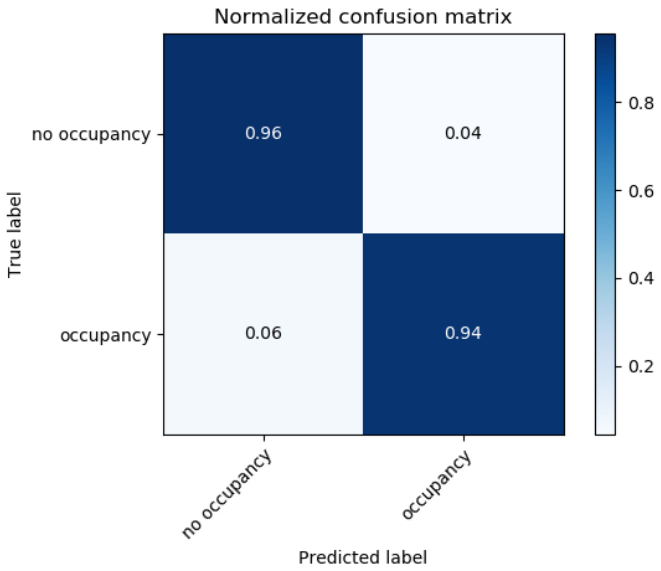


Figure 2.9. Confusion matrix for classification model. Normalized confusion matrix for classifying grid points with and without water occupancy using neural network model.

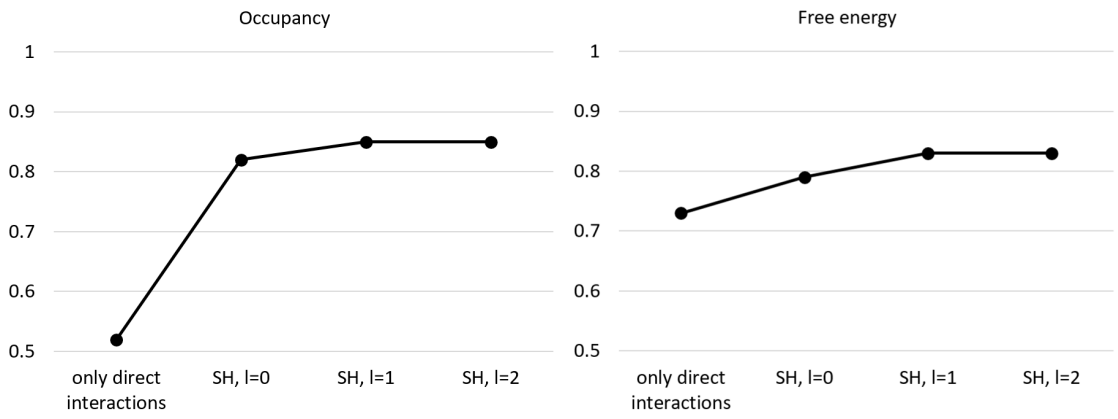


Figure 2.10. Accuracy of regression model. Regression coefficient r for correlating occupancy and free energy values of neural network predictions with original WATsite data.

interactions with the protein at the specific grid point location (no inclusion of nearby grid points), a mediocre correlation between predicted and ground truth water occupancy was identified ($r = 0.52$) (Figure 2.10). Using only the radial distribution of interaction profiles of nearby grid points (l=0) increases the regression coefficient to $r = 0.82$. Increasing the depth of the spherical harmonics (l=1) only slightly increases the regression coefficient further to $r = 0.85$. Further addition of angular functions to represent the environmental grid points (l=2) does not further improve the regression between ground truth and predicted occupancy values. Consequently, we used the regression model with l=1 for subsequent analysis (see below).

The same trend, although weaker in magnitude, was observed in the regression outcome for the free energy of desolvation at the grid points with occupancy. A maximum r value of 0.83 was achieved.

For further evaluation of the neural network performance, 5-fold cross-validation was used. Again, only a fifth of the data set was used for training in each cross-validation step and four-fifth were used for testing the model. All five models exhibited very similar test set performance. For occupancy the r values ranged between 0.85 and 0.86 (standard deviation of 0.004), for free energy it ranged between 0.83 and 0.84 (standard deviation of 0.0044).

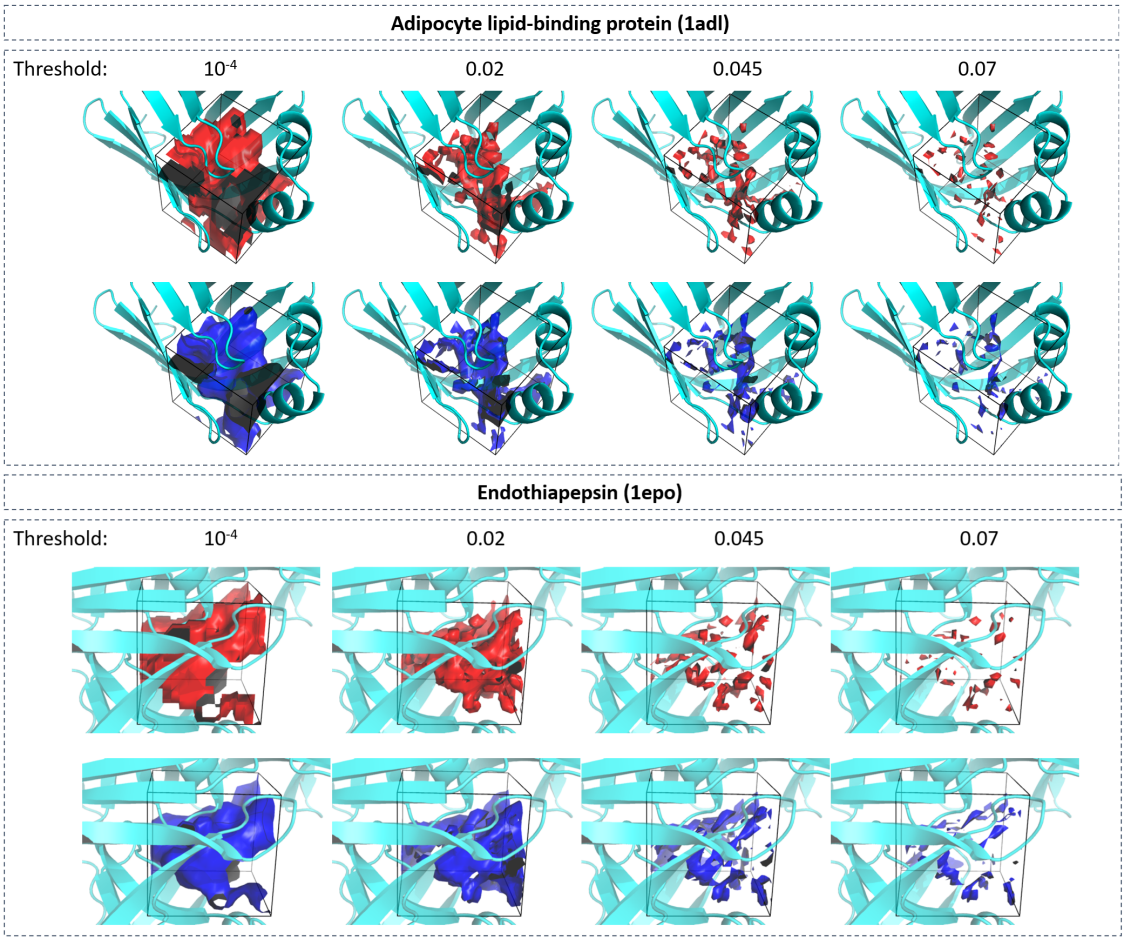


Figure 2.11. Accuracy of regression model. Visual comparison between group truth (red) and neural-network predicted (blue) water occupancy for adipocyte lipid-binding protein (PDB-code: 1adl) and endothiapepsin (1epo). Predictions were performed using regression neural network. Isosurfaces at four different occupancy values (10^{-4}, 0.02, 0.045, and 0.07) are shown.

This highlights the robustness of the model, independent of the specific protein systems used for training.

Figure 2.11 shows the comparison of predicted and ground truth water occupancy at isolevels of 10^{-4}, 0.02, 0.045 and 0.07 for two different protein systems. Excellent overlap between predicted water occupancy and ground truth was observed with slight deterioration in accuracy for the highest density maps at 0.07. This visual observation can be quantified by measuring the precision and recall values at different classification threshold values of 0.02, 0.03, 0.045, 0.06 and 0.07 (Table 2.4). Relatively unchanged precision and recall values

were observed up to an occupancy threshold of 0.045. Lower accuracy was observed for occupancy values of 0.06 and 0.07. This observation is consistent with previously discussed imbalance between large number of low-occupancy and small number of high-occupancy grid points.

Table 2.4. Precision and recall of regression neural network. Precision and recall values for prediction of WATsite occupancy using regression neural network at five different levels of occupancy threshold values.

Occupancy threshold	Precision	Recall
0.02	0.79 ± 0.03	0.79 ± 0.06
0.03	0.79 ± 0.04	0.77 ± 0.06
0.045	0.78 ± 0.04	0.76 ± 0.06
0.06	0.75 ± 0.04	0.66 ± 0.06
0.07	0.75 ± 0.04	0.66 ± 0.06

Similar trends were observed for the prediction of free energy values (Figure 2.12). Here infrequent negative desolvation values were less accurately predicted compared to positive values. Even regions containing high positive desolvation values were predicted with relatively high quality.

2.3.3 Comparison with other machine learning approaches

Failure of machine learning based on protein density descriptors

Protein densities distributed on a 3D grid have been used as input descriptors for docking applications [13]. Here, we tested if a similar approach could be used to predict hydration information in the binding site. In detail, an atom is distributed on a 3D grid according to its atom type using a Gaussian distribution function centered on the atom center. Using this Gaussian smearing reduces the sparsity of the input data which would result in poor learning in neural networks since the gradients propagated throughout the network will be sparse as well [102]. Furthermore, Gaussian smearing better represents the spatial extension of the protein and therefore local accessibility of water to the protein surface.

Whereas these input data show good performance for binding pose prediction of chemicals binding to proteins [13], no significant learning was observed in the context of water

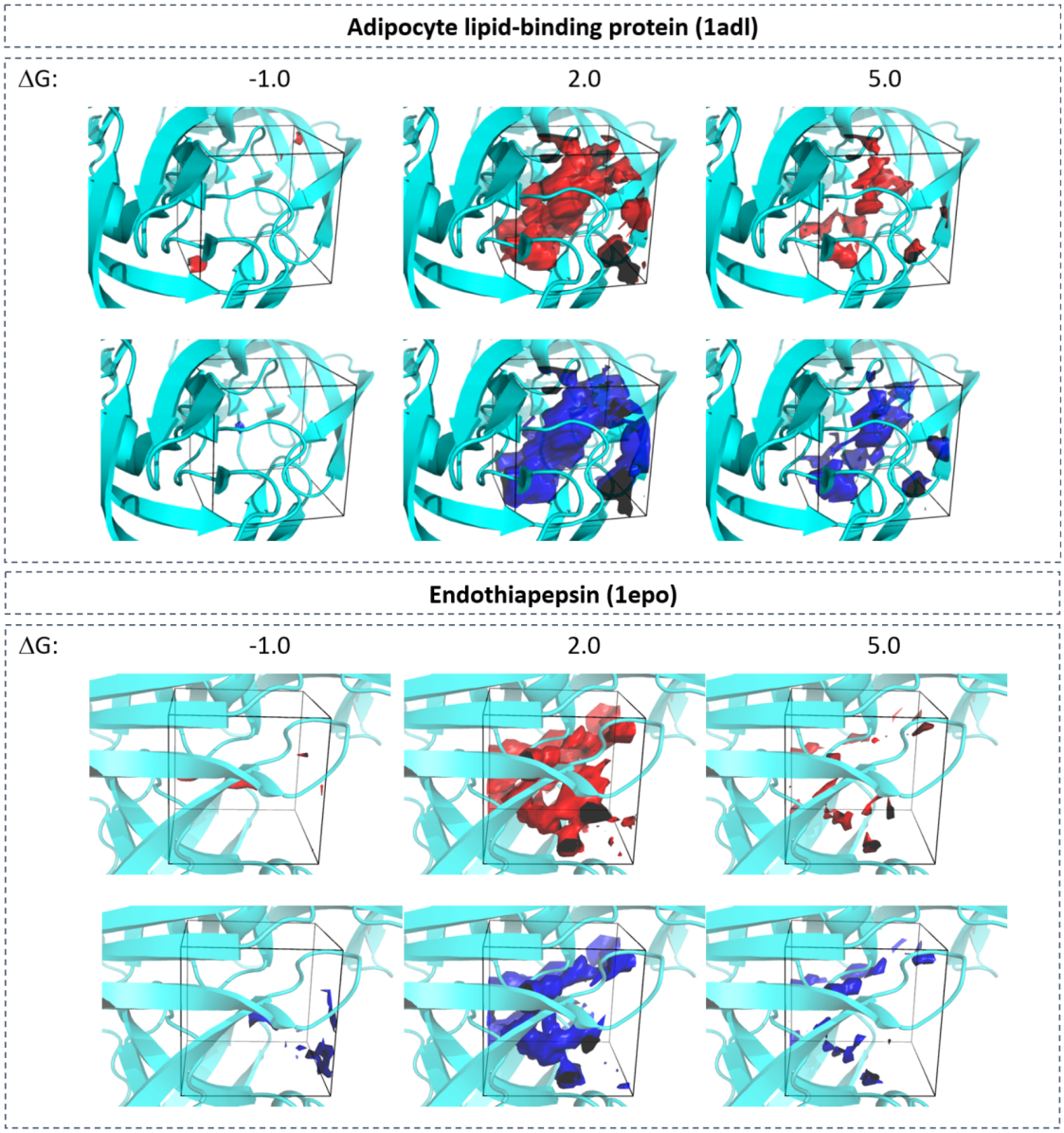


Figure 2.12. Accuracy of regression model. Visual comparison between group truth (red) and neural-network predicted (blue) desolvation free energy for adipocyte lipid-binding protein (PDB-code: 1adl) and endothiapepsin (1epo). Predictions were performed using regression neural network. Isosurfaces at three different free energy values (-1 kcal mol^{-1}, 2 kcal mol^{-1}, and 5 kcal mol^{-1}) are shown.

occupancy prediction (data not shown). This failure can be interpreted by the lack of modeling of long-range protein-water interactions and water-water interactions. CNNs based on protein density would allow modeling of local correlation between protein shape/properties and adjacent water occupancy. The stability of water molecules in protein binding sites, however, is strongly influenced by long-range electrostatic interactions and by the formation of hydrogen-bonding water networks [103], [104]. Both contributions are difficult to model using localized features extracted by the layers of the CNN.

Failure of point-to-point correlations using MIFs

In an another alternative approach, we represented the protein indirectly using molecular interaction fields (MIFs) data [105]. MIFs were generated as described previously. As described in Methods:Probe selection, 12 probes were selected to generate 12 different channels for the input layer. Neural networks were designed for simple point-to-point correlations, where the different MIF input channels were correlated with WATsite occupancy. In our tests, however, neural networks or other machine learning algorithms were unsuccessful in finding any significant point-to-point correlations. From this observation, we concluded that even the MIFs generated with a water probe differ significantly from the WATsite predictions. This can be explained by the fact that the MIFs only represent direct protein-probe interactions and therefore lack the incorporation of water-water interactions. Thus, the interaction value with a probe at a given point does not provide enough information for a network to infer water occupancy. For example, a grid point in an occluded space buried deep inside a protein may have a similar interaction profile with the protein in context of the MIFs to another grid point in a solvent exposed area. The former point, however, may have lower occupancy due to the lack of stabilizing water-water interactions.

WATsite in contrast includes water-water network interactions explicitly. Furthermore, it explicitly includes entropic contributions, as the water distribution is sampled from a canonical statistical ensemble during the MD simulation. To predict water occupancy at a certain location, the neural network requires not only the interaction information on the corresponding grid point, but also the context of the grid point, i.e. interaction with other

water-molecules. Those interactions can be represented either by directly including information of neighboring grid points or by the explicit design of input descriptors that include environmental information. The latter approach was described in the section "Neural networks for point-wise prediction using spherical harmonics expansion", the former was discussed in the section "Neural network for semantic segmentation".

2.3.4 Applications

The two NN approches for the generation of hydration information were applied to three different topics, i.e. the prediction of hydration site locations in X-ray structures, the qualitative and quantitative analysis of structure-activity relationships (SAR) data, and the improvement of CNN-based pose ranking in docking applications.

Prediction of hydration site locations

In the first application, we tested the potential of both NN approaches to reproduce the position of crystallographic water molecules in the binding site of four protein systems: Acetylcholinesterase (1ea5), heat shock protein 90-alpha (1uyl), trypsin I (1s0q) and fatty acid binding protein adipocyte (3q6l) (Figure 2.13). Both of our methods were compared to WATsite [59] and GAsol (3D-RISM) [75]. It should be noted that WATsite had been previously tested to reproduce X-ray water molecules [59], [61], [68]. We show the prediction performance of finding hydration sites within 1.0 Å, 1.5 Å and 2.0 Å distance to the corresponding X-ray water location. Hydration sites with distances greater than 2 Å to the corresponding X-ray water locations are considered as failed predictions. WATsite is the most accurate of all methods (Figure 2.13), in particular considering small spatial deviations. Both neural networks-based methods either perform equally well or better than GAsol (3D-RISM) and approximate WATsite performance for most systems at a deviation of 1.5 Å or 2 Å.

It should be noted that a comparison between X-ray water molecules and hydration sites has overall its limitations: First, fit of water positions into electron density obtained from X-ray experiments is not free of errors. Second, X-ray structures are typically resolved at low

temperatures underestimating entropic effects. Third, crystal effects may have an influence on water networks, in particular if the binding site is partially or fully solvent exposed. Fourth, the identified hydration sites depend on cluster algorithm and settings, thus adding additional inaccuracies to the grid-based prediction of hydration density. In light of those arguments, we believe the hydration site predictions using both NN are reasonably accurate, considering their significantly higher efficiency compared to running MD simulations.

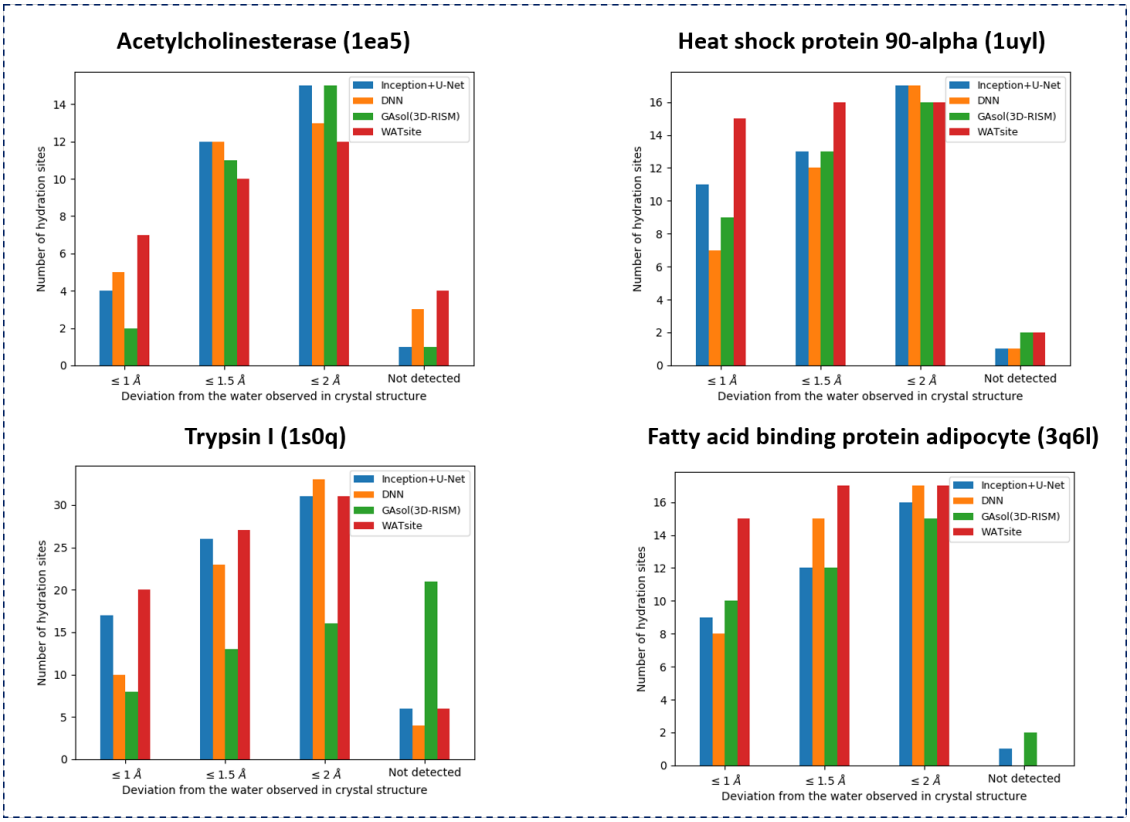


Figure 2.13. Reproducing hydration sites observed in X-ray crystal structures. Comparison among Inception+U-Net, deep neural network (DNN) based on spherical-harmonics expansion, GAsol/3D-RISM and WATsite. "Not detected" means no hydration site within 2 Å of X-ray water molecule.

Structure-activity relationships guided by hydration analysis

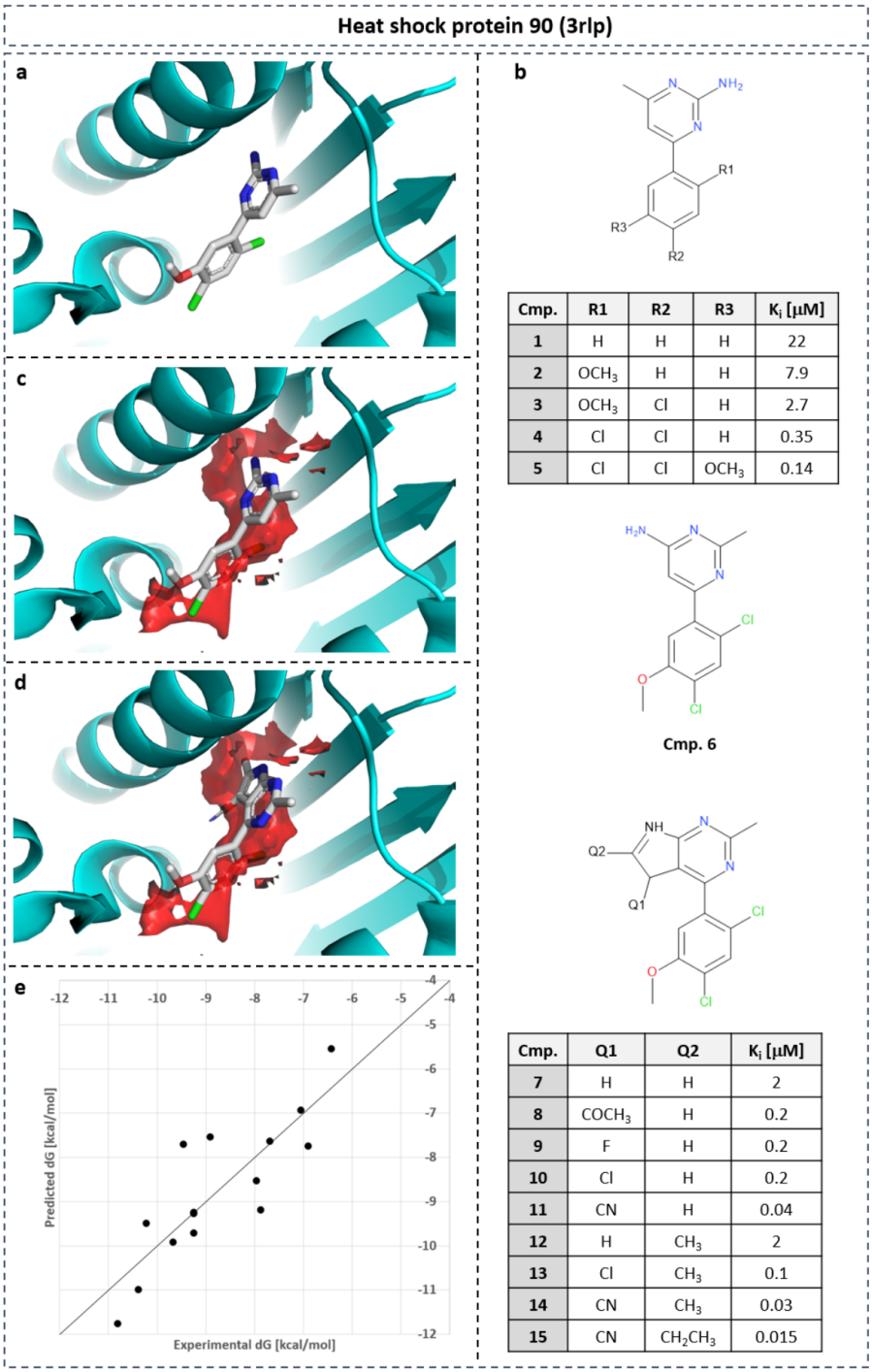
Hydration site prediction using MD-based methods such as WaterMAP or WATsite have been utilized in many recent medicinal chemistry projects to understand ligand binding

and structure-activity relationships(SAR), as well as for the guidance of lead optimization. Recently, Bucher et al. demonstrated the superiority of simulation-based water prediction using WaterMAP compared to other commercial methods SZMAP, WaterFLAP and 3D-RISM [62] for the analysis of the structure-activity relationships of lead series of different target systems. To demonstrate that the instantaneous prediction of thermodynamic hydration information based on our neural networks can be used with similar confidence in lead optimization projects, we performed three retrospective SAR analyses on heat shock protein 90 (HSP90), beta-secretase 1 (BACE-1) and major urinary protein (MUP).

In a study of Kung et al.[106], a series of HSP90 inhibitors were synthesized and tested (Figure 2.14). The design of the molecules was guided by replacing water molecules resolved in the X-ray structure of HSP90. We performed hydration profiling on the X-ray structure 3rlp of HSP90 with the co-crystallized ligand removed using the point-wise neural network model. Water density with high positive (unfavorable) desolvation free energy (Figure 2.14c, red surface, isolevel for ΔG=7.5 kcal mol^{-1}) is located around the phenyl ring of compound A (Figure 2.14b). Subsequent substitution of hydrophobic groups on the phenyl ring at positions R1, R2 and R3 increases the affinity of the compound from 22 μM to 0.14 μM by replacing an increasing number of energetically unfavorable water molecules. Additional water density with unfavorable free energy is located adjacent to the pyrimidine ring of the initial scaffold. Extending the pyrimidine scaffold to a pyrrolo-pyrimidine group and adding substituent at Q1 and Q2 position replaces those additional unfavorable water molecules which increases the affinity by almost 10-fold to 15 nM.

Quantitative regression analysis was performed with the aim to correlate desolvation free energy obtained from the point-wise NN with experimental binding affinities. For each ligand atom, the desolvation free energy is computed by trilinear interpolation based on the hydration free energies on the eight grid points that surround the atom. All atomistic desolvation free energies are summed up. Linear regression between desolvation and binding free energy yielded a regression coefficient of r^2=0.70 (Figure 2.14e).

A similar retrospective analysis was performed on BACE-1 (Figure 2.15). Focusing on the R-group of the terminal phenyl ring (Figure 2.15b), density with unfavorable free energy is found adjacent to the R-group (Figure 2.15a, red surface on the right). Methoxy substitution



Cmp.	R1	R2	R3	K_i [µM]
1	H	H	H	22
2	OCH_3	H	H	7.9
3	OCH_3	Cl	H	2.7
4	Cl	Cl	H	0.35
5	Cl	Cl	OCH_3	0.14

Cmp. 6

Cmp.	Q1	Q2	K_i [µM]
7	H	H	2
8	$COCH_3$	H	0.2
9	F	H	0.2
10	Cl	H	0.2
11	CN	H	0.04
12	H	CH_3	2
13	Cl	CH_3	0.1
14	CN	CH_3	0.03
15	CN	CH_2CH_3	0.015

Figure 2.14. SAR of HSP90 inhibitors. SAR of HSP90 inhibitors guided by gain in desolvation free energy based on point-wise neural network model. (a) Co-crystalized compound 5 in PDB structure with ID 3rlp. (b) SAR table of 15 inhibitors with substituents replacing water density with unfavorable free energy (c/d: isolevel: 7.5 kcal mol^{-1}). (d) Compound 8 from X-ray structure 3rlr. (e) Linear regression between predicted desolvation and experimental binding free energy for SAR series (r^2=0.70).

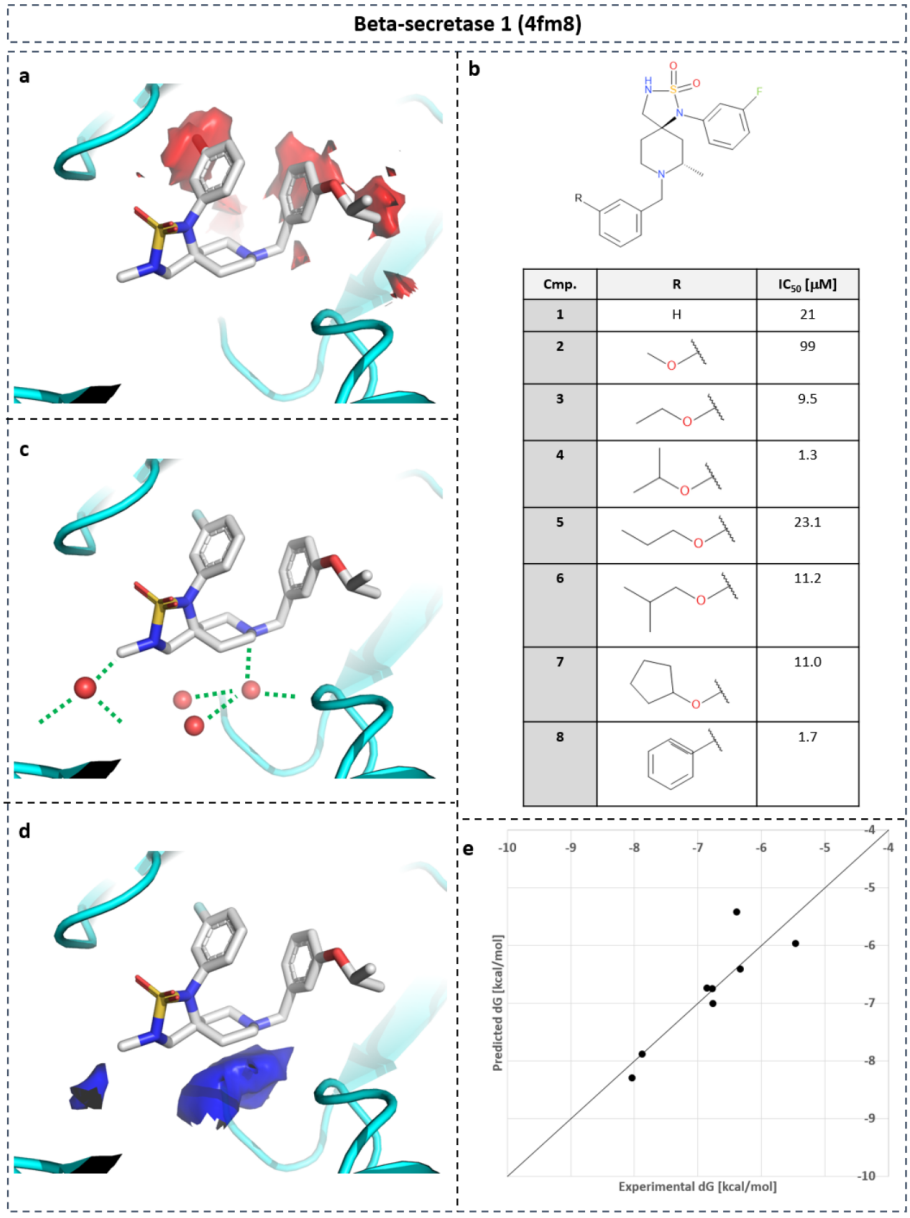


Figure 2.15. SAR of BACE-1 inhibitors. SAR of BACE-1 inhibitors guided by gain in desolvation free energy based on point-wise neural network model. (a) Co-crystalized compound 4 in PDB structure with ID 4fm8. (b) SAR table of eight inhibitors with substituents replacing water density with unfavorable free energy (a: isolevel: 7.5 kcal mol^{-1}). (c) Water-mediated protein-ligand interactions overlap with water density with favorable enthalpy (d: isolevel: -3 kcal mol^{-1}). (e) Linear regression between predicted desolvation and experimental binding free energy for SAR series (r^2=0.78).

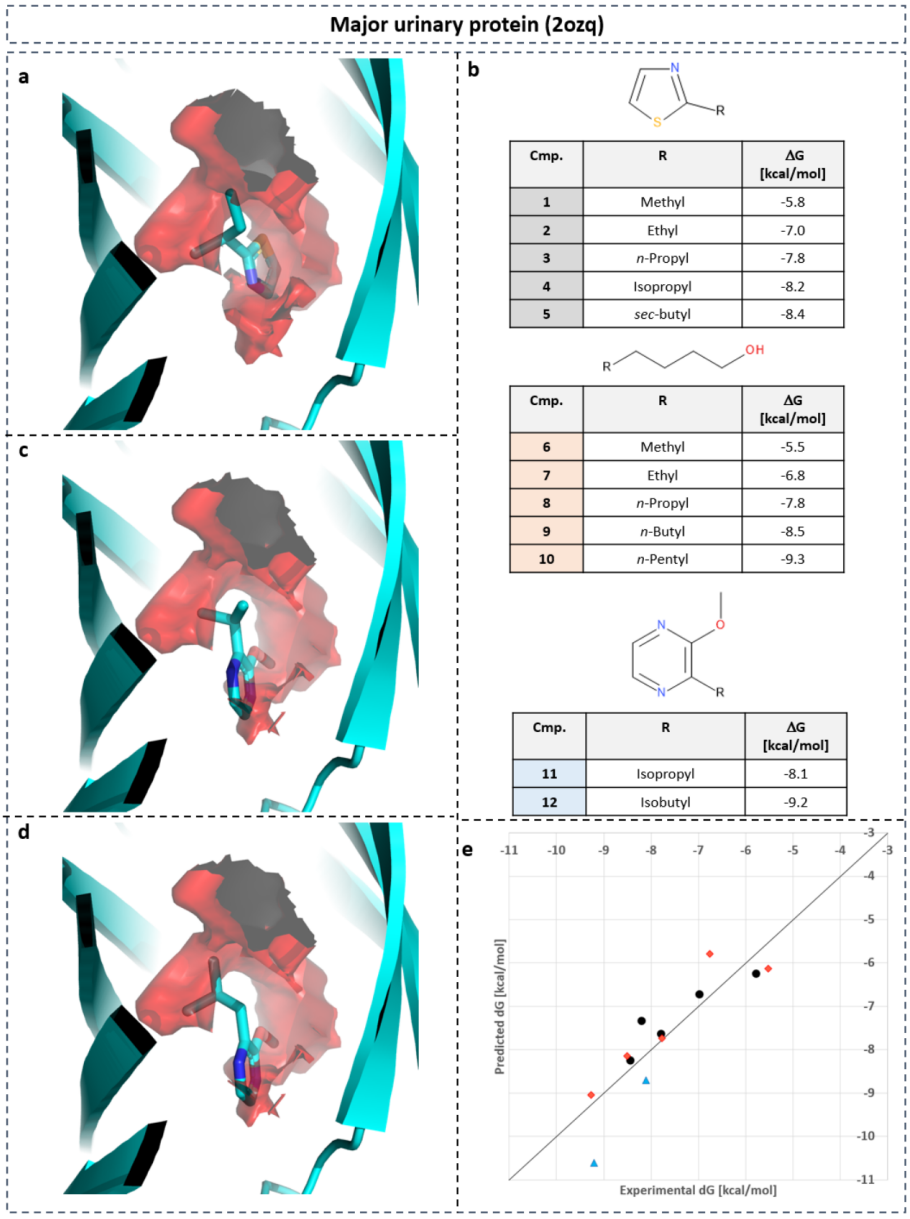


Figure 2.16. SAR of MUP inhibitors. SAR of MUP inhibitors guided by gain in desolvation free energy based on point-wise neural network model. (a) Co-crystalized compound 5 in PDB structure with ID 1i06 with water density with unfavorable free energy (isolevel: 8 kcal mol^{-1}). (b) SAR table of 12 inhibitors with three different scaffolds and substituents replacing water density with unfavorable free energy. (c) Compound 11 from X-ray structure 1qy2. (d) Compound 12 from X-ray structure 1qy1. (e) Linear regression between predicted desolvation and experimental binding free energy for SAR series (r^2=0.77). Compounds 1-5 are displayed as black spheres, compounds 6-10 as red diamonds, and compounds 11-12 as blue triangles.

(Compound **2**) is not able to replace the water density, highlighted by a decrease in affinity. Elongated substituents such as O-ethyl (**3**) and O-isopropyl (**4**) spatially overlap with the unfavorable water density, replacing those water molecules. This results in significant affinity increase from 21 μM to 1.3 μM. For BACE-1, two regions with favorable water enthalpy were observed (Figure 2.15d, blue surface) that coincides with X-ray water molecules (Figure 2.15c) which mediate interactions between protein and ligand. Replacement of those water molecules should be considered with great care, as it may lead to a decrease in binding affinity.

Quantitative regression analysis between desolvation and binding free energy was performed for a congeneric series of eight ligands (Figure 2.15e). An excellent correlation was obtained with a regression coefficient of r^2=0.78. A similar linear regression study on the exact same dataset was previously performed using MD-simulation based hydration site analysis with WaterMap [107]. This analysis achieved an r^2 value of 0.82. This demonstrates that our NN-based efficient thermodynamic profiling of desolvation is able to generate thermodynamic profiles for hydration comparable to the time-consuming hydration analysis based on MD simulations.

Retrospective analysis was performed on major urinary protein (MUP) (Figure 2.16) [108], [109]. The series consists of twelve compounds with three different scaffolds. Figure 2.16a shows compound 5 in its X-ray structure 1i06. The two terminal methyl groups of the *sec*−butyl substituent overlaps with water density with highly unfavorable hydration free energy. Increasingly smaller substituents display decreasing overlap with positive desolvation free energy grids in agreement with reduced binding affinity. Figures 2.16c and d display compounds 11 and 12 in their corresponding X-ray structures 1qy2 and 1qy1, respectively. Compound 12 has larger overlap with water density with the most positive desolvation free energy. This results in higher binding free energy compared to compound 11.

Interestingly, quantitative regression analysis between desolvation and binding free energy revealed that not only an excellent regression within a congeneric series (black spheres: compounds 1-5; red diamonds: compounds 6-10; blue triangles: compounds 11-12) could be obtained but also among all 12 compounds that contain three different scaffolds (Figure 2.16e). An excellent correlation was obtained with a regression coefficient of r^2=0.77. A

similar linear regression study on the exact same dataset was previously performed using MD-simulation based hydration site analysis with WATsite [61]. This analysis achieved an r^2 value of 0.63. This analysis also demonstrates that our NN-based efficient thermodynamic profiling of desolvation is able to generate thermodynamic profiles for hydration similar to the time-consuming hydration analysis based on MD simulations.

These three examples highlight the potential of our neural network approach to guide SAR-series expansion by incorporating critical desolvation information including the replacement of unfavorable water molecules and enthalpically favorable molecules which mediate critical protein-ligand interactions.

Improved CNN-based pose prediction

In the second application we investigated if the hydration data instantaneously generated by the U-Net neural network model can be utilized to guide ligand pose prediction. It has been shown previously, that solvent site information generated from MD simulations can assist in detecting protein-ligand interactions and improve docking [110]. Built on these findings, the method AutoDock Bias uses such information to modify and bias the energy terms in order to achieve better performance in docking [111]. Similarly, in our previous study [70] we showed significant improvement in pose prediction accuracy by adding WATsite occupancy grids as additional input layers to a classification CNN model based on Gnina software [13]. The major issue with this approach is that generating water occupancy grids for a large dataset of protein systems using WATsite or any MD-based water prediction program is computationally expensive. Here, the idea was to investigate if water grids generated via our CNN model can replace the data produced by WATsite to enhance the performance of Gnina.

In Gnina, protein and ligand density are distributed on a 3D grid that encompasses the binding site. For this distribution, a Gaussian distribution function centered on each heavy atom centroid is used. For each atomic element, a separate distribution is computed for protein and ligand. This ensemble of occupancy grids is used as different channels of the input layer of a CNN that classifies native-like poses (RMSD $<$ 2 Å) from decoy poses

(RMSD $>$ 4 Å). Water occupancy grids predicted by our CNN model was used as additional input channel to the Gnina CNN.

To provide water occupancy data for Gnina, we retrained the water predictor network using 2288 and 1133 PDBs for training and test set, respectively. The training and test sets were based on the reduced set from Ragoza et al.[13]. However, we increased the number of bad poses for a more realistic scenario. For each target protein, only one native-like pose with RMSD $<$ 2 Å was selected. Since we aimed to utilize the Gnina CNN with and without hydration information for pose reranking, systems with no good poses were removed. The final data set consists of 1394 and 593 protein targets for training and test, respectively. The training was performed for 10000 iterations. We used the default parameters and the reference model for pose prediction which is made available on Gnina's Github page (https://github.com/gnina/gnina).

Here, we evaluated the performance of Gnina+water against Gnina alone and Vina/Smina. The results for Vina were obtained from Ragoza et al. [13].

As it can be seen in Figure 2.17, inclusion of hydration occupancy from our neural network model into Gnina significantly increased the performance of Gnina on the test set.

2.4 Conclusion

Hydration is a key player for biochemical association processes such as protein-ligand and protein-protein binding. The binding partners and the association process itself influence hydration patterns and thermodynamic properties. In order to accurately model hydration in tasks such as flexible protein-ligand or protein-protein docking, the hydration data needs to be computed in an efficient manner without performing time-consuming simulations. In this paper, we demonstrate that instantaneous prediction of thermodynamic properties of biochemical systems is possible due to the development of machine learning algorithms and due to our ability to generate large amount of thermodynamic data. Here, we present the very first deep learning methods to instantaneously predict thermodynamic hydration data, thus providing an efficient alternative to time-consuming MD simulations for the calculation of those properties.

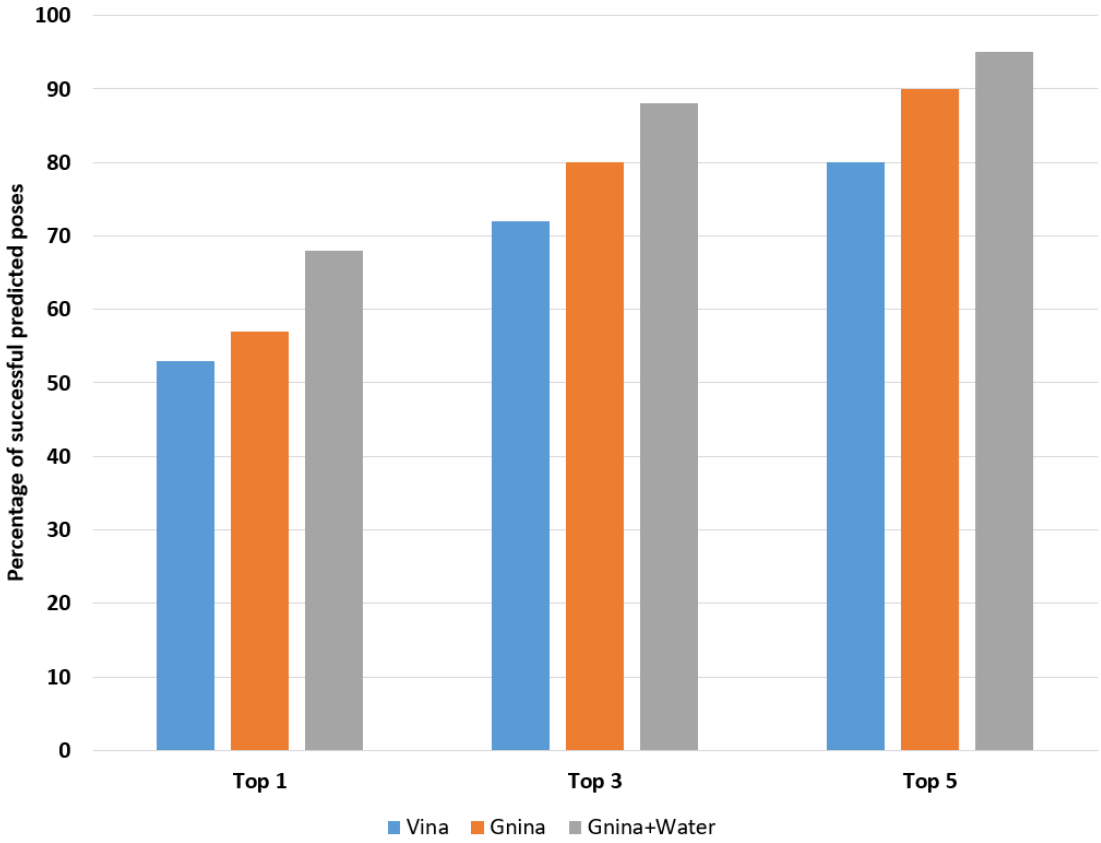


Figure 2.17. Ranking of docking poses. Percentage of protein systems with native pose (RMSD < 2 Å) in the test set within the top-1, top-3, and top-5 ranked poses using different scoring functions: Vina (blue), CNN with protein and ligand information (orange), and CNN with protein, ligand and WATsite occupancy information generated by U-Net model (grey).

We have developed two alternative deep learning approaches. One method predicts the complete binding site hydration information in a single network calculation in form of U-Net neural networks. The second method relies on descriptors that include potential protein-water and water-water interactions calculated on each grid point. The networks were able to generate precise hydration occupancy and, in case of the point-wise model, also thermodynamics data.

Application of the predicted hydration information to SAR analysis and binding-mode prediction demonstrated the potential of these methods for structure-based ligand design. Future applications include the marriage of protein flexibility and desolvation data in ensemble docking. Due to the efficiency of the methods, precise hydration data could be computed for alternative protein structures, different ligands and their binding poses in modest computation time, which has been an unfeasible task until now. The routine inclusion of explicit desolvation, water-mediated interactions and enthalpically stable hydration networks around the protein-ligand complex [70] may become possible in structure-based ligand design in the near future.

3. SEQ2MOL: AUTOMATIC DESIGN OF DE NOVO MOLECULES CONDITIONED BY THE TARGET PROTEIN SEQUENCES THROUGH DEEP NEURAL NETWORKS

3.1 Introduction

De novo design of molecules is an important approach for the discovery and development of drugs. In recent years, artificial intelligence-based methods, in particular deep learning-based methods, have been employed to facilitate this process and open new possibilities for the design of new molecules. Many generative types of architectures have been used for the task of de novo molecule generation, such as autoencoder-based models, generative adversarial neural networks (GANs), recurrent neural networks (RNNs) and models combined with reinforcement learning [112]. Those approaches have been focused on the generation of compounds based on a pool of training compounds and have been mostly focused on engineering molecules with specific physicochemical properties. Whereas most approaches display inherent structural similarity of the generated molecules to the original "seeds", some newer approaches aim to increase the diversity of generate molecules compared to the training set [33]. In the context of drug design, current approaches aim to generate novel compounds that resemble features (e.g. physicochemical properties or pharmacophore features) from already known binders to a specific target. A major drawback to this approach is that in many instances, e.g. when a target is newly discovered, there is no known ligand for the specific target. Therefore, no known pharmacophore elements or physicochemical features of known compounds can guide the de novo molecule generation process. Even if compounds for a target are known they are often limited in number, making a target specific training without overfitting unlikely. Therefore, there is a need for methods to be able to condition the generative process on the biochemical features of the target.

Our method for the generation of de novo compounds is conditioned on the sequence of a target protein. The concept is based on the "Show and Tell" image captioning method developed by Vinyals et al [113]. The question is, what do image captioning and de novo molecule design have in common? To answer this question, we take a closer look of how the

"Show and Tell" image captioning method functions. In image captioning, the task is to generate a sentence with the following two essential properties: First, it should be relevant to the image, and second it should be grammatically correct and meaningful. In image captioning, essential features of a given image are extracted, e.g. using a neural network. The embedding vector is then used to generate a meaningful sentence which describes the image. Similarly, in de novo compound generation essential features of the target protein are extracted ('image features') and molecules are generated in form of SMILES strings ('caption'). The target protein is represented by its sequence. A neural network learns embeddings that should represent the features of the target essential for ligand binding. These features are linked to their corresponding 'caption', that is, the characters of the SMILES strings representing the chemical compound. The caption generator network also tries to learn to generate valid and meaningful sentences with the correct grammar, which is also critical in the task of SMILES generation to obtain chemically valid molecules.

In this work, we use protein sequence embeddings generated using a pre-trained bi-directional language model ELMo, and use an LSTM model combined with reinforcement learning to generate SMILES strings of de novo compounds for two important target families: GPCR and Tyrosine Kinases. Protein-ligand datasets published in BindingDB are used for training and validation of the model [114].

3.2 Methods

3.2.1 Datasets

We used the dataset from the work by Karimi et al [115] which originates from BindingDB [114]. The original dataset which contains all IC_{50}-labeled ligand-target pairs from BindingDB was reduced to protein-ligand pairs with IC_{50}-values of less than 1 mM. The data was directly taken from "BindingDBAll2018m8.tsv.zip" file provided by BindingDB, which contains protein sequences as well as their corresponding 2D compounds structure in SMILES. For correct and independent validation of the resulting model on two families of protein targets, GPCRs and Tyrosine Kinases, any protein-ligand pair matching one of those two classes was removed from the training based on records from the Uniprot database

[116]. Also, any compound that existed both in training set and test sets (even when bound to a different target than GPCR or Tyrosine Kinases) was removed from the test sets. This resulted in a training set of 127546 entries and test sets for GPCR and Tyrosine Kinases with 276 and 109 target proteins and 37749 and 25578 binding molecules, respectively.

3.2.2 General workflow

Figure 3.1 shows the overall workflow of the method described in this paper. Our method takes advantage of the embedding generator network developed by Heinzinger et al [117] which generates the embeddings of protein sequences. Then the embeddings are used as inputs to the molecule generator network for the initial training. Subsequently, the model is retrained using reinforcement learning to increase the diversity and novelty of the generated compounds.

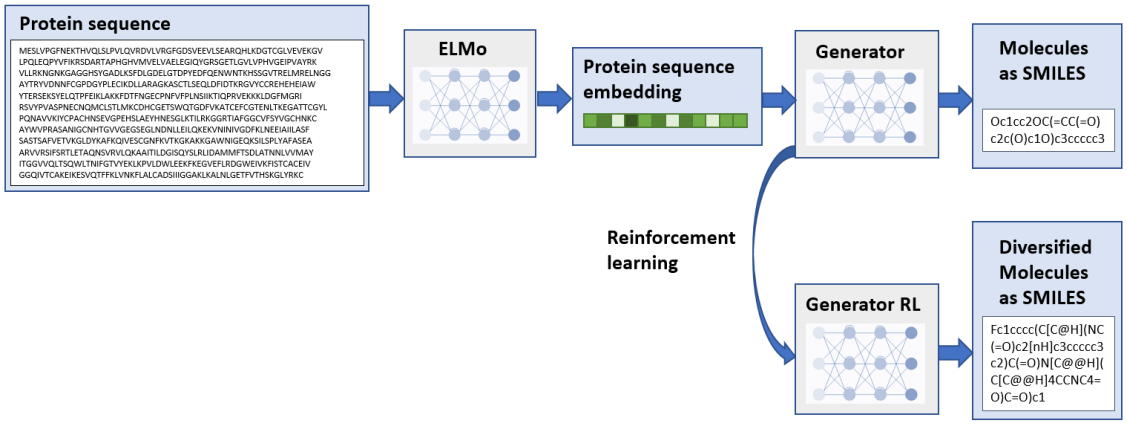


Figure 3.1. Overall workflow of de novo compound generation method using deep neural networks. First, sequence embeddings are generated using the network from Heinzinger et al [117]. Then the compound generator is trained using the embeddings as input. After the initial training, the network is re-trained using a reinforcement learning scheme using the dissimilarity to the training set as reward to get more diverse compounds.

3.2.3 Fingerprint generation

We used Morgan fingerprints available in RDKit library [118] to analyze similarities between compounds. A radius of 4 and the bit vector length of 2048 was used to generate the fingerprints. We used Tanimoto distance (1 - Tanimoto similarity) to report the similarity in our studies, with a value of zero measuring exact identity while a value of one indicating complete dissimilarity of molecules.

3.2.4 Random molecule set generation

To measure the target specificity of the molecules generated with our neural network, those compounds were compared with randomly selected compounds. The latter molecules were selected from the emolecules database www.emolecules.com using the following criteria: Molecules were only selected if their log P value and molecular weight is similar to the corresponding values of known GPCR and Tyrosine Kinase binders, respectively. In detail, only compounds with log P values and molecular weight are selected that deviate by less than the standard deviation from the mean of the corresponding values of the known binders for the two target families. All properties were calculated using the RDKit package. This selection process guarantees to test the model for its ability to generate target-specific ligands and not just compounds with similar physicochemical properties as known binders.

3.2.5 Generation of protein sequence embeddings

To generate embeddings of protein sequences we used the model provided by Heinzinger et al [117]. Embedding vectors of length 1024 are generated. The network consists of one CNN layer and two bidirectional LSTM layers, which provide context information of the surrounding residues. For any query protein sequence, the output feature vectors of length 1024 of all three layers were summed component-wise and finally averaged over all residues of the whole protein sequence to obtain a single vector representing a protein sequence. This sequence-embedding vector is used as input for the molecule generator network.

3.2.6 Compound generation from protein sequence embeddings

Architecture of generator model

The image-captioning network architecture "Show and Tell"[113] based on a Keras [94] implementation was used to generate molecules in form of SMILES strings. The model is a LSTM-based sentence generator based on given embeddings. The LSTM model's task is to predict a new SMILES character based on previously predicted SMILES characters and the protein's sequence, with probability $p\left(s_t|I, s_0, \ldots, s_{t-1}\right)$, where I is the protein's sequence embedding, and s_t is the SMILES character at position t. The model is illustrated in Figure 3.2. For any given position t the output of the LSTM cell depends on the cell state which is the result of the current input and previous cell state at position $t-1$. Therefore, the LSTM network keeps memory of past characters. The protein's sequence embedding is only used for the initiation of the LSTM cell and hidden states. Due to the recurrence of LSTM networks the protein sequence, however, influences the cell state for subsequent character predictions.

$$x_{-1} = \mathrm{ELMo}(I) \tag{3.1}$$

Thus the prediction of all characters is influenced by the conditioning from the protein's sequence.

Each possible character in the SMILES string S at position t, s_t, is tokenized and represented as one-hot vectors. Every sequence begins with a special character representing the start of the string at position $t = 0$ and ends with an end character token (position $t = N$). Each character token passes through an embedding layer (W_e) prior to being input to the LSTM cell.

$$x_t = W_\mathrm{e} s_t, \quad t \in \{0 \ldots N-1\} \tag{3.2}$$

The LSTM cell predicts the probability for a character at position $t+1$ by

$$p(s_{t+1}) = \mathrm{LSTM}\left(x_t\right), \quad t \in \{0 \ldots N-1\} \tag{3.3}$$

The loss is computed by summing the negative log likelihood of each correct character token:

$$L(I, S) = -\sum_{t=1}^{N} \log p_t\left(s_t\right) \tag{3.4}$$

The SMILES strings of the known training compounds were tokenized in character-level and were fed to the network for training (Figure 3.2). The network was trained for 50 epochs with batch size 512. The dimensions of the embedding layer (W_e) was set to 2048, and the protein embeddings were simply tiled to have the same dimensions, as the character embedding layer, i.e 2048. The dimensions of the one-hot encoding vectors were 47×102 which are the number of possible SMILES characters in the dataset and the maximum length of the SMILES string, respectively.

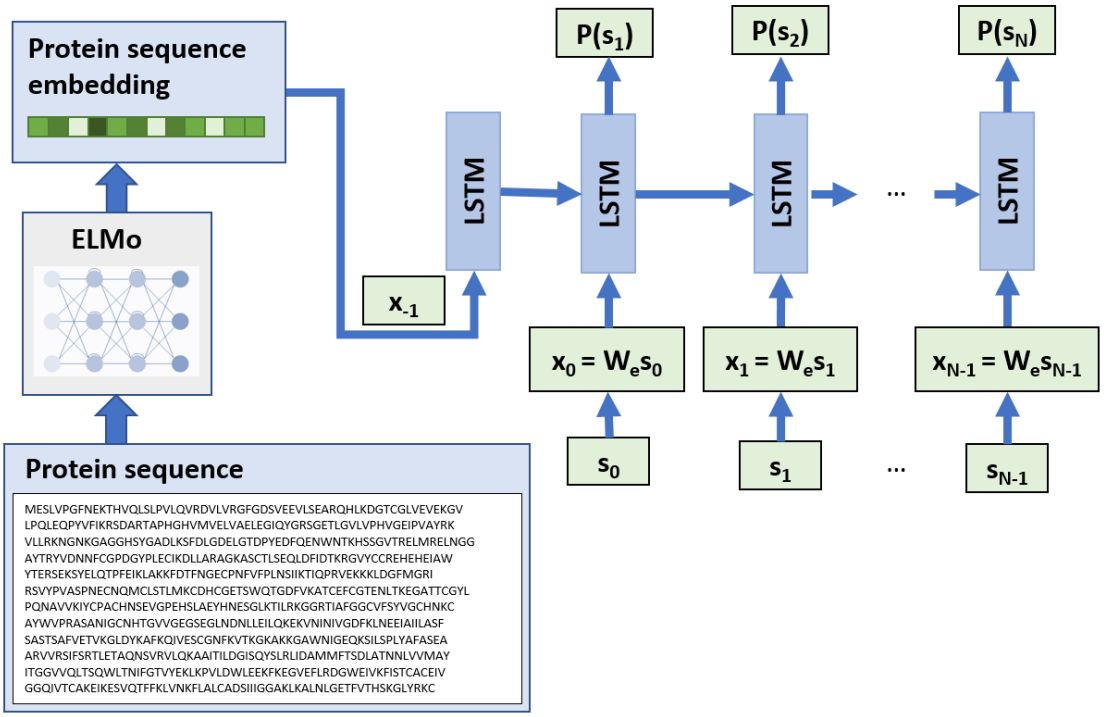


Figure 3.2. Molecule generator network is defined by combining LSTM model and protein sequence embedder model. The LSTM model is showed in unrolled form, where recurrent connections are shown as feed-forward connections. During training, target sequence tokens (s_t) are learned by maximizing $P(s_t)$, where t denotes the character position in the SMILES string. Each token is passed through an embedding layer prior to LSTM.

Generation of new molecules

The trained model can be used to generate new molecules for a given protein sequence. The protein's embedding and the start token are given as initial input to the network (cf. Figure 3.2). Subsequently, the LSTM network is utilized to generate the characters of the SMILES string until the end token was selected as output of the LSTM. To increase the diversity of the generated molecules, the LSTM network is used within the framework of a beam search. Thus, not only one character is selected at each position t but the top k SMILES strings at position t are selected and passed on to the next LSTM iteration, keeping the best k strings and so forth. (Figure 3.3). In our case, we used a value of 46 for k. This approach allows to generate a diverse set of molecules specific to the input sequence of the target protein. Whereas selecting only one (the most probable) character at each position (greedy search) is the best choice for that specific position, it often results in sub-optimal solutions when the full string is considered [113]. In de novo compound generation, it is often desirable to have a diverse set of compounds generated for a target, rather than just one compound, especially when invalid or already known compounds are obtained by using a greedy search.

Reinforcement learning

To increase the novelty of the compounds when compared to the training set, a reinforcement learning procedure was employed using a concept adapted from the work by Olivecrona et al [35]. Figure 3.4 illustrates the reinforcement learning procedure. The network that was initially trained following the procedure described in the in the previous section functions as Prior. The Agent is instantiated by a copy of the Prior network. Thus, initially, both the Prior and the Agent are identical. During reinforcement learning, a policy is learned by the Agent to generate compounds with desired features, here potential binders to a target protein but diverse to the initial training set. As described in the previous section, generated compounds are represented by SMILES strings. Those strings are generated by sampling one character at each LSTM step until the end token is reached. This process can be considered

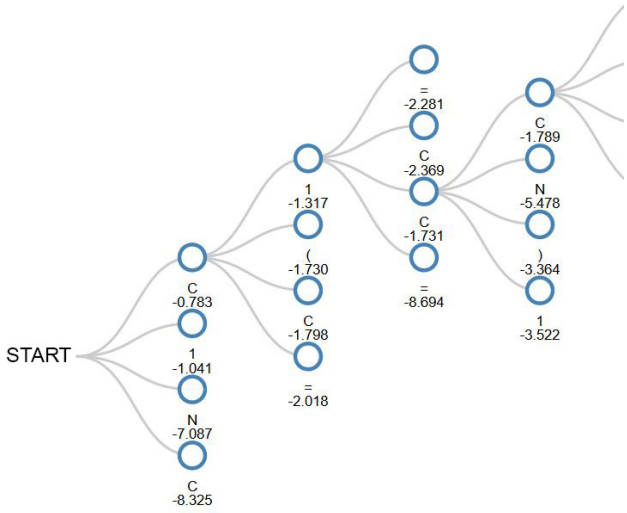


Figure 3.3. An example of beam search for generating new molecules. In each step, character candidates are ranked based on scores (natural logarithm of probabilities predicted by the network). Top k best candidates are considered. At each step a path is generated from one layer to the next layer forming a tree. Each path from the start token to the end token is considered a full SMILES string (molecule). For simplicity and a clearer illustration, only one path and a segment of the full tree is shown here.

as a set of actions $A = a_1, a_2, ...a_N$ that composes an episode of a SMILES string generation. The likelihood for a SMILES string generated by the model is

$$P(A) = \prod_{t=1}^{N} \boldsymbol{\pi}\left(a_t \mid x_t\right) \tag{3.5}$$

where $\boldsymbol{\pi}$ is the policy learned by the model and x_t being the input to the LSTM at step t.

To increase the likelihood the generation of SMILES strings different to the training set, a scoring function $\Sigma(A)$ is added to the Prior log-likelihood

$$\log P(A;\theta)_{\text{Augmented}} = \log P(A;\theta)_{\text{Prior}} + \sigma\, \Sigma(A) \tag{3.6}$$

where $\Sigma(A)$ measures the diversity of the generated compound with respect to the training set. Thus, high $\log P(A;\theta)_{\text{Augmented}}$ is achieved by the generation of SMILES with high probability based on the Prior network and with diversity to the training set. θ are the

trained weights of the Prior network. σ is a user-defined coefficient, which in our case was set to 60.

Using this augmented log-likelihood the Agent's policy $\boldsymbol{\pi}$ is updated from the Prior's policy $\boldsymbol{\pi}_{Prior}$ to approximate the augmented likelihood $\log P(A;\theta)_{\text{Augmented}}$. Thus, the weights θ of the Agent's network are optimized using the loss function

$$L(A;\theta) = \left[\log P(A;\theta)_{\text{Augmented}} - \log P(A;\theta)_{\mathbb{A}}\right]^2 \tag{3.7}$$

which measured the squared difference of the current Agent's likelihood over a set of actions A, $\log P(A)_{\mathbb{A}}$, and the augmented likelihood.

The Agent was trained for 100 iterations with a batch size of 512.

In our case, the goal was to increase the diversity of the generated compounds compared to the training set. To achieve this goal, Morgan fingerprints are computed for all training data and each generated molecule. The Tanimoto distances between the fingerprint of the generated molecule (m_g) and the set of fingerprints of all training molecules (M_t) are computed. The scoring function to calculate the reward for a generated molecule is then determined by identifying the minimum Tanimoto distance ($\mathrm{T_d}$):

$$\Sigma(A) = \min_{\forall m_t \in M_t} \mathrm{T_d}(m_g, m_t) \tag{3.8}$$

Therefore, compounds with higher Tanimoto distance compared to the training data are rewarded.

3.2.7 Benchmark

The MOSES framework [119] was used to compare our method with other approaches to generate molecules. The MOSES framework provides pre-defined models and metrics available for comparison. The metrics and models that were used to compare the performance of our model with other models are briefly described below.

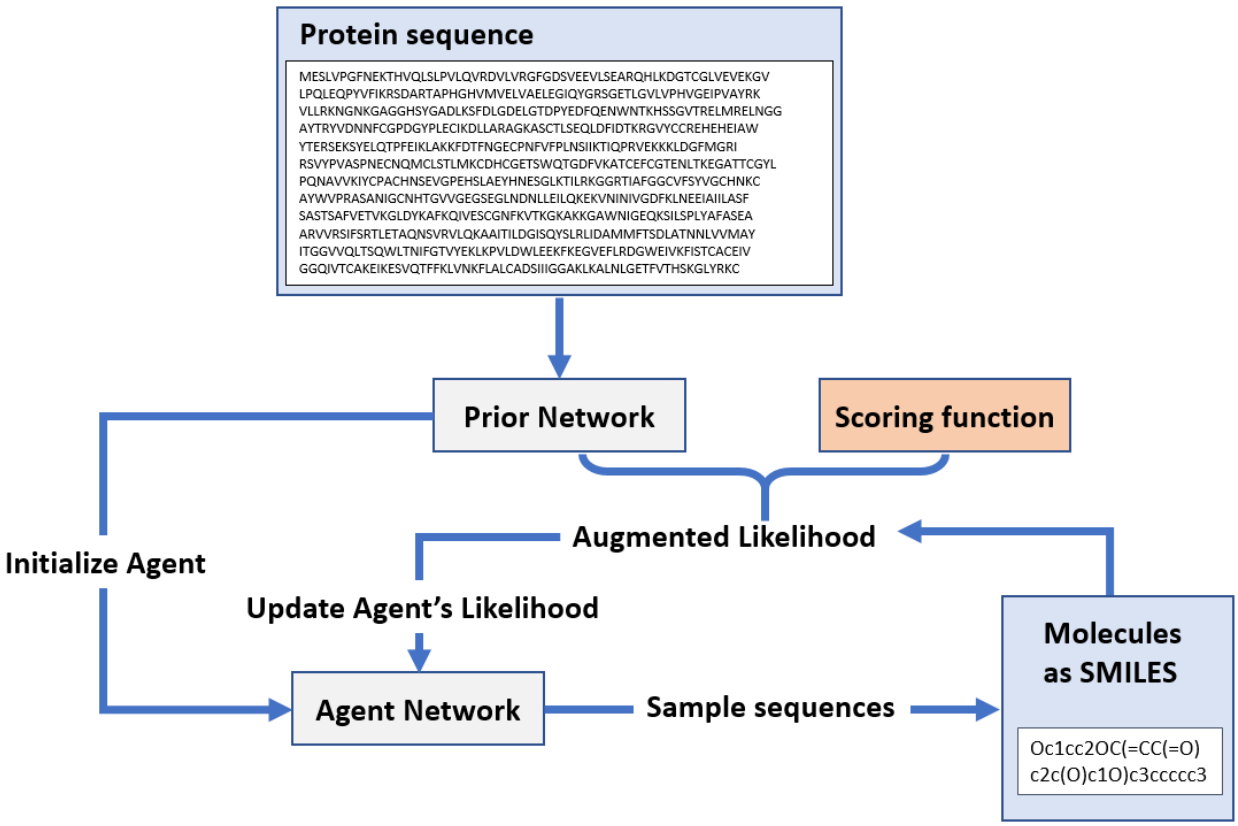


Figure 3.4. Encouraging diversity and novelty of generated molecules through reinforcement learning. First, the Agent network is initialized from the already trained Prior network. The Prior likelihood is then augmented by the addition of a score that measures the structural diversity of the generated compound to all training molecules. This likelihood is used to train the Agent network.

Benchmark metrics

Fragment similarity

The BRICS algorithm [120] in RDKit is used to fragment molecules and measure the cosine distance between fragment frequencies vectors:

$$\text{Frag}(G, R) = 1 - \cos\left(f_G, f_R\right)$$

f_G and f_R represent frequency vectors of generated and reference molecule, respectively. The size of the fragment vocabulary of the whole data set determines the size of the frequency vector and the elements of the vectors are the frequencies for each fragment in the molecules.

Scaffold similarity

This metric measures the cosine similarity between vectors representing the scaffold of generated (G) and reference (R) molecules:

$$\text{Scaff}(G, R) = 1 - \cos\left(s_G, s_R\right) \tag{3.9}$$

The scaffolds are generated using Bemis–Murcko scaffolds algorithm [121] implemented in RDKit.

Distance to the nearest neighbor

The similarity is computed by averaging over the Tanimoto similarity value (T) between a molecule m in the generated and reference sets. The default configurations of the MOSES framework was used to generate Morgan fingerprints for this task.

$$\text{SNN}(G, R) = 1 - \frac{1}{|G|} \sum_{m_G \in G} \max_{m_R \in R} T\left(m_G, m_R\right) \tag{3.10}$$

Internal diversity

This metric measures the diversity among the molecules within the generated set.

$$\text{IntDiv}(G) = 1 - \frac{1}{|G|^2} \sum_{m_1, m_2 \in G} T\left(m_1, m_2\right) \tag{3.11}$$

Other metrics

In addition to metrics above, other commonly used metrics were used to evaluate the quality of the generated molecules. The metrics used are the following:

LogP: The water-octanol partition coefficient, calculated using the approach of Crippen [78].

Synthetic accessibility score: [122] A score to estimate synthetic accessibility of a molecules. Values closer to 1 indicate the compound is likely to be synthetically accessible, while values closer to 10 are expected to be difficult to synthesize.

Quantitative Estimation of Drug-likeness (QED): A metric developed by Bickerton et al. [123] which addresses the drug-likeness of molecules based on the notion of desirability and can range between 0 to 1.

Natural product-likeness score: A measure to estimate into which of the following three categories a molecule will fall into: (1) A natural product (score between 0 and 5), (2) a synthetic product ([-5, 0]) and a drug molecule ([-3, 3]). The metric uses several substructure descriptors to determine the score [124]

Molecular weight: The sum of atomic weights in a molecule.

Benchmark models and training

All models used to benchmark against our model are pure ligand-generation models without consideration of any information about the target protein. The models generate novel SMILES strings based on known molecules that are also represented as SMILES. We used only the ligands (without target sequences) from our training set as training data for the benchmark models to generate novel molecules.

Character-level recurrent neural networks (CharRNN)[125]:

This model considers the SMILES as a language model and treats each SMILES character as a word. CharRNN contains three LSTM layers and each hidden is of size 768. Dropout layer are added with dropout probability of 0.2. A Softmax function is used as the activation function of the output layer. Adam optimizer [6] is used to optimize the model's parameters using Maximum likelihood estimation (MLE). The training is done in 80 epochs with batch

size equal to 64 with a learning rate set to 10^{-3} which is halved every 10 epochs. We used the model implemented in MOSES.

Variational Autoencoder (VAE)

This model consists of two components, an encoder and a decoder. The encoder maps the input data to a lower-dimensional representation (embedding) and the decoder converts it back. The encoder is a bi-directional Gated Recurrent Unit (GRU) with a linear activation function. The decoder consists of three GRU layers with size of 512 and dropout layers with probability of 0.2. The training is done in 100 epochs with batch size of 128 by using Adam optimizer with learning rate set to 3×10^{-4} to minimize the loss containing reconstruction loss between reconstructed and input SMILES strings and Kullback-Leibler (KL) divergence in latent space. The KL term weight is linearly increased from 0 to 1 during the training. Gradient clipping with the value set to 50 is used.

Adversarial Autoencoder (AAE)

In this architecture of autoencoder the Kullback-Leibler divergence loss is no longer present. Instead, an adversarial loss is used to train the generator model in form of a discriminator network which is trained simultaneously with the autoencoder [28]. The encoder and decoder are created from a 1-layer bidirectional LSTM and a 2-layer LSTM respectively, both with size of 512, and an embedding layer with a size of 128 which is shared by both. The discriminator is composed of two fully connected layers with size of 640 and 256, using Exponential Linear Unit (ELU) activation function. The model uses Adam [6] optimizer trained for 120 epochs with batch size 512. Learning rate is halved every 20 epochs with the initial value of 10^{-3}.

3.3 Results

3.3.1 Protein's sequence embeddings for ligand generation

The type of ligand that can bind to a target protein depends on the topology and physicochemical properties of the binding site of the protein. The form and properties of the binding site is determined by the structure of the protein, which is dictated by the sequence of the protein. As there exist only 20 natural amino acids but many thousands of structurally diverse protein structures, the sequence of the amino acids gives the structural "meaning" to the protein object. This is similar to Natural Language Processing (NLP) where an enormous number of different sentences are constructed from a smaller library of words.

It is common to use word embeddings in the field of Natural Language Processing (NLP). Similar words can have different contexts appearing in different sentences, represented by different embeddings. In the same way, models used in NLP such as Embeddings from Language Model (ELMo) can be used to generate contextualized embeddings of the sequences. Heinzinger et al. trained such a model on the Uniref50 dataset. Using a fixed model for sequence embedding that has been already trained on such a large dataset (33 M sequences) provided better performance in our study compared to a network that learns the sequence embeddings in parallel to the training of a generator network for molecules. One major reason for this observance is that the existing protein-ligand binding datasets contain a much smaller number of protein sequences compared to Uniref50. This approach was also taken in the work "Show and Tell", by using embeddings generated from VGG16 trained on ImageNet, one of the largest image datasets available. The actual data set, that the "Show and Tell" model was trained on, was much smaller. Figure 3.5 shows the separation of embedding vectors for all protein sequences in our data set as generated by the SeqVec model. The embedding vectors are projected on a reduced 2D representation using T-SNE available in scikit-learn library [126].

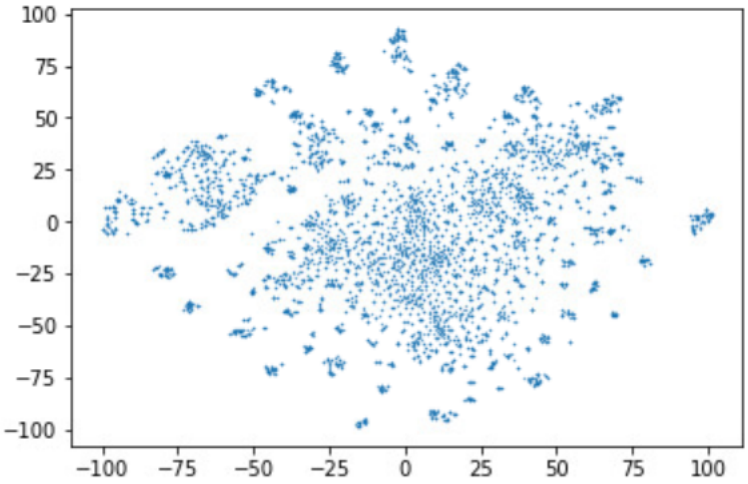
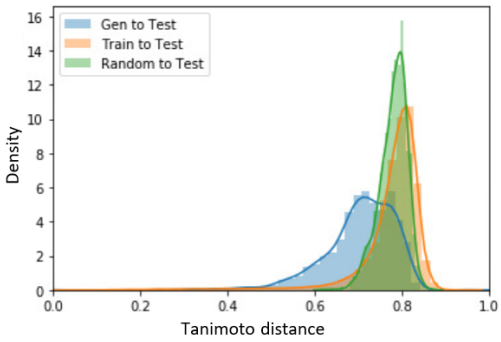


Figure 3.5. Sequence embeddings of protein targets in our data set generated by SeqVec, visualized using T-SNE.

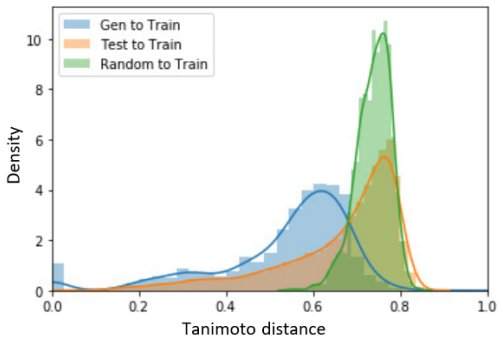
3.3.2 Encouraging compound diversity and novelty via reinforcement learning

We generated 1653 unique novel molecules targeting 109 different Tyr kinases and 1672 compounds targeting 276 different GPCRs. Figure 3.6a displays the similarity of our generated compounds compared to known Tyr kinase ligands (Test set). The generated compounds show overall a higher similarity to known Tyr kinase binders compared to the set of training molecules or randomly selected compounds from emolecules. Figure 3.6b, however, shows that despite reinforcement learning for diversity, the generated compounds still show similarity with compounds from the training set that exceeds that of randomly selected molecules. There are, however, also several test compounds that have high similarity to at least one training compound. This observation is due to the fact that the training set contains other kinases (non-Tyrosine kinases) with ligands similar to Tyrosine kinase binders.

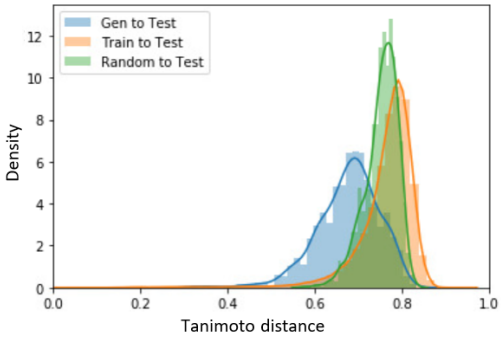
Similarly, compounds generated for GPCRs display a higher similarity to known GPCR ligand compared to training compounds or randomly selected molecules (Figure 3.6c). While the generated compounds again have inherent similarities to some training set molecules, this similarity is less pronounced compared to the Tyrosine kinase case (Figure 3.6c), as the target family of GPCRs has no similar proteins in the training set. Despite the remaining



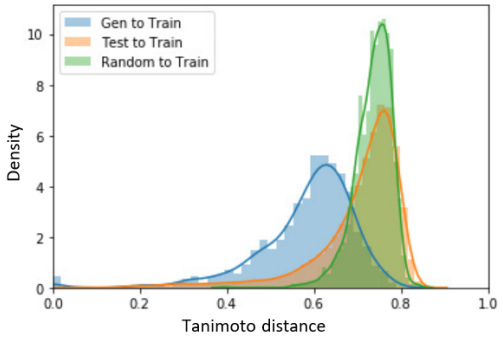
(a) Similarity of the generated molecules (Gen) for Tyr kinases to the test set (Test) that contains Tyr kinase ligands. Higher similarity was observed compared to the similarity between training (Train) and test set or randomly selected molecules (Random) to test set.



(b) Similarity of the generated molecules for Tyr kinases to the training set. Several compounds show high similarity to training compounds. Whereas the training set lacks any Tyr kinase ligands it still contains compounds similar to those ligands.



(c) Similarity of the generated molecules for GPCRs to the test set that contains GPCR ligands.

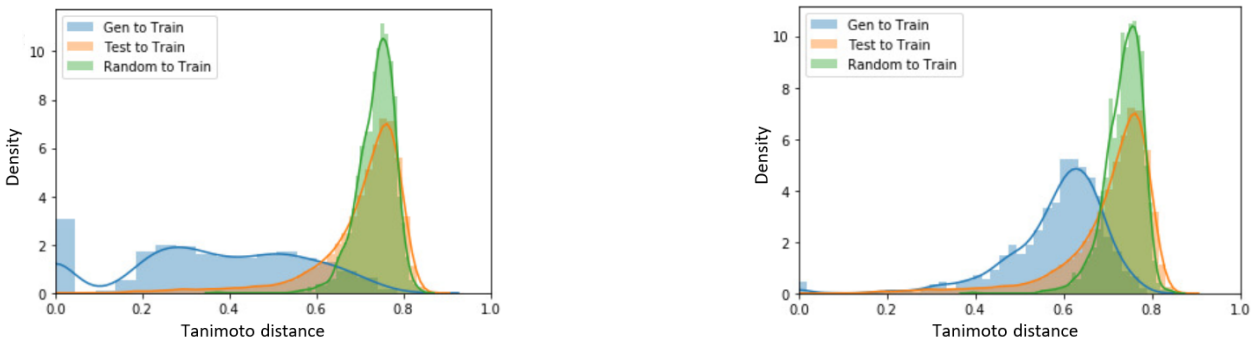


(d) Similarity of the generated molecules for GPCRs to the training set that does not contain GPCR ligands.

Figure 3.6. Similarity of the generated compounds for the Tyrosine kinase targets compared to the Tyrosine kinase test set (a) and the training set (b). The network generates more similar compounds for the Kinase targets than when compounds are selected randomly. In (b) it is observed that some compounds in the test set are similar to compounds of the training set. The reason for this are the existence of other kinase targets (non-Tyr kinases) in the training set that share similar compounds. The same plots are shown for the generation of GPCR ligand in (c) and (d). Density graphs were smoothed using kernel density estimation (KDE) available in the Seaborn library [127].

similarity between test and some training compounds, Figure 3.7 demonstrates the effects of reinforcement learning to increase the novelty of the molecules.

As discussed in the Material and Methods section, no member of the two target families used for the test of the models was present in the training set. Nevertheless, it is known that binding pockets of different protein families can share similarities in their binding pocket topology and properties. These similarities can results in sharing the same endogeneous ligand. For example, ATP can bind to many different target families [128]. Another clinical observation of binding pocket similarity is the occurrence of side effects of drugs binding to multiple targets, primary or secondary. This fact can also be exploited in drug repurposing. Therefore, it is not unlikely that ligands similar to the training set are generated for the two target families of our test set.



(a) Before reinforcement training

(b) After reinforcement training

Figure 3.7. Encouraging diversity using reinforcement learning. Similarity distributions for GPCR targets (Tanimoto measure) are shown for the generated compounds before (a) and after (b) reinforcement training (blue color). It can be observed that without reinforcement the network generates mostly identical or very similar compounds to the training set. With reinforcement learning the similarity between generated and training molecules is significantly decreased.

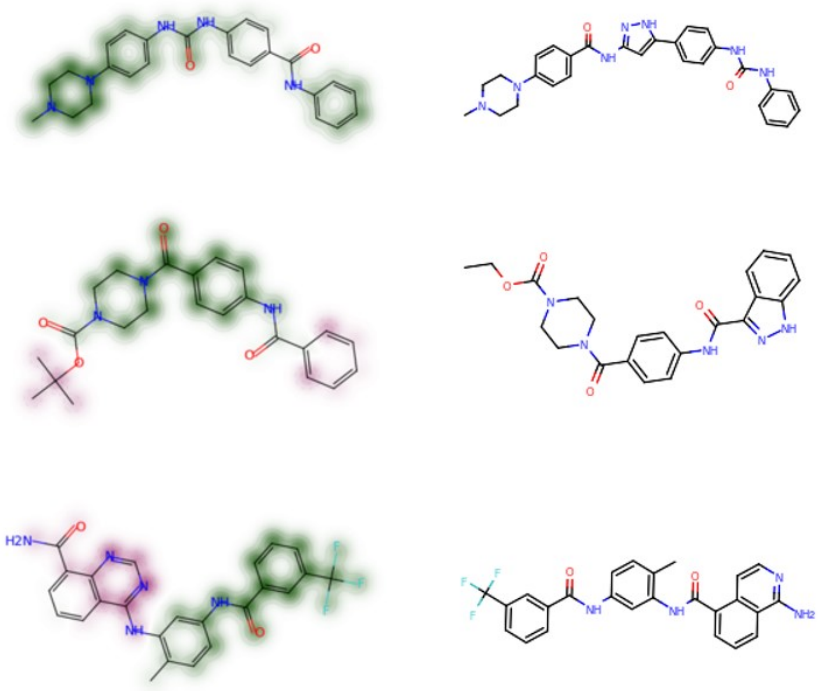
Figure 3.8 shows some examples of compounds generated for (a) Tyrosine kinases and (b) GPCRs together with their most similar known binder to the corresponding targets. They typically share a significant portion of the scaffold with known binders.

3.3.3 Comparison with benchmark models

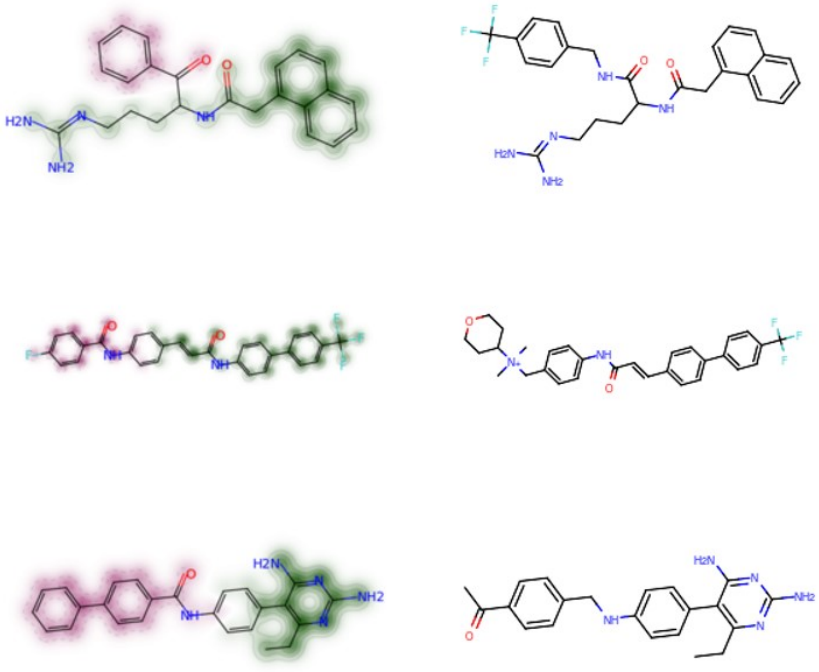
To compare the quality of our model, we used three baseline models, AAE, CharRNN, VAE as described in the MOSES framework [119]. Compounds were generated based on the

same training set and evaluated against the same test sets as described above. The metrics used in MOSES and described in the Materials and Methods section were used to evaluate model performances (Table 1). Also the comparison of fingerprint similarity with the test set for the models was carried out (Figure 9). Whereas the three benchmark models generate compounds with similar fingerprint distribution to the training distribution, the compounds generated by our model are more similar to the target compounds forming the test set, despite the fact that those compounds or targets were never seen by the network (GPCR or Tyrosine Kinase ligands). This discrepancy between our and the benchmark models is encouranging but not surprising, since benchmark models produce compounds based on the probability distribution of tokens learned from the training data. On the other hand, in our model we bias compound generation by the protein's sequence. An embedding of the protein can be thought as a "barcode" that the token probabilities are conditioned on.

Table 1 shows the comparisons of compound properties between the generated compounds by each model and the reference sets (Tyr Kinase and GPCR). The first three metrics show the similarity of the compounds to the reference set from different aspects. The first two metrics show the similarity of the sets in terms of fragments and scaffolds. Our model was able to generate compounds with smaller distance (higher similarity) in fragments for Tyrosine Kinase and GPCR targets and smaller distance in scaffold for GPCR targets, while scaffold distance is slightly higher for Seq2Mol model molecules. Distance to the nearest neighbor in test sets is lower for both Tyrisine Kinase and GPCR targets generated by our model. Benchmark models generated molecules with higher diversity. We believe this is due to the fact that benchmark models are not biased towards specific targets, so the diversity of their generated molecules is similar to the training set, which contains compounds for many different protein targets (therefore, more diverse in molecular structure). Table 2 shows metrics for basic molecular properties. It can be seen that the models were able to generate compounds with those properties within the acceptable range.

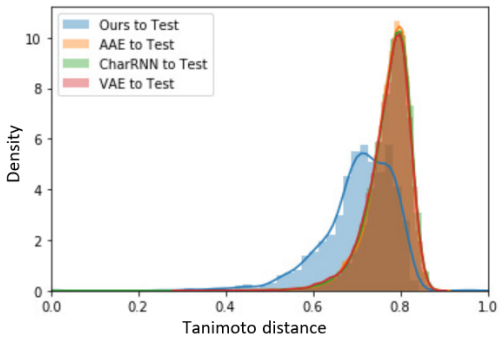


(a) Tyrosine Kinase

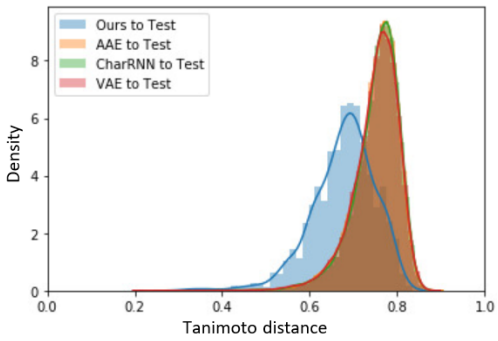


(b) GPCR

Figure 3.8. Examples of 2D similarity maps of some generated compounds (left) with low Tanimoto distance to their most similar compounds from test sets (right). Substructures with high similarity are highlighted as green, dissimilar in red. Similarity maps were generated using RDKit's similarity map function [129].



(a) Tyrosine Kinase



(b) GPCR

Figure 3.9. Comparison of similarity between generated compounds to the test sets for Tyrosine Kinase (a) and GPCR (b) using our model, AAE, CharRNN and VAE, respectively. While the benchmark models generate compounds with a similarity distribution similar to the training set, our model generates compounds more similar to the test sets, even though all models used the same set of compounds for training.

3.3.4 Limitations

Number of generated compounds

One limitation of the approach described in this manuscript is that the number of possible compounds that can be generated per target is limited to the maximum size of the beam width. One approach to enrich the pool of compounds is to use the generated compounds as seeds for other generative models to generate similar compounds with desired properties.

Diversity versus relevance to the target

Another limitation of this approach is the ideal choice of the hyperparameters in reinforcement learning. The number of iterations and the amount of reward need to be carefully balanced. Otherwise the network may generate increasingly diverse compounds that are however no longer relevant to the target protein. In addition, pushing the model to generate highly diverse molecules might lead to the generation of compounds which are synthetically unfeasible. Tuning the hyperparameters and therefore the amount of diversity to the training set may vary based on the needs of the drug design project and the target of interest.

Generation of ligands for targets with very similar sequence

There are cases that some targets can have very similar sequences (such as different isoforms of kinases). In such cases where the targets only differ in a few residues, the embeddings generated for the target proteins will be very similar, and the molecules generated for such targets will be highly similar. The reason is that the embedding of each residue over a sequence is generated and the embeddings are averaged to gain one vector representing the whole protein, therefore, the emebeddings for those residues does not affect the resulting vector of the protein. In conclusion, the model will generate compounds relevant for a given target but will be unable to differentiate small variations in protein sequence and therefore protein-ligand interactions.

Targets with multiple binding sites

Some targets such as kinase proteins, have multiple binding sites and therefore different molecules with different structural properties may bind to the same target, although at different binding sites. For some targets, the binding site, the number of additional bindings sites or which binding site a specific compound binds to are unknown. This issue can create challenges for any drug design approach in which the 3D structure of the target, or the binding mode of the ligand is unknown. This is the case for a large portion of the experimental binding data in BindingDB. This problem could be overcome if only data for compounds with known binding site us used to train the network. In this case, it may also be possible only to use embeddings generated from the binding site sequences of the targets. The problem, however, is that currently no datasets large enough to train such a neural network are available.

Table 3.1. Various metrics measured using the MOSES framework for the generated compounds. Duplicates and invalid compounds were removed. Results for Tyrosine Kinase and GPCR targets are shown.

(a) Tyrosine Kinase

Metric	Seq2Mol	AAE	CharRNN	VAE
Fragment similarity distance	0.847	0.929	0.942	0.941
Scaffold similarity distance	0.014	0.011	0.010	0.010
Distance to the nearest neighbor	0.566	0.662	0.662	0.659
Internal diversity	0.765	0.852	0.856	0.855

(b) GPCR

Metric	Seq2Mol	AAE	CharRNN	VAE
Fragment similarity distance	0.806	0.920	0.938	0.942
Scaffold similarity distance	0.017	0.057	0.062	0.055
Distance to the nearest neighbor	0.533	0.634	0.644	0.633
Internal diversity	0.742	0.852	0.856	0.855

Table 3.2. Various metrics measurements via MOSES framework for the generated compounds after duplicates and invalid compounds are removed. Results for Tyrosine Kinase and GPCR targets are shown.

(a) Reference sets and our generated compounds corresponding to each set

Metric	Reference (Tyr Kinase)	Seq2Mol (Tyr Kinase)	Reference (GPCR)	Seq2Mol (GPCR)
LogP	3.99 ± 1.46	5.17 ± 1.75	4.50 ± 1.57	5.41 ± 1.72
Synthetic accessibility	2.93 ± 0.55	2.40 ± 0.55	3.06 ± 0.68	2.27 ± 0.49
QED	0.47 ± 0.17	0.39 ± 0.18	0.48 ± 0.19	0.38 ± 0.17
Natural product-likeness	-1.21 ± 0.55	-1.06 ± 0.43	-0.97 ± 0.64	-0.83 ± 0.50
Molecular weight	450.27 ± 85.20	471.77 ± 87.95	465.57 ± 95.74	464.33 ± 86.83

(b) Benchmark models

Metric	AAE	CharRNN	VAE
LogP	2.46 ± 1.00	2.44 ± 0.98	2.47 ± 0.94
Synthetic accessibility	2.49 ± 0.47	2.47 ± 0.47	2.43 ± 0.46
QED	0.80 ± 0.10	0.80 ± 0.10	0.81 ± 0.09
Natural product-likeness	-1.65 ± 0.59	-1.68 ± 0.64	-1.67 ± 0.64
Molecular weight	318.89 ± 30.91	308.50 ± 29.87	304.64 ± 28.65

3.4 Conclusion

In this work we have developed a method for the de novo generation of molecules based on the sequence of the target. Unlike previous works in this area, our method does not need the knowledge of already known binders to a target protein as templates for molecule generation. The sequence of the target protein is sufficient to generate target-specific molecules instead. We showed that the pool of compounds generated for two large and important protein target familes, i.e. GPCRs and Tyrosine kinases, display meaningful similarity to already known binders to these targets. With a continuous increase in number of experimentally resolved or computationally predicted protein structures, other types of protein embeddings based on 3D structure information may be used in the future, such as binding site embeddings based on 3D grids or graphs describing the binding site volume or arrangement of residues. respectively.

4. IDENTIFICATION OF REGIONS IN PROTEIN SEQUENCE PRONE TO STRUCTURAL CHANGES THROUGH DEEP NEURAL NETWORKS

4.1 Introduction

The structure of a protein is important for the function the protein and changes in the structure may be responsible for protein activation or inactivation. Such changes can be intrinsically driven [130], or be imposed by a bound ligand [131], another interacting protein, or chemical reactions such as post-translational modifications, e.g. phosphorylation [132]. Knowing regions in proteins prone to such changes is important in understanding a protein's function and for determining conformational changes associated with ligand binding. Structural flexibility is of particular importance in protein-ligand binding. Throughout the binding process proteins can take different conformations sometimes induced by the bound ligand [133]. Typically, not all accessible protein conformations have been experimentally resolved, sometimes only the apo structure is known. Identifying energetically feasible protein conformations is important for structure-based drug design methods, as methods such as docking generate inaccurate results when the incorrect conformation of a protein is used [134]. To incorporate protein flexibility in protein-ligand docking, methods such as ensemble docking [135] and induced-fit docking [136] have been developed. Whereas molecular dynamics (MD) simulations can be used to generate alternative protein structures, large conformational changes, such as alterations of the secondary structure content, are hard to obtain with those simulations [134]. Focus on specific flexible protein region may allow for accelerated sampling of alternative conformations of the protein. Therefore, it is important to be able to identify those regions in proteins which are prone to structural changes.

The structures of proteins can be categorized into structured (i.e. regular secondary structure elements such as alpha-helices and beta-sheets) and unstructured elements (i.e. lack of regular secondary structure elements including loops and disordered regions). Sometimes conformational changes in proteins are associated with structured secondary structure elements becoming disordered and vice versa, for instance in Glycogen phosphorylases, sev-

eral residues of the protein undergo order/disorder transformation during activation [137]. To understand those conformational transitions it is important to predict the propensity of protein sequence elements towards being able to form structured and unstructured configurations. Whereas multiple methods have been predicted to identify sequences which form alpha-helices or beta-sheets, and methods to predict protein disorder propensity, no method has been focused on identifying regions that are likely to form both categories dependent on external bias, here ligand binding.

Methods that predict protein disorder can be categorized into four categories [138]: physicochemical-based methods which identify disordered regions by physical principles [139]–[142], machine-learning-based which use machine learning algorithms to predict protein disorder [117], [143]–[148], template-based which uses homology models to identify disordered regions [149], [150] and meta which combine various methods [150]–[154].

Furthermore, previous knowledge-based methods were trained on X-ray crystallographic data to predict the propensity for disorder for protein sequences. Although X-ray crystal structures provide an abundant amount of information about the 3D structure of a protein, they are static in nature, do not reflect the dynamics of a protein and therefore do not provide information about the structural stability of elements of the protein. On the other hand, structures based on nucelar magnetic resonance (NMR) can provide better understanding of the protein dynamics, as the protein's structure is investigated in its solvent, unfrozen state.

To derive a predictive model for the characterization of a protein's propensity for alterations in secondary structure, we here present a deep neural network method trained on NMR data which contains structured and disordered protein elements. The model is finally tested on a series of protein system with structure elements that change between structured and disordered character due to ligand binding. Our model demonstrates its ability in those examples to identify the regions with high propensity of those structural changes.

4.2 Methods

4.2.1 Secondary structure propensity values from NMR

Tamiola et al. [130] defined a quantity, named secondary structural propensity value (SSP), that quantifies the propensity of a residue within a protein sequence region to form ordered secondary structures or being in a disordered state which can range from -1 to 1. Structural propensity values are calculated based on the difference between the observed experimental chemical shifts and the predicted shielding constants of similar intrinsically disordered proteins. The disordered chemical shift value of nucleus n of a residue a in a three residue peptide sequence $x - a - y$ is calculated as follows:

$$\delta^n_{calc}(x, a, y) = \Delta^n_p(x) + \delta^n(a) + \Delta^n_n(y) \tag{4.1}$$

where $\delta^n(a)$ is the chemical shift of residue a in a disordered sequence of $G - a - G$ where G is Glycine. $\Delta^n_p(x)$ and $\Delta^n_n(y)$ are the correction values of residues x and y, which precede and succeed residue a and n denotes the chemical shift type of atoms of the residues $n \in \left\{{}^1H^N, {}^1H^\alpha, {}^{13}C^\alpha, {}^{13}C^\beta, {}^{13}C^O, {}^{15}N\right\}$. The correction values are computed by minimizing the following expression:

$$\Delta_\epsilon = \min \begin{cases} + \mid \sum_{\mathrm{i}=1}^{N} \left(\delta^{{}^{13}C^\beta}(\mathrm{i}) - \frac{\delta^{{}^{13}C^\alpha}(\mathrm{i}) \delta^{{}^{13}C^\beta}(\mathrm{i},\alpha)}{\delta^{{}^{13}C^\alpha}(\mathrm{i},\alpha)} \right) \mid & \text{if } \delta^{{}^{13}C^\alpha}(\mathrm{i}) - \delta^{{}^{13}C^\beta}(\mathrm{i}) \geq 0 \\ + \mid \sum_{\mathrm{i}=1}^{N} \left(\delta^{{}^{13}C^\alpha}(\mathrm{i}) - \frac{\delta^{{}^{13}C^\beta}(\mathrm{i}) \delta^{{}^{13}C^\alpha}(\mathrm{i},\beta)}{\delta^{{}^{13}C^\beta}(\mathrm{i},\beta)} \right) \mid & \text{if } \delta^{{}^{13}C^\alpha}(\mathrm{i}) - \delta^{{}^{13}C^\beta}(\mathrm{i}) < 0 \end{cases} \tag{4.2}$$

where $\delta^{{}^{13}C^\alpha}(\mathrm{i}, \alpha)$ and $\delta^{{}^{13}C^\alpha}(\mathrm{i}, \beta)$ are the secondary chemical shift values in alpha helix and beta sheet structures.

The chemical shift of a residue in an ordered secondary structure element, $\delta^n(a)$, is calculated as follows:

$$\delta^n(a) = \delta^n_{exp}(a) - \delta^n_{calc}(a) \tag{4.3}$$

where $\delta^n_{exp}(a)$ is the experimental shift value from [155] .

The chemical shifts of alpha-helical or beta-sheet structures are then calculated as follows:

$$\delta^n(a, SS) = \delta^n_{SS}(a) - \delta^n_{calc}(a) \tag{4.4}$$

where the value of $\delta^n_{SS}(a)$, the average chemical shift of the residue a in a fully-formed alpha-helix or beta-sheet secondary structure and it is acquired from the chemical shift library compiled by [156].

Finally, neighbor-corrected SSP value Ψ can be defined as the following for residue at position k and neighborhood residue w:

$$\Psi(k, w) = \frac{\sum_n \sum_{\mathrm{j}=k-w}^{k+w} C\theta^n(SS)\frac{\delta^n(\mathrm{j})}{\delta^n(\mathrm{j},SS)}}{\sum_n \sum_{\mathrm{j}=k-w}^{k+w} \theta^n(SS)\frac{\delta^n(\mathrm{j},SS)}{\sigma^n(\mathrm{j},SS)}} \tag{4.5}$$

where $\delta^n(\mathrm{j})$ denotes type n secondary chemical shift value for residue at position j, $\delta^n(\mathrm{j}, SS)$ the chemical shift value for a fully formed secondary structure of type SS (alpha or beta), $\sigma(\mathrm{j}, SS)$ the standard deviation of the secondary structure SS chemical shift from the database curated by Wang et al. [156], $\theta^n(SS)$ is the parameter which indicates the relative sensitivity of the chemical shift of type n to the secondary structure type SS and is described for each chemical shift type in [130]. To discriminate between secondary structure types, constant C is used which is derived using the following equation [130]:

$$\delta^n(\mathrm{j}, SS) = \begin{cases} \delta^n(\mathrm{j}, \alpha) \wedge C = 1 & \text{if } \delta^n(\mathrm{j})\delta^n(\mathrm{j}, \alpha) > 0 \\ \delta^n(\mathrm{j}, \beta) \wedge C = -1 & \text{if } \delta^n(\mathrm{j})\delta^n(\mathrm{j}, \beta) > 0 \end{cases} \tag{4.6}$$

4.2.2 Dataset

We used 7094 protein resonance assignments from the structural propensity database of proteins [130] to train our network to predict SSP values. The data is composed of resonance values for diverse protein systems obtained in solution and solid state at near-physiological conditions. Each residue in a sequence can take a value between -1 and 1; -1 indicates the sequence with beta-strand propensity, while 1 indicates alpha-helix propensity. The value of 0 means the residue is part of a disordered region of the protein. Any other values between

these numbers represents the ratio between the folding states in the NMR structure ensemble. For instance, a value of 0.5 indicates that the specific residue is in half of the NMR ensemble part of a disordered region and in the other half it adopts an alpha-helix structure. For training and testing, the protein sequences were broken down into continuous fragments of 20 residues length. 142652 and 4834 samples were used for training and testing, respectively. The residues that did not have resonance assignments were disregarded during training for the loss calculation.

4.2.3 Network structure and training

Figure 4.1 shows the overall workflow of the method. The fragments are represented as one-hot encoding and passed through an initial embedding layer with dimensions (20,10) followed by a spatialDropout1D with a rate of 0.2. Next A 1D convolutional layer with 64 filters and kernel size of 5 and zero padding is applied, followed by a 1D MaxPooling layer with pool size of 2. Two bi-directional gated recurrent unit (GRU) [157] layers both with dimension length set to 20 were applied subsequently. Finally, a Dense layer with size of 20 is applied which generates the output values. The motivation for this choice in network architecture was that CNN layers were placed to extract the most essential features of the sequences followed by GRU units to support sequence processing. All intermediate layers used rectified linear unit (ReLU) activation. The last output layer is a dense layer that outputs SSP values between -1 and 1 (Figure 4.2). Mean squared error (MSE) was used as the loss function to optimize the weights of the network:

$$\mathrm{MSE} = \frac{1}{n}\sum_{\mathrm{i}=1}^{n}\left(y_{\mathrm{i}} - \tilde{y}_{\mathrm{i}}\right)^2 \tag{4.7}$$

Where y and $\tilde{y}$ are experimental and predicted values of the dataset, and n is the number of samples in the batch. The network was trained for 200 epochs with batch size set to 2048.

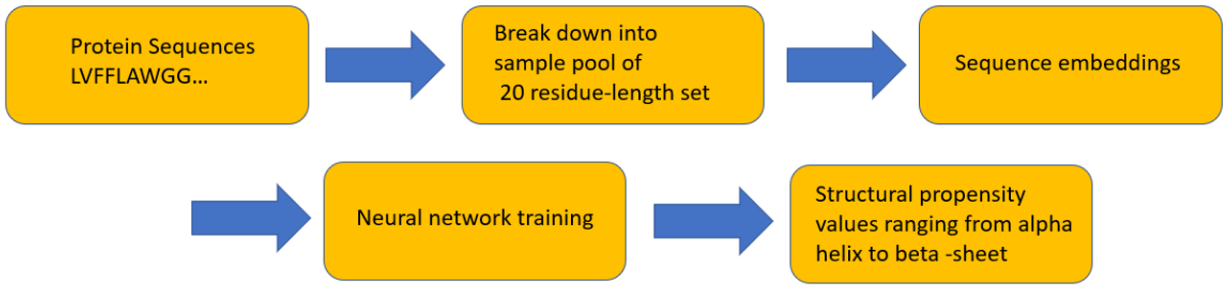


Figure 4.1. Overview of method for prediction of structural propensities from sequence. The sequence is first broken down into 20-residues long fragments. After featurization and embedding, subsequent neural network layers predict propensity values.

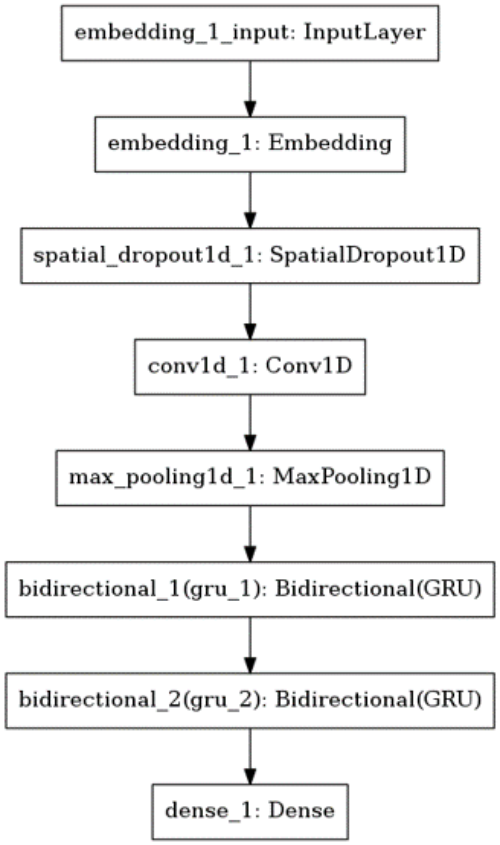


Figure 4.2. The neural network architecture used in our approach for the prediction of structural propensities.

4.2.4 Case studies

Six protein systems crystallized in different conformations were chosen to exemplify our method's potential to study regions with propensity for structural changes: Heat shock protein 90 (HSP90, PDB: 5j9x,1yet,5j64), chemosensory Protein (PDB: 1kx9, 1n8v), PLP-dependent acyl-CoA synthase (PDB: 1bs0, 1dj9), beta-1,4-galactosyltransferase 1 (PDB: 1pzt, 1o0r), dehydrosqualene synthase (PDB: 2zco, 2zcq), and lipase A. (PDB: 1i6w,1r4z) All of the mentioned structures except HSP90 were acquired from the Protein Structural Change DataBase (PSCDB) [158].

4.3 Results

4.3.1 Network performance

The neural network achieved a mean absolute error of 0.35 for the training and 0.38 for the validation set. Figure 4.3 shows the error convergence plots for the model. In the following section, we demonstrate the prediction performance in individual cases of proteins, which show duality in secondary structures as captured by x-ray crystallography.

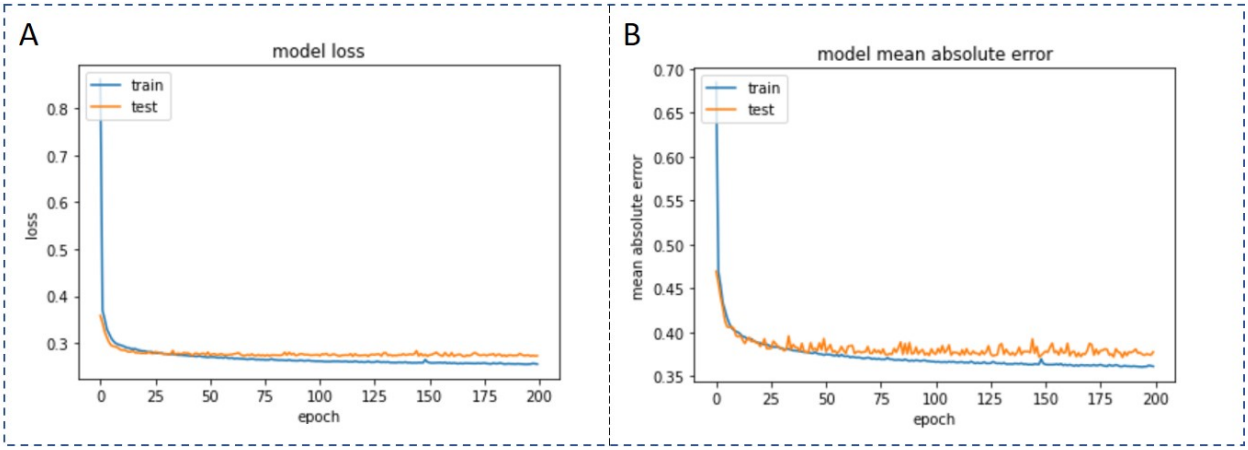


Figure 4.3. The model's error convergence plots A) Model's loss (MSE) B) Mean absolute error (MAE) after 200 epochs.

4.3.2 Case studies

Heat shock protein 90 (HSP 90)

Figure 4.4 shows three different conformations of HSP 90 induced or stabilized by three different bound ligands. The protein region colored in green adopts three different conformations: (A) Complete helix when interacting with N-Butyl-5-[4-(2-fluoro-phenyl)-5-oxo-4,5-dihydro-1H-[1,2,4]triazol-3-yl]-2,4-dihydroxy-N-methyl-benzamide, (B) loop-out (interaction with Geldanamycin) and (C) loop-in (interaction with 5-(2,4-Dihydroxy-phenyl)-4-(2-fluoro-phenyl)-2,4-dihydro-[1,2,4]triazol-3-one). The structure propensity prediction of the green region reveals a largely alpha-helical propensity at both ends of this region and both disordered and helical propensity around residue Lys-112 in the center of the region (Figure 3 (D)). As the different co-crystal structures demonstrate this region can indeed form helical secondary structure as well as disordered conformations depending on the type of the bound ligand.

Chemosensory Protein

The N-terminal region of the chemosensory protein forms an alpha-helix in the unbound, apo conformation of the protein. Upon ligand binding the helix partially unfolds and assumes a disordered state. In one conformation the helix spans across residues 4-20, on the other hand in another conformation, the helix half of these residues are disordered. Our method predicts that the residues Glu-1 to Asn-12 have low propensity for alpha-helix, but the other half (Leu-13 to Lys-19) is predicted to have a high alpha-helix propensity. This is in agreement with the crystallized structures as shown in Figure 4.5A and B. The SSP values of the residues Glu-1 to Asn-12 represents the lower fraction of conformations in which these residues are in alpha-helix, thus rather disordered, compared to the disorder values from Leu-13 to Lys-19 which are higher, therefore exhibit alpha-helix in higher fraction of conformations (Figure 4.5C)

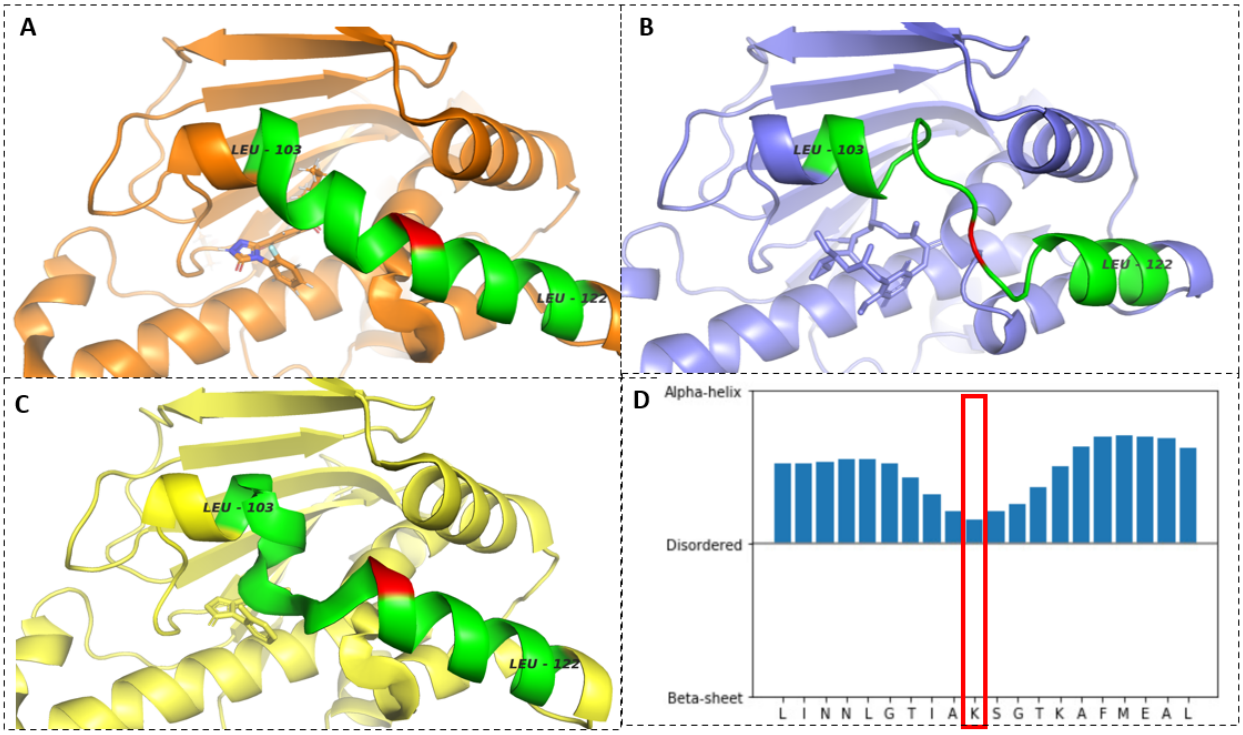


Figure 4.4. Three ligand bound conformations (A,B and C) of HSP-90 (PDB codes 5j9x, 1yet,and 5j64, respectively) and our SSP prediction of the 20 residue-length sub-sequence (D) between Leu-103 and Leu-122. Prediction for Lys-112 residue shows the lowest helical propensity and it is revealed that this region can both assume helical and disordered structure character.

PLP-dependent acyl-CoA synthase

The selected region in PLP-dependent acyl-CoA synthase (Figure 4.6, green) shows a conformational transition between a beta-strand structure to a loop in region (Pro-322 to Asn-328), with subsequent alpa-helix. This transition propensity between loop and beta-sheet is clearly reproduced by our prediction (Figure 4.6C). The interesting observation is that the values predicted for the beta-strand part have a low magnitude, indicating both forms of beta-strand and disordered are possible, which is observed in the two structures (Figure 4.6).

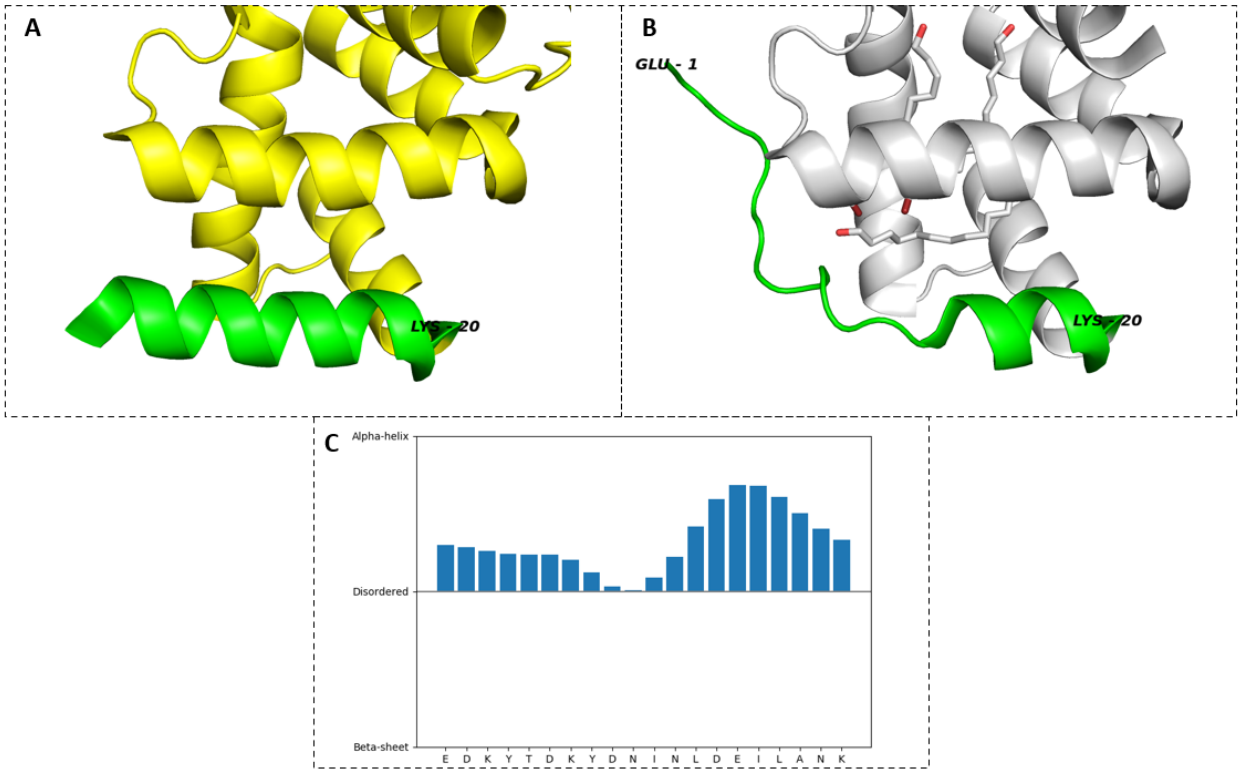


Figure 4.5. Ligand-free (A) and ligand-bound (B) conformation of chemosensory protein (PDB codes 1kx9 and 1nv8, respectively) and propensity prediction of the 20 residue-long N-terminal region (C), Glu-1 to Lys-20. Note that part of the disordered region is not crystallized in A.

Beta-1,4-galactosyltransferase 1

The region of interest in beta-1,4-galactosyltransferase 1 is a bent helix (Figure 4.7B), half of which is found as a loop in another structure (Figure 4.7A). The whole selected region takes a low alpha-helix propensity values (between 0 and 0.5), predicting the existence of both conformations in which this region exhibit alpha-helix and in which it is a disordered region. The crystal structures shown in Figure 4.7A and B confirms the prediction; in one structure (Figure 4.7A) residues Asn-356 to Ala-364 lose the helix structure and in another structure (Figure 6B) residues Asn-356 to Ala-364 take a bent form, not maintaining a full helix structure. (Figure 4.7).

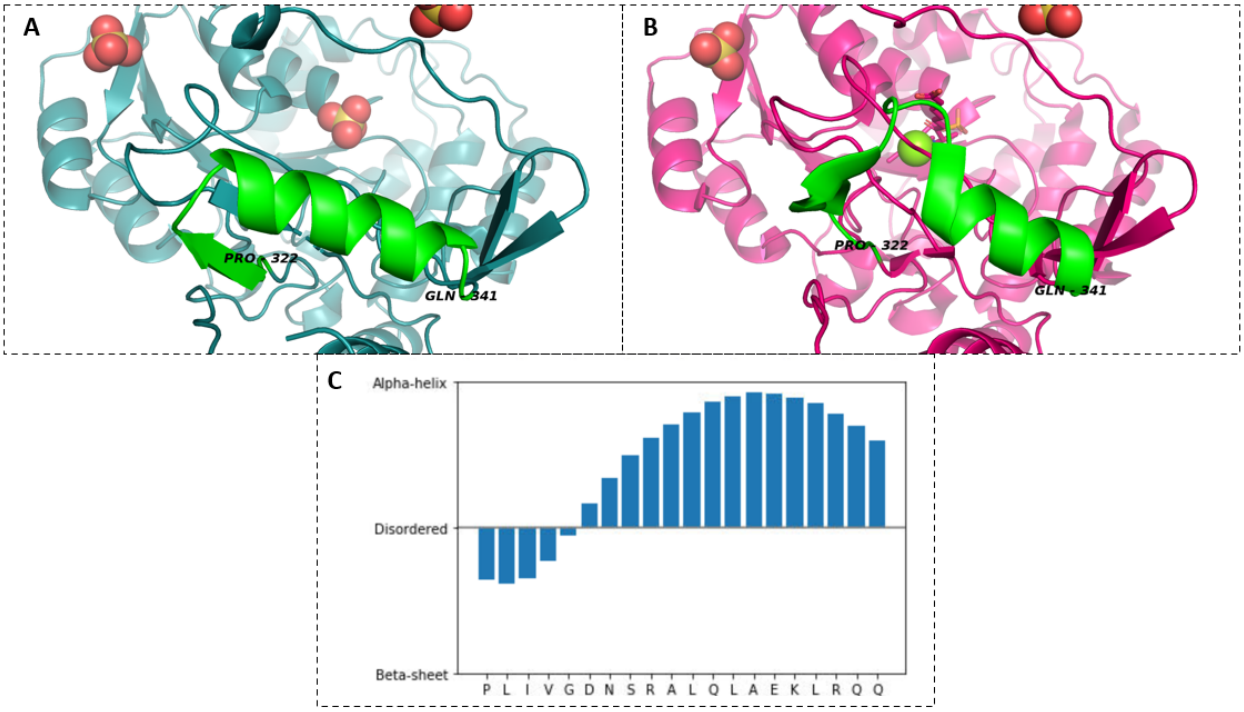


Figure 4.6. Ligand-free (A) and ligand-bound (B) conformations of PLP-dependent acyl-CoA synthase (PDB codes 1bs0 and 1dj9 respectively) and the propensity prediction of the 20 residue-long region Pro-322 to Gln-341 (C).

Dehydrosqualene synthase

The selected region in dehydrosqualene synthase (Figure 4.8) shows a transition between helix and loop. The helix structure can become disordered in one conformation (Figure 4.8A) which is clearly in agreement with the disorder prediction, where the helix propensity falls close to the loop and the residues immediately before the loop (D49-D52) have a low helix propensity. Although the values assume small negative values for the loop region this propensity for beta-strand character is very low.

Lipase A

Figure 4.9 shows a region of lipase A protein that consists of a disordered and a helix structures. The prediction (Figure 8C) shows a beta-sheet propensity, however with a low magnitude, which indicates a high propensity for a disordered structure. The second part of

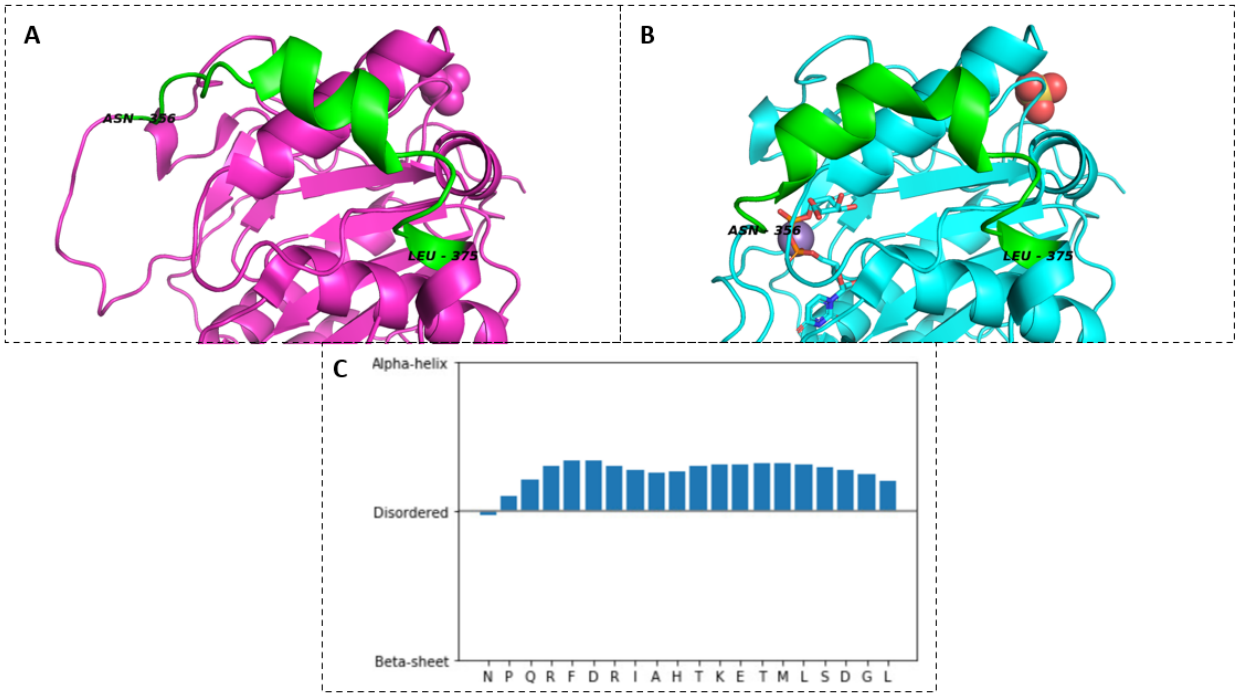


Figure 4.7. Ligand-free (A) and ligand-bound (B) conformations of beta-1,4-galactosyltransferase 1 (PDB codes 1pzt and 1oOr respectively) and the propensity prediction of the 20 residue-long region (C), ranging from Asn-356 to Leu-375.

this region (Phe-19 to Ser-28) has propensity for alpha-helix character. The values, however, are also relatively low indicating likely instability of the helix. That coincides with the observed bend in the helix in Figure 8B.

4.4 Conclusion

In this study, we showed how protein disorder data derived from NMR ensembles can be used to train predictors to detect regions of proteins susceptible to structural change and likely to show dual order-disorder character. We developed a deep neural network method which uses only the sequence as features and uses them to predict propensity of protein regions to form structured secondary or disordered structures. We tested our model in a number of cases of apo and holo conformations of proteins, which shows excellent agreement between predicted and experimental X-ray structure data. We believe our method can have applications in focusing enhanced sampling techniques based on collective variables, e.g.

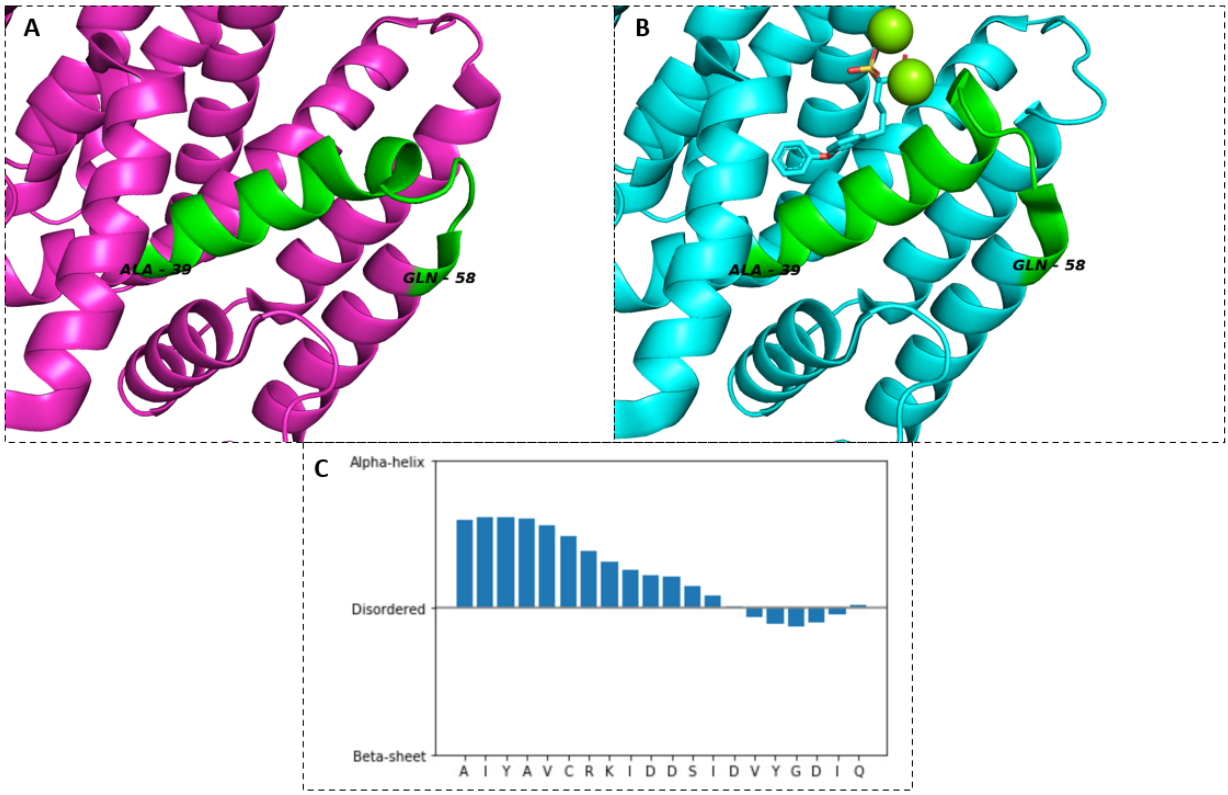


Figure 4.8. Ligand-free (A) and ligand-bound (B) conformations of dehydrosqualene synthase (PDB codes 2zco and 2zcq respectively) and the propensity prediction of the 20 residue-llong sub-sequence (C), ranging from Ala-39 to Gln-58.

metadynamics [159], to enforce conformational sampling of flexible proteins. Furthermore, the method may be utilized to select regions to be treated as flexible in flexible protein docking methods. In future, the method may be improved by adding more curated features, such as evolutionary information or physicochemical properties of amino acids in addition to sequences to achieve better prediction accuracy.

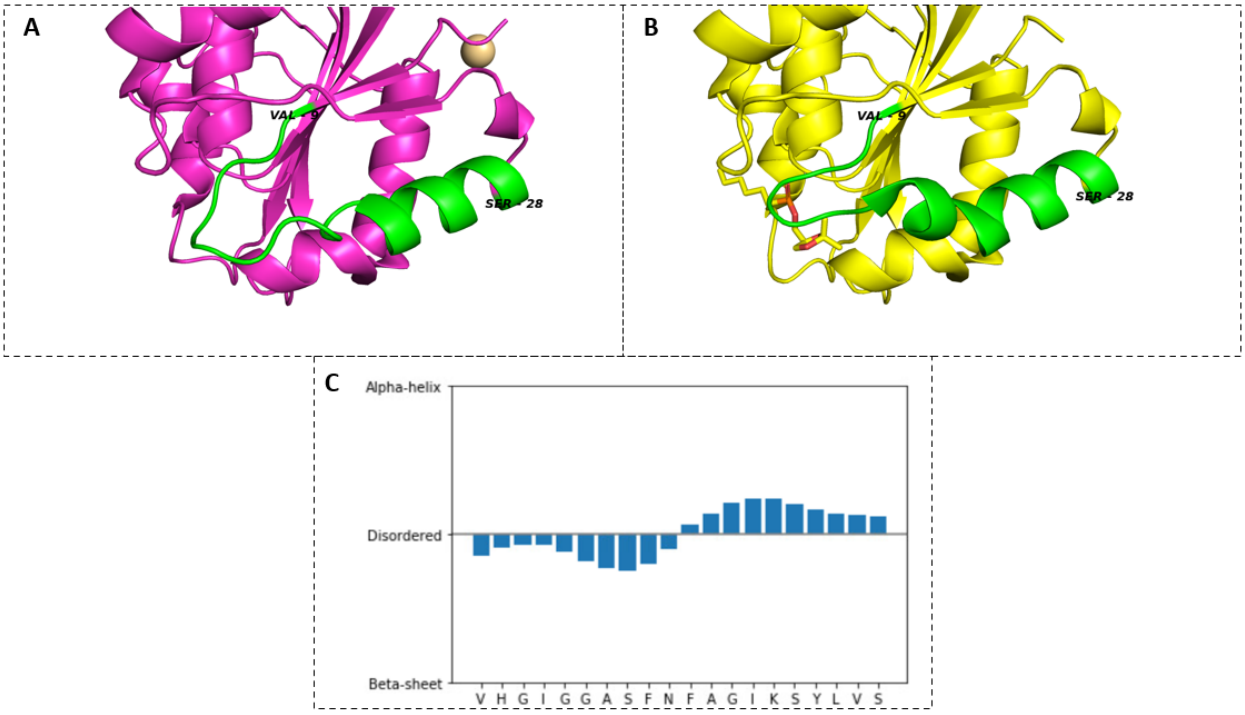


Figure 4.9. Ligand-free (A) and ligand-bound (B) conformations of Lipase A (PDB codes 1i6w and 1r4z respectively) and the propensity prediction of the 20 residue-long sub-sequence (C), ranging from Val-9 to Ser-28.

5. FUTURE DIRECTIONS

In this chapter, I will discuss current limitations, potential improvements and future applications of the three methods developed throughout my thesis research.

5.1 Prediction of protein's hydration properties

5.1.1 Possible future improvements in methodology

Training data

In this study, described in chapter 2, data used for training the neural networks were generated by WATsite. Although this strategy made training data generation accessible and cheaper compared to experimental data, it should be noted that training on MD-generated data allows the model to perform as good as the MD method at best. The possibility that neural network models can even outperform MD-based predictions using experimental data is yet to be explored. As crystallographic hydration data is sparse, augmentation with MD-generated data should be considered. Furthermore, the models presented in this thesis aim to generate hydration density maps on a 3D grid, generated by WATsite. Crystallographic data on the other hand display hydration sites as points without differences in occupancy values. For that reason, occupancy data from WATsite may first have to be represented as hydration sites to be mixed with crytallographic data. A model can then be trained to predict the hydration sites as points, end-to-end, that is predicting hydration sites coordinates from atomic coordinates of the protein as input, by representing the data as point cloud data or graph using model concepts such as transformers [160].

Data representations

In chapter 2, it was shown how convolutional neural networks can be used to map molecular interaction fields grids to water occupancy grids. This approach seems reasonable because the output data which the model was to predict was also represented as a 3D grid, making the task a grid-to-grid mapping problem. However, as also mentioned in the same chapter, the sparsity of high water occupancy regions was a great challenge to overcome. A loss

function designed for sparse data was used in the fully-convolutional approach to address the issue and it was successful to some extent. Another issue with the grid-to-grid mapping approach was the fixed size of the grid, that is inherent with the convolution approach. The fixed size of the grids may not contain information about long range effects of atoms and water networks falling outside the grid, therefore not capturing the full picture and making the model perform less optimally.

Another problem of the grid-based approach is that generating hydration sites based on the predicted occupancy grids will depend heavily on the clustering algorithm used and depend on the hyper parameters of the clustering algorithm. This adds another layer of hyper-parameter optimization for the clustering algorithm to the overall prediction task. That is another reason why future models should aim to generate hydration sites end-to-end, without the need to use grids.

In the second approach described in chapter 2, spherical harmonics expansion was used to generate descriptors for a fully-connected network. The design was conceived such that the output of the network would predict the occupancy/thermodynamic properties of a point on the protein's surface while also receiving information regarding the surroundings by features generated through spherical harmonics expansion. This approached removes the sparsity problem, by allowing one to use a more balanced dataset of occupancy values, so that the low occupancy values do not severely dominate the dataset. This approach however, relies more on feature engineering, since the parameters for generating descriptors need to be chosen by the user. Also, the output of the model is used to construct the occupancy grids which again will require designing an optimal clustering algorithm.

Architecture improvements

As mentioned in the previous sections, moving away from voxelized representations and subsequently 3D convolutions is a possible direction to move forward with the project. In addition moving away from image-like representations may improve training time and convergence, since instead of having to process so many redundant voxels, only atoms represented as nodes of a graph will be processed.. Using voxelized representations of protein

data has a few issues: It does not contain any bond information between atoms. Also, it may be inefficient, since many voxels are generated to cover a few number of atoms, and the convolution operation becomes expensive as the dimensions of the data increase. In addition, orientations and transformations may negatively impact the learning and prediction by CNN models. Furthermore, our current model's prediction quality at the corners of the grid is worse compared to points in the center of the grid. Using graph-based or point-cloud based approaches may be a more natural representation of the underlying protein data and therefore may result in improved prediction quality.

Addressing descriptor generation overhead

Whereas the neural network models require little time to predict the hydration data, generating the descriptors is currently done by sequential programs and thus rather time consuming. While this is more of a technical issue rather than methodological, it is important to address the issue if the method is going to be used in downstream tasks such as docking applications. Code optimizations and parallel programming are viable options for faster descriptor generation.

Inclusion of the ligand

The methods presented in this thesis focus on predicting hydration properties in apo protein structures. This has been the approach in most hydration site prediction applications, since generating accurate thermodynamic properties is computationally too expensive to be performed for a large number of protein-ligand pairs. Training the models with ligand information can lead to interesting findings, such as how ligands can affect the hydration on protein surfaces and enhance our understandings of ligand-water-protein interactions, ultimately leading to more optimal methods in drug discovery.

5.1.2 Potential applications

As mentioned in chapter 2, instantaneous generation of thermodynamic properties of hydration sites allows the integration of explicit (de)solvation in scoring functions for dock-

ing application, possibly resulting in better performance. Whereas most scoring methods rely on simplified typically implicit models of hydration, our lab recently developed an artificial intelligence model to integrate explicit (de)solvation into scoring function [70]. In this method, the scoring method relies on a single conformation of the protein and generates hydration properties based on it. Although the hydration properties are the outcome of molecular simulations, the protein's backbone is usually restrained in these methods to achieve convergence in hydration site profiling. Such inherent rigidity also will transfer to the docking process. However, with our method, one could sample different conformations of proteins during an MD simulation and use the conformations sampled for ensemble docking and generate hydration properties instantly for all conformations. Instant generation of hydration properties could be used in reverse virtual screening, where one or a small number of compounds are screened against a library of protein targets. In this case, instead of having to run a separate MD simulation for each target, our method can be used to generate the data in a reasonable time. Our method can also be used in scoring poses in induced-fit docking, assuming the overhead issue in descriptor generation is addressed.

5.2 Target-based generation of de novo molecules

5.2.1 Possible future improvements in methodology

Data embeddings

As mentioned in Chapter 3, there are limitations in using sequences as protein features for target-specific compound generation. Whereas sequences determine the protein structure, using them directly as features for the protien target in our model lacks a 3D representation of the binding site. Whereas some structural information can be inferred by neural network models from the sequence, directly adding structural information to the embeddings or using structural embeddings may improve the quality of the prediction. Similar to text embeddings, structures, whether represented as graphs or voxels, can yield embeddings by using proper models such as autoencoders [161] or transformers [160]. The challenge in this regards, however, is the limited size of structral data available. Experimental structures of protein binding sites exist only for a fraction of all the targets in data sets. To partially over-

come this limitation, transfer learning may be used, where the network learns the structural features of proteins for one task such as binding site classification, or structure reconstruction and then the layer outputs are used as embeddings for another task, which in this case is molecule generation.

In our study, we used embeddings of the whole protein sequence by computing the average over all residue embeddings. It is, however, known that the residues of the binding pocket matter most in the context of protein-ligand interaction. Averaging over all protein residues therefore may ”blur” the information of the key residues for protein-ligand binding. Again, this issue is caused by the limitation in data. The 3D structures of many of the targets in the dataset are not known, therefore no information about the interacting residues with ligand are available. However, the binding residues could be ”guessed” by methods that detect binding pockets. Such methods could be used to select residues to generate embeddings from, with the non-important residues being masked.

Molecule representations

As mentioned in chapter 1 and 3, representing chemical data as SMILES strings makes it easy to use models designed for NLP to process the data. However, the SMILES representation or similar ones are very sensitive to typing errors and missing characters in the string, therefore a large fraction of the generated output can be invonvertible to a molecular structure. More recent approaches are trying to replace SMILES representation with graph representations for do novo compound generation. Graph-based approaches may be of investigated in our method as well.

Architecture improvements

The approach described for molecule generation involves utilizing two different models. ELMo model is used to generate the contextual embeddings of the protein sequences, and a LSTM decoder then is used to generate sequences relevant to the embeddings. Newer models have been developed for generating contextual embeddings such as BERT [38] which uses

transformer model architecture and attention mechanism along with other improvements to generate embeddings.

Overall, any architecture modification will depend on the data representations, for example, using graph data to represent the protein and/or the molecules will demand a different architecture than LSTM cells which are designed to process sequential data.

5.3 Prediction of protein disorder through DNNs

5.3.1 Possible future improvements in methodology

Architecture improvements

The architecture of this model can be further improved by using state-of-the-art language models such as BERT, pre-trained on large datasets. This way more accurate embeddings can be generated for a sequence and performance may increase.

Model benchmarks and comparisons

The current results discussed in chapter 3 suggest the utility of the model to qualitatively sense regions prone to structural changes. However, more investigations are needed for this method to understand its advantages and limitations. A comparison with other prediction methods would be beneficial. Many prediction methods rely on x-ray crystallographic data for training and building models, a comparison between the models based on crystallographic data and our model which is trained on NMR data would be of interest.

5.3.2 Potential applications

Prediction of protein disorder has been the interest of the scientific community. In the context of drug design, knowing the correct conformation(s) of proteins is important in structure-based drug design. The current method can be used for directing flexible docking applications and reducing sampling space by focusing on the regions more likely to disorder. Also, it can be used to bias protein modeling programs for considering non-template

conformations. It can be helpful in protein design, where a certain secondary structure is desired.

REFERENCES

[1] L. P. Hammett, "The effect of structure upon the reactions of organic compounds. benzene derivatives," *J. Am. Chem. Soc.*, vol. 59, no. 1, pp. 96–103, 1937.

[2] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[3] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, IEEE, vol. 1, 1995, pp. 278–282.

[4] J. A. Hertz, *Introduction to the theory of neural computation*. CRC Press, 2018.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[6] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv e-prints*, arXiv:1412.6980, arXiv:1412.6980, Dec. 2014. arXiv: 1412.6980 [cs.LG].

[7] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE international symposium on circuits and systems*, IEEE, 2010, pp. 253–256.

[8] H. Lu, Y. Li, T. Uemura, Z. Ge, X. Xu, L. He, S. Serikawa, and H. Kim, "Fdcnet: Filtering deep convolutional network for marine organism classification," *Multimedia tools and applications*, vol. 77, no. 17, pp. 21 847–21 860, 2018.

[9] J. F. Kolen and S. C. Kremer, *A field guide to dynamical recurrent networks*. John Wiley & Sons, 2001.

[10] J. Schmidhuber and S. Hochreiter, "Long short-term memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[12] F. Shaheen, B. Verma, and M. Asafuddoula, "Impact of automatic feature extraction in deep learning architecture," in *2016 International conference on digital image computing: techniques and applications (DICTA)*, IEEE, 2016, pp. 1–8.

[13] M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri, and D. R. Koes, "Protein-ligand scoring with convolutional neural networks," *J. Chem. Inf. Model.*, vol. 57, no. 4, pp. 942–957, 2017, PMID: 28368587. DOI: 10.1021/acs.jcim.6b00740. eprint: https://doi.org/10.1021/acs.jcim.6b00740. [Online]. Available: https://doi.org/10.1021/acs.jcim.6b00740.

[14] M. Simonovsky and J. Meyers, "Deeplytough: Learning structural comparison of protein binding sites," *Journal of Chemical Information and Modeling*, vol. 60, no. 4, pp. 2356–2366, 2020.

[15] J. Jiménez, M. Škalič, G. Martínez-Rosell, and G. D. Fabritiis, "KDEEP: Protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks," *J. Chem. Inf. Model.*, vol. 58, no. 2, pp. 287–296, Jan. 2018, ISSN: 1549-9596. DOI: 10.1021/acs.jcim.7b00650. [Online]. Available: https://doi.org/10.1021/acs.jcim.7b00650.

[16] J. Jiménez, S. Doerr, G. Martínez-Rosell, A. S. Rose, and G. De Fabritiis, "Deepsite: Protein-binding site predictor using 3d-convolutional neural networks," *Bioinformatics*, vol. 33, no. 19, pp. 3036–3042, 2017.

[17] L. Pu, R. G. Govindaraj, J. M. Lemoine, H.-C. Wu, and M. Brylinski, "Deepdrug3d: Classification of ligand-binding pockets in proteins with a convolutional neural network," *PLOS Computational Biology*, vol. 15, no. 2, pp. 1–23, Feb. 2019. DOI: 10.1371/journal.pcbi.1006718. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1006718.

[18] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, no. suppl_1, pp. i47–i56, 2005.

[19] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems*, 2015, pp. 2224–2232.

[20] A. Strokach, D. Becerra, C. Corbi-Verge, A. Perez-Riba, and P. M. Kim, "Fast and flexible protein design using deep graph neural networks," *Cell Systems*, vol. 11, no. 4, pp. 402–411, 2020.

[21] R. Salakhutdinov, "Learning deep generative models," *Annual Review of Statistics and Its Application*, vol. 2, pp. 361–385, 2015.

[22] N. Anand and P. Huang, "Generative modeling for protein structures," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc., 2018, pp. 7494–7505. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/afa299a4d1d8c52e75dd8a24c3ce534f-Paper.pdf.

[23] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[24] D. Merk, L. Friedrich, F. Grisoni, and G. Schneider, "De novo design of bioactive small molecules by artificial intelligence," *Molecular informatics*, vol. 37, no. 1-2, p. 1 700 153, 2018.

[25] A. Gupta, A. T. Müller, B. J. Huisman, J. A. Fuchs, P. Schneider, and G. Schneider, "Generative recurrent networks for de novo drug design," *Molecular informatics*, vol. 37, no. 1-2, p. 1 700 111, 2018.

[26] M. H. Segler, T. Kogej, C. Tyrchan, and M. P. Waller, "Generating focused molecule libraries for drug discovery with recurrent neural networks," *ACS central science*, vol. 4, no. 1, pp. 120–131, 2018.

[27] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, 2. MIT press Cambridge, 2016, vol. 1.

[28] A. Kadurin, A. Aliper, A. Kazennov, P. Mamoshina, Q. Vanhaelen, K. Khrabrov, and A. Zhavoronkov, "The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology," *Oncotarget*, vol. 8, no. 7, p. 10 883, 2017.

[29] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[30] J. Yang, T. Li, G. Liang, W. He, and Y. Zhao, "A simple recurrent unit model based intrusion detection system with dcgan," *IEEE Access*, vol. 7, pp. 83 286–83 296, 2019.

[31] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[32] N. De Cao and T. Kipf, "Molgan: An implicit generative model for small molecular graphs," *arXiv preprint arXiv:1805.11973*, 2018.

[33] B. Sanchez-Lengeling, C. Outeiral, G. L. Guimaraes, and A. Aspuru-Guzik, *Optimizing distributions over molecular space. an objective-reinforced generative adversarial network for inverse-design chemistry (organic)*, Aug. 2017. DOI: 10.26434/chemrxiv.5309668.v3. [Online]. Available: https://chemrxiv.org/articles/preprint/ORGANIC_1_pdf/5309668/3.

[34] G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias, and A. Aspuru-Guzik, "Objective-reinforced generative adversarial networks (organ) for sequence generation models," *arXiv preprint arXiv:1705.10843*, 2017.

[35] M. Olivecrona, T. Blaschke, O. Engkvist, and H. Chen, "Molecular de novo design through deep reinforcement learning," *CoRR*, vol. abs/1704.07555, 2017. arXiv: 1704.07555. [Online]. Available: http://arxiv.org/abs/1704.07555.

[36] E. Putin, A. Asadulaev, Y. Ivanenkov, V. Aladinskiy, B. Sanchez-Lengeling, A. Aspuru-Guzik, and A. Zhavoronkov, "Reinforced adversarial neural computer for de novo molecular design," *Journal of chemical information and modeling*, vol. 58, no. 6, pp. 1194–1204, 2018.

[37] A. Chattopadhyay, P. Manupriya, A. Sarkar, and V. N. Balasubramanian, "Neural network attributions: A causal perspective," *arXiv preprint arXiv:1902.02302*, 2019.

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[39] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[40] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

[41] P.-j. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim, "The (un)reliability of saliency methods," 2017. [Online]. Available: https://arxiv.org/pdf/1711.00867.pdf.

[42] F. Noé, S. Olsson, J. Köhler, and H. Wu, "Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning," *Science*, vol. 365, no. 6457, eaaw1147, Sep. 2019. DOI: 10.1126/science.aaw1147. [Online]. Available: http://science.sciencemag.org/content/365/6457/eaaw1147.abstract.

[43] F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi, "Machine Learning for Molecular Simulation," *Annual Review of Physical Chemistry*, Feb. 2020, ISSN: 0066-426X. DOI: 10.1146/annurev-physchem-042018-052331. [Online]. Available: https://doi.org/10.1146/annurev-physchem-042018-052331.

[44] C. Wehmeyer and F. Noé, "Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics," *The Journal of Chemical Physics*, vol. 148, no. 24, p. 241 703, Mar. 2018, ISSN: 0021-9606. DOI: 10.1063/1.5011399. [Online]. Available: https://doi.org/10.1063/1.5011399.

[45] Y. Wang, J. M. L. Ribeiro, and P. Tiwary, "Past–future information bottleneck for sampling molecular reaction coordinate simultaneously with thermodynamics and kinetics," *Nature Communications*, vol. 10, no. 1, p. 3573, 2019, ISSN: 2041-1723. DOI: 10.1038/s41467-019-11405-4. [Online]. Available: https://doi.org/10.1038/s41467-019-11405-4.

[46] Z. Shamsi, K. J. Cheng, and D. Shukla, "Reinforcement Learning Based Adaptive Sampling: REAPing Rewards by Exploring Protein Conformational Landscapes," *The Journal of Physical Chemistry B*, vol. 122, no. 35, pp. 8386–8395, Sep. 2018, ISSN: 1520-6106. DOI: 10.1021/acs.jpcb.8b06521. [Online]. Available: https://doi.org/10.1021/acs.jpcb.8b06521.

[47] M. T. Degiacomi, "Coupling Molecular Dynamics and Deep Learning to Mine Protein Conformational Space," *Structure*, vol. 27, no. 6, 1034–1040.e3, 2019, ISSN: 0969-2126. DOI: https://doi.org/10.1016/j.str.2019.03.018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0969212619301145.

[48] W. Chen and A. L. Ferguson, "Molecular enhanced sampling with autoencoders: On-the-fly collective variable discovery and accelerated free energy landscape exploration," *J. Comput. Chem.*, vol. 39, no. 25, pp. 2079–2102, Sep. 2018, ISSN: 0192-8651. DOI: 10.1002/jcc.25520. [Online]. Available: https://doi.org/10.1002/jcc.25520.

[49] H. Jung, R. Covino, and G. Hummer, "Artificial Intelligence Assists Discovery of Reaction Coordinates and Mechanisms from Molecular Dynamics Simulations," Jan. 2019. arXiv: 1901.04595. [Online]. Available: http://arxiv.org/abs/1901.04595.

[50] E. Nittinger, F. Flachsenberg, S. Bietz, G. Lange, R. Klein, and M. Rarey, "Placement of water molecules in protein structures: From large-scale evaluations to single-case examples," *J. Chem. Inf. Model.*, vol. 58, no. 8, pp. 1625–1637, 2018, PMID: 30036062. DOI: 10.1021/acs.jcim.8b00271. eprint: https://doi.org/10.1021/acs.jcim.8b00271. [Online]. Available: https://doi.org/10.1021/acs.jcim.8b00271.

[51] G. A. Ross, G. M. Morris, and P. C. Biggin, "Rapid and accurate prediction and scoring of water molecules in protein binding sites," *PLOS ONE*, vol. 7, no. 3, pp. 1–13, Mar. 2012. DOI: 10.1371/journal.pone.0032036. [Online]. Available: https://doi.org/10.1371/journal.pone.0032036.

[52] G. Rossato, B. Ernst, A. Vedani, and M. Smieško, "Acquaalta: A directional approach to the solvation of ligand-protein complexes," *J. Chem. Inf. Model.*, vol. 51, no. 8, pp. 1867–1881, 2011, PMID: 21714532. DOI: 10.1021/ci200150p. eprint: https://doi.org/10.1021/ci200150p. [Online]. Available: https://doi.org/10.1021/ci200150p.

[53] A. Kovalenko and F. Hirata, "Three-dimensional density profiles of water in contact with a solute of arbitrary shape: A rism approach," *Chem. Phys. Lett.*, vol. 290, no. 1, pp. 237–244, 1998, ISSN: 0009-2614. DOI: https://doi.org/10.1016/S0009-2614(98)00471-0. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0009261498004710.

[54] A. S. Bayden, D. T. Moustakas, D. Joseph-McCarthy, and M. L. Lamb, "Evaluating free energies of binding and conservation of crystallographic waters using szmap," *J. Chem. Inf. Model.*, vol. 55, no. 8, pp. 1552–1565, 2015, PMID: 26176600. DOI: 10.1021/ci500746d. eprint: https://doi.org/10.1021/ci500746d. [Online]. Available: https://doi.org/10.1021/ci500746d.

[55] G. A. Ross, M. S. Bodnarchuk, and J. W. Essex, "Water sites, networks, and free energies with grand canonical monte carlo," *J. Am. Chem. Soc.*, vol. 137, no. 47, pp. 14 930–14 943, 2015, PMID: 26509924. DOI: 10.1021/jacs.5b07940. eprint: https://doi.org/10.1021/jacs.5b07940. [Online]. Available: https://doi.org/10.1021/jacs.5b07940.

[56] E. D. López, J. P. Arcon, D. F. Gauto, A. A. Petruk, C. P. Modenutti, V. G. Dumas, M. A. Marti, and A. G. Turjanski, "WATCLUST: a tool for improving the design of drugs based on protein-water interactions," *Bioinformatics*, vol. 31, no. 22, pp. 3697–3699, Jul. 2015, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btv411. eprint: https://academic.oup.com/bioinformatics/article-pdf/31/22/3697/5026765/btv411.pdf. [Online]. Available: https://doi.org/10.1093/bioinformatics/btv411.

[57] T. Young, R. Abel, B. Kim, B. J. Berne, and R. A. Friesner, "Motifs for molecular recognition exploiting hydrophobic enclosure in protein–ligand binding," *PNAS*, vol. 104, no. 3, pp. 808–813, 2007, ISSN: 0027-8424. DOI: 10.1073/pnas.0610202104. eprint: https://www.pnas.org/content/104/3/808.full.pdf. [Online]. Available: https://www.pnas.org/content/104/3/808.

[58] R. Abel, T. Young, R. Farid, B. J. Berne, and R. A. Friesner, "Role of the active-site solvent in the thermodynamics of factor xa ligand binding," *J. Am. Chem. Soc.*, vol. 130, no. 9, pp. 2817–2831, 2008, PMID: 18266362. DOI: 10.1021/ja0771033. eprint: https://doi.org/10.1021/ja0771033. [Online]. Available: https://doi.org/10.1021/ja0771033.

[59] B. Hu and M. A. Lill, "Watsite: Hydration site prediction program with pymol interface," *J. Comput. Chem.*, vol. 35, no. 16, pp. 1255–1260, 2014. DOI: 10.1002/jcc.23616. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.23616. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.23616.

[60] Y. Yang, B. Hu, and M. A. Lill, "Watsite2.0 with pymol plugin: Hydration site prediction and visualization," in *Methods Mol. Biol. (N.Y., NY, U.S.)* Springer, 2017, pp. 123–134. DOI: 10.1007/978-1-4939-7015-5_10.

[61] M. R. Masters, Y. Y. Mahmoud Amr H., and M. A. Lill, "Efficient and accurate hydration site profiling for enclosed binding sites," *J. Chem. Inf. Model.*, vol. 58, no. 11, pp. 2183–2188, 2018, PMID: 30289252. DOI: 10.1021/acs.jcim.8b00544. eprint: https://doi.org/10.1021/acs.jcim.8b00544. [Online]. Available: https://doi.org/10.1021/acs.jcim.8b00544.

[62] D. Bucher, P. Stouten, and N. Triballeau, "Shedding light on important waters for drug design: Simulations versus grid-based methods," *J. Chem. Inf. Model.*, vol. 58, no. 3, pp. 692–699, 2018, PMID: 29489352. DOI: 10.1021/acs.jcim.7b00642. eprint: https://doi.org/10.1021/acs.jcim.7b00642. [Online]. Available: https://doi.org/10.1021/acs.jcim.7b00642.

[63] R. Abel, N. K. Salam, J. Shelley, R. Farid, R. A. Friesner, and W. Sherman, "Contribution of explicit solvent effects to the binding affinity of small-molecule inhibitors in blood coagulation factor serine proteases," *ChemMedChem*, vol. 6, no. 6, pp. 1049–1066, 2011. DOI: 10.1002/cmdc.201000533. eprint: https://chemistry-europe.onlinelibrary.wiley.com/doi/pdf/10.1002/cmdc.201000533. [Online]. Available: https://chemistry-europe.onlinelibrary.wiley.com/doi/abs/10.1002/cmdc.201000533.

[64] C. Higgs, T. Beuming, and W. Sherman, "Hydration site thermodynamics explain sars for triazolylpurines analogues binding to the a2a receptor," *ACS Medicinal Chemistry Letters*, vol. 1, no. 4, pp. 160–164, 2010. DOI: 10.1021/ml100008s. eprint: https://doi.org/10.1021/ml100008s. [Online]. Available: https://doi.org/10.1021/ml100008s.

[65] T. Lazaridis, "Inhomogeneous fluid approach to solvation thermodynamics. 1. theory," *The Journal of Physical Chemistry B*, vol. 102, no. 18, pp. 3531–3541, 1998. DOI: 10.1021/jp9723574. eprint: https://doi.org/10.1021/jp9723574. [Online]. Available: https://doi.org/10.1021/jp9723574.

[66] C. N. Nguyen, T. Kurtzman Young, and M. K. Gilson, "Grid inhomogeneous solvation theory: Hydration structure and thermodynamics of the miniature receptor cucurbit[7]uril," *The Journal of Chemical Physics*, vol. 137, no. 4, p. 044 101, 2012. DOI: 10.1063/1.4733951. eprint: https://doi.org/10.1063/1.4733951. [Online]. Available: https://doi.org/10.1063/1.4733951.

[67] M. A. Lill, "Efficient incorporation of protein flexibility and dynamics into molecular docking simulations," *Biochemistry*, vol. 50, no. 28, pp. 6157–6169, Jul. 2011. DOI: 10.1021/bi2004558. [Online]. Available: https://doi.org/10.1021/bi2004558.

[68] Y. Yang, B. Hu, and M. A. Lill, "Analysis of factors influencing hydration site prediction based on molecular dynamics simulations," *J. Chem. Inf. Model.*, vol. 54, no. 10, pp. 2987–2995, Oct. 2014. DOI: 10.1021/ci500426q. [Online]. Available: https://doi.org/10.1021/ci500426q.

[69] Y. Yang and M. A. Lill, "Dissecting the influence of protein flexibility on the location and thermodynamic profile of explicit water molecules in protein-ligand binding," *J. Chem. Theory Comput.*, vol. 12, no. 9, pp. 4578–4592, 2016, PMID: 27494046. DOI: 10.1021/acs.jctc.6b00411. eprint: https://doi.org/10.1021/acs.jctc.6b00411. [Online]. Available: https://doi.org/10.1021/acs.jctc.6b00411.

[70] A. H. Mahmoud, M. R. Masters, Y. Yang, and M. A. Lill, "Elucidating the multiple roles of hydration for accurate protein-ligand binding prediction via deep learning," *Communications Chemistry*, vol. 3, no. 1, Feb. 2020. DOI: 10.1038/s42004-020-0261-x. [Online]. Available: https://doi.org/10.1038/s42004-020-0261-x.

[71] Z. Li and T. Lazaridis, "The effect of water displacement on binding thermodynamics: Concanavalin a," *The Journal of Physical Chemistry B*, vol. 109, no. 1, pp. 662–670, 2005, PMID: 16851059. DOI: 10.1021/jp0477912. eprint: https://doi.org/10.1021/jp0477912. [Online]. Available: https://doi.org/10.1021/jp0477912.

[72] A. Kovalenko and F. Hirata, "Three-dimensional density profiles of water in contact with a solute of arbitrary shape: A rism approach," *Chem. Phys. Lett.*, vol. 290, no. 1, pp. 237–244, 1998, ISSN: 0009-2614. DOI: https://doi.org/10.1016/S0009-2614(98)00471-0. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0009261498004710.

[73] D. J. Sindhikara, N. Yoshida, and F. Hirata, "Placevent: An algorithm for prediction of explicit solvent atom distribution-application to hiv-1 protease and f-atp synthase," *J. Computational Chemistry*, vol. 33, pp. 1536–1543, 2012. DOI: 10.1002/jcc.22984.

[74] D. J. Sindhikara and F. Hirata, "Analysis of biomolecular solvation sites by 3d-rism theory," *The J. Phys. Chem. B*, vol. 117, no. 22, pp. 6718–6723, 2013, PMID: 23675899. DOI: 10.1021/jp4046116. eprint: https://doi.org/10.1021/jp4046116. [Online]. Available: https://doi.org/10.1021/jp4046116.

[75] L. Fusani, I. Wall, D. Palmer, and A. Cortes, "Optimal water networks in protein cavities with gasol and 3d-rism," *Bioinformatics*, bty024, 2018. DOI: 10.1093/bioinformatics/bty024.

[76] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv e-prints*, arXiv:1505.04597, arXiv:1505.04597, May 2015. arXiv: 1505.04597 [cs.CV].

[77] R. Wang, X. Fang, Y. Lu, and S. Wang, "The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures," *J. Med. Chem.*, vol. 47, no. 12, pp. 2977–2980, Jun. 2004. DOI: 10.1021/jm030580l. [Online]. Available: https://doi.org/10.1021/jm030580l.

[78] R. Wang, X. Fang, Y. Lu, C.-Y. Yang, and S. Wang, "The PDBbind database: methodologies and updates," *J. Med. Chem.*, vol. 48, no. 12, pp. 4111–4119, Jun. 2005. DOI: 10.1021/jm048957q. [Online]. Available: https://doi.org/10.1021/jm048957q.

[79] C. R. Søndergaard, M. H. Olsson, M. Rostkowski, and J. H. Jensen, "Improved treatment of ligands and coupling effects in empirical calculation and rationalization of p k a values," *J. Chem. Theory Comput.*, vol. 7, no. 7, pp. 2284–2295, 2011.

[80] M. H. Olsson, C. R. Søndergaard, M. Rostkowski, and J. H. Jensen, "Propka3: Consistent treatment of internal and surface residues in empirical p k a predictions," *J. Chem. Theory Comput.*, vol. 7, no. 2, pp. 525–537, 2011.

[81] D. A. Case, R. Betz, D. Cerutti, T. Cheatham, T. Darden, R. Duke, T. Giese, H. Gohlke, A. Goetz, N. Homeyer, *et al.*, "Amber 2016 reference manual," *University of California, San Francisco*, pp. 1–923, 2016.

[82] M. Baroni, G. Cruciani, S. Sciabola, F. Perruccio, and J. S. Mason, "A common reference framework for analyzing/comparing proteins and ligands. fingerprints for ligands and proteins (flap): Theory and application," *J. Chem. Inf. Model.*, vol. 47, no. 2, pp. 279–294, 2007, PMID: 17381166. DOI: 10.1021/ci600253e. eprint: https://doi.org/10.1021/ci600253e. [Online]. Available: https://doi.org/10.1021/ci600253e.

[83] S. Cross, M. Baroni, L. Goracci, and G. Cruciani, "Grid-based three-dimensional pharmacophores i: Flappharm, a novel approach for pharmacophore elucidation," *J. Chem. Inf. Model.*, vol. 52, no. 10, pp. 2587–2598, 2012, PMID: 22970894. DOI: 10.1021/ci300153d. eprint: https://doi.org/10.1021/ci300153d. [Online]. Available: https://doi.org/10.1021/ci300153d.

[84] G. Cruciani, *Molecular interaction fields: applications in drug discovery and ADME prediction*. Vch Verlagsgesellschaft Mbh, 2006, vol. 1.

[85] P. J. Goodford, "A computational procedure for determining energetically favorable binding sites on biologically important macromolecules," *J. Med. Chem.*, vol. 28, no. 7, pp. 849–857, 1985.

[86] R. Gowers, M. Linke, J. Barnoud, T. Reddy, M. Melo, S. Seyler, J. Domański, D. Dotson, S. Buchoux, I. Kenney, and O. Beckstein, "MDAnalysis: A python package for the rapid analysis of molecular dynamics simulations," in *Proceedings of the 15th Python in Science Conference*, SciPy, 2016. DOI: 10.25080/majora-629e541a-00e. [Online]. Available: https://doi.org/10.25080/majora-629e541a-00e.

[87] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein, "Mdanalysis: A toolkit for the analysis of molecular dynamics simulations," *J. Comput. Chem.*, vol. 32, no. 10, pp. 2319–2327, 2011. DOI: 10.1002/jcc.21787. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.21787. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.21787.

[88] E. Tyantov, *Kaggle ultrasound nerve segmentation competition*, https://github.com/EdwardTyantov/ultrasound-nerve-segmentation, 2016.

[89] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. arXiv: 1512.03385 [cs.CV].

[90] A. Khan, A. Sohail, U. Zahoora, and A. Saeed Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks," *arXiv e-prints*, arXiv:1901.06032, arXiv:1901.06032, Jan. 2019. arXiv: 1901.06032 [cs.CV].

[91] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations," *arXiv e-prints*, arXiv:1707.03237, arXiv:1707.03237, Jul. 2017. arXiv: 1707.03237 [cs.CV].

[92] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," *arXiv e-prints*, arXiv:1606.04797, arXiv:1606.04797, Jun. 2016. arXiv: 1606.04797 [cs.CV].

[93] W. R. Crum, O. Camara, and D. L. G. Hill, "Generalized overlap measures for evaluation and validation in medical image analysis," *IEEE Trans. Med. Imag.*, vol. 25, no. 11, pp. 1451–1461, Nov. 2006, ISSN: 0278-0062. DOI: 10.1109/TMI.2006.880587.

[94] F. Chollet, *Keras*, https://github.com/fchollet/keras, 2015.

[95] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: http://tensorflow.org/.

[96] J. Li, R. Abel, K. Zhu, Y. Cao, S. Zhao, and R. A. Friesner, "The VSGB 2.0 model: A next generation energy model for high resolution protein structure modeling," *Proteins: Structure, Function, and Bioinformatics*, vol. 79, no. 10, pp. 2794–2812, Aug. 2011. DOI: 10.1002/prot.23106. [Online]. Available: https://doi.org/10.1002/prot.23106.

[97] A. Rudling, A. Orro, and J. Carlsson, "Prediction of ordered water molecules in protein binding sites from molecular dynamics simulations: The impact of ligand binding on hydration networks," *J. Chem. Inf. Model.*, vol. 58, no. 2, pp. 350–361, 2018.

[98] N. Weill and D. Rognan, "Alignment-free ultra-high-throughput comparison of druggable protein-ligand binding sites," *J. Chem. Inf. Model.*, vol. 50, no. 1, pp. 123–135, Jan. 2010. DOI: 10.1021/ci900349y. [Online]. Available: https://doi.org/10.1021/ci900349y.

[99] Z. Huang, "Clustering large data sets with mixed numeric and categorical values," in *In The First Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 1997, pp. 21–34.

[100] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery*, vol. 2, no. 3, pp. 283–304, Sep. 1998, ISSN: 1573-756X. DOI: 10.1023/A:1009769707641. [Online]. Available: https://doi.org/10.1023/A:1009769707641.

[101] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. [Online]. Available: https://doi.org/10.1023/A:1010933404324.

[102] D. Kuzminykh, D. Polykovskiy, A. Kadurin, A. Zhebrak, I. Baskov, S. Nikolenko, R. Shayakhmetov, and A. Zhavoronkov, "3d molecular representations based on the wave transform for convolutional neural networks," *Mol. Pharmaceutics*, vol. 15, no. 10, pp. 4378–4385, 2018, PMID: 29473756. DOI: 10.1021/acs.molpharmaceut.7b01134. eprint: https://doi.org/10.1021/acs.molpharmaceut.7b01134. [Online]. Available: https://doi.org/10.1021/acs.molpharmaceut.7b01134.

[103] B. Breiten, M. R. Lockett, W. Sherman, S. Fujita, M. Al-Sayah, H. Lange, C. M. Bowers, A. Heroux, G. Krilov, and G. M. Whitesides, "Water networks contribute to enthalpy/entropy compensation in protein–ligand binding," *J. Am. Chem. Soc.*, vol. 135, no. 41, pp. 15 579–15 584, 2013, PMID: 24044696. DOI: 10.1021/ja4075776. eprint: https://doi.org/10.1021/ja4075776. [Online]. Available: https://doi.org/10.1021/ja4075776.

[104] S. Vaitheeswaran, H. Yin, J. C. Rasaiah, and G. Hummer, "Water clusters in nonpolar cavities," *PNAS*, vol. 101, no. 49, pp. 17 002–17 005, 2004, ISSN: 0027-8424. DOI: 10.1073/pnas.0407968101. eprint: https://www.pnas.org/content/101/49/17002.full.pdf. [Online]. Available: https://www.pnas.org/content/101/49/17002.

[105] A. Artese, S. Cross, G. Costa, S. Distinto, L. Parrotta, S. Alcaro, F. Ortuso, and G. Cruciani, "Molecular interaction fields in drug discovery: Recent advances and future perspectives," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 3, no. 6, pp. 594–613, 2013. DOI: 10.1002/wcms.1150. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1150. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1150.

[106] P.-P. Kung, P.-J. Sinnema, P. Richardson, M. J. Hickey, K. S. Gajiwala, F. Wang, B. Huang, G. McClellan, J. Wang, K. Maegley, S. Bergqvist, P. P. Mehta, and R. Kania, "Design strategies to target crystallographic waters applied to the hsp90 molecular chaperone," *Bioorganic & Medicinal Chem. Lett.*, vol. 21, no. 12, pp. 3557–3562, Jun. 2011. DOI: 10.1016/j.bmcl.2011.04.130. [Online]. Available: https://doi.org/10.1016/j.bmcl.2011.04.130.

[107] M. A. Brodney, G. Barreiro, K. Ogilvie, E. Hajos-Korcsok, J. Murray, F. Vajdos, C. Ambroise, C. Christoffersen, K. Fisher, L. Lanyon, J. Liu, C. E. Nolan, J. M. Withka, K. A. Borzilleri, I. Efremov, C. E. Oborski, A. Varghese, and B. T. O'Neill, "Spirocyclic sulfamides as -secretase 1 (bace-1) inhibitors for the treatment of alzheimer's disease: Utilization of structure based drug design, watermap, and cns penetration studies to identify centrally efficacious inhibitors," *J. Med. Chem.*, vol. 55, no. 21, pp. 9224–9239, 2012, PMID: 22984865. DOI: 10.1021/jm3009426. eprint: https://doi.org/10.1021/jm3009426. [Online]. Available: https://doi.org/10.1021/jm3009426.

[108] S. D. Sharrow, M. V. Novotny, and M. J. Stone, "Thermodynamic analysis of binding between mouse major urinary protein-i and the pheromone 2-sec-butyl-4,5-dihydrothiazole," *Biochemistry*, vol. 42, no. 20, pp. 6302–6309, 2003, PMID: 12755635. DOI: 10.1021/bi026423q. eprint: https://doi.org/10.1021/bi026423q. [Online]. Available: https://doi.org/10.1021/bi026423q.

[109] R. Malham, S. Johnstone, R. J. Bingham, E. Barratt, S. E. V. Phillips, C. A. Laughton, and S. W. Homans, "Strong solute-solute dispersive interactions in a protein-ligand complex," *J. Am. Chem. Soc.*, vol. 127, no. 48, pp. 17 061–17 067, 2005, PMID: 16316253. DOI: 10.1021/ja055454g. eprint: https://doi.org/10.1021/ja055454g. [Online]. Available: https://doi.org/10.1021/ja055454g.

[110] J. P. Arcon, L. A. Defelipe, C. P. Modenutti, E. D. Lopez, D. Alvarez-Garcia, X. Barril, A. G. Turjanski, and M. A. Martí, "Molecular dynamics in mixed solvents reveals protein–ligand interactions, improves docking, and allows accurate binding free energy predictions," *J. Chem. Inf. Model.*, vol. 57, no. 4, pp. 846–863, 2017.

[111] J. P. Arcon, C. P. Modenutti, D. Avendaño, E. D. Lopez, L. A. Defelipe, F. A. Ambrosio, A. G. Turjanski, S. Forli, and M. A. Marti, "Autodock bias: Improving binding mode prediction and virtual screening using known protein–ligand interactions," *Bioinformatics*, vol. 35, no. 19, pp. 3836–3838, 2019.

[112] D. Xue, Y. Gong, Z. Yang, G. Chuai, S. Qu, A. Shen, J. Yu, and Q. Liu, "Advances and challenges in deep generative models for de novo molecule generation," *WIREs Computational Molecular Science*, vol. 9, no. 3, e1395, 2019. DOI: 10.1002/wcms.1395. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1395. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1395.

[113] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," *CoRR*, vol. abs/1411.4555, 2014. arXiv: 1411.4555. [Online]. Available: http://arxiv.org/abs/1411.4555.

[114] T. Liu, Y. Lin, X. Wen, R. N. Jorissen, and M. K. Gilson, "Bindingdb: A web-accessible database of experimentally determined protein-ligand binding affinities," eng, *Nucleic acids research*, vol. 35, no. Database issue, pp. D198–D201, Jan. 2007, gkl999[PII], ISSN: 1362-4962. DOI: 10.1093/nar/gkl999. [Online]. Available: https://doi.org/10.1093/nar/gkl999.

[115] M. Karimi, D. Wu, Z. Wang, and Y. Shen, "DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks," *Bioinformatics*, vol. 35, no. 18, pp. 3329–3338, Feb. 2019, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz111. eprint: https://academic.oup.com/bioinformatics/article-pdf/35/18/3329/30024539/btz111.pdf. [Online]. Available: https://doi.org/10.1093/bioinformatics/btz111.

[116] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and U. Consortium, "Uniref clusters: A comprehensive and scalable alternative for improving sequence similarity searches," eng, *Bioinformatics (Oxford, England)*, vol. 31, no. 6, pp. 926–932, Mar. 2015, btu739[PII], ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btu739. [Online]. Available: https://doi.org/10.1093/bioinformatics/btu739.

[117] M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes, and B. Rost, "Modeling aspects of the language of life through transfer-learning protein sequences," *BMC Bioinformatics*, vol. 20, no. 1, p. 723, Dec. 2019, ISSN: 1471-2105. DOI: 10.1186/s12859-019-3220-8. [Online]. Available: https://doi.org/10.1186/s12859-019-3220-8.

[118] G. Landrum, "Rdkit: Open-source cheminformatics software," 2019. [Online]. Available: https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4.

[119] D. Polykovskiy, A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy, M. Veselov, A. Kadurin, S. I. Nikolenko, A. Aspuru-Guzik, and A. Zhavoronkov, "Molecular sets (MOSES): A benchmarking platform for molecular generation models," *CoRR*, vol. abs/1811.12823, 2018. arXiv: 1811.12823. [Online]. Available: http://arxiv.org/abs/1811.12823.

[120] J. Degen, C. Wegscheid-Gerlach, A. Zaliani, and M. Rarey, "On the art of compiling and using 'drug-like' chemical fragment spaces," *ChemMedChem*, vol. 3, no. 10, pp. 1503–1507, 2008. DOI: 10.1002/cmdc.200800178. eprint: https://chemistry-europe.onlinelibrary.wiley.com/doi/pdf/10.1002/cmdc.200800178. [Online]. Available: https://chemistry-europe.onlinelibrary.wiley.com/doi/abs/10.1002/cmdc.200800178.

[121] G. W. Bemis and M. A. Murcko, "The properties of known drugs. 1. molecular frameworks," *Journal of Medicinal Chemistry*, vol. 39, no. 15, pp. 2887–2893, 1996, PMID: 8709122. DOI: 10.1021/jm9602928. eprint: https://doi.org/10.1021/jm9602928. [Online]. Available: https://doi.org/10.1021/jm9602928.

[122] P. Ertl and A. Schuffenhauer, "Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions," *Journal of cheminformatics*, vol. 1, no. 1, p. 8, 2009.

[123] G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan, and A. L. Hopkins, "Quantifying the chemical beauty of drugs," *Nature Chemistry*, vol. 4, no. 2, pp. 90–98, Feb. 2012, ISSN: 1755-4349. DOI: 10.1038/nchem.1243. [Online]. Available: https://doi.org/10.1038/nchem.1243.

[124] P. Ertl, S. Roggo, and A. Schuffenhauer, "Natural product-likeness score and its application for prioritization of compound libraries," *Journal of Chemical Information and Modeling*, vol. 48, no. 1, pp. 68–74, 2008, PMID: 18034468. DOI: 10.1021/ci700286x. eprint: https://doi.org/10.1021/ci700286x. [Online]. Available: https://doi.org/10.1021/ci700286x.

[125] M. H. Segler, T. Kogej, C. Tyrchan, and M. P. Waller, "Generating focused molecule libraries for drug discovery with recurrent neural networks," *ACS central science*, vol. 4, no. 1, pp. 120–131, 2018.

[126] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[127] M. Waskom, O. Botvinnik, D. O'Kane, P. Hobson, S. Lukauskas, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. de Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, A. Miles, Y. Ram, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, Brian, C. Fonnesbeck, A. Lee, and A. Qalieh, *Mwaskom/seaborn: V0.8.1 (september 2017)*, version v0.8.1, Sep. 2017. DOI: 10.5281/zenodo.883859. [Online]. Available: https://doi.org/10.5281/zenodo.883859.

[128] J. Adachi, M. Kishida, S. Watanabe, Y. Hashimoto, K. Fukamizu, and T. Tomonaga, "Proteome-wide discovery of unknown atp-binding proteins and kinase inhibitor target proteins using an atp probe," *Journal of Proteome Research*, vol. 13, no. 12, pp. 5461–5470, 2014, PMID: 25230287. DOI: 10.1021/pr500845u. eprint: https://doi.org/10.1021/pr500845u. [Online]. Available: https://doi.org/10.1021/pr500845u.

[129] S. Riniker and G. A. Landrum, "Similarity maps-a visualization strategy for molecular fingerprints and machine-learning methods," *Journal of cheminformatics*, vol. 5, no. 1, p. 43, 2013.

[130] K. Tamiola, M. M. Heberling, and J. Domanski, "Structural propensity database of proteins," *bioRxiv*, 2017. DOI: 10.1101/144840. eprint: https://www.biorxiv.org/content/early/2017/06/01/144840.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2017/06/01/144840.

[131] L. M. T. Lima and G. de Prat-Gay, "Conformational changes and stabilization induced by ligand binding in the dna-binding domain of the e2 protein from human papillomavirus," *Journal of Biological Chemistry*, vol. 272, no. 31, pp. 19 295–19 303, 1997.

[132] L. N. Johnson and R. J. Lewis, "Structural basis for control by phosphorylation," *Chemical reviews*, vol. 101, no. 8, pp. 2209–2242, 2001.

[133] D. A. Antunes, D. Devaurs, and L. E. Kavraki, "Understanding the challenges of protein flexibility in drug design," *Expert opinion on drug discovery*, vol. 10, no. 12, pp. 1301–1313, 2015.

[134] W. Sherman, T. Day, M. P. Jacobson, R. A. Friesner, and R. Farid, "Novel procedure for modeling ligand/receptor induced fit effects," *Journal of medicinal chemistry*, vol. 49, no. 2, pp. 534–553, 2006.

[135] S.-Y. Huang and X. Zou, "Ensemble docking of multiple protein structures: Considering protein structural variations in molecular docking," *Proteins: Structure, Function, and Bioinformatics*, vol. 66, no. 2, pp. 399–421, 2007.

[136] W. Sherman, H. S. Beard, and R. Farid, "Use of an induced fit receptor structure in virtual screening," *Chemical biology & drug design*, vol. 67, no. 1, pp. 83–84, 2006.

[137] V. L. Rath, M. Ammirati, P. K. LeMotte, K. F. Fennell, M. N. Mansour, D. E. Danley, T. R. Hynes, G. K. Schulte, D. J. Wasilko, and J. Pandit, "Activation of human liver glycogen phosphorylase by alteration of the secondary structure and packing of the catalytic core," *Molecular cell*, vol. 6, no. 1, pp. 139–148, 2000.

[138] F. Meng, V. N. Uversky, and L. Kurgan, "Comprehensive review of methods for prediction of intrinsic disorder and its molecular functions," *Cellular and Molecular Life Sciences*, vol. 74, no. 17, pp. 3069–3090, 2017.

[139] J. Prilusky, C. E. Felder, T. Zeev-Ben-Mordehai, E. H. Rydberg, O. Man, J. S. Beckmann, I. Silman, and J. L. Sussman, "Foldindex©: A simple tool to predict whether a given protein sequence is intrinsically unfolded," *Bioinformatics*, vol. 21, no. 16, pp. 3435–3438, 2005.

[140] R. Linding, R. B. Russell, V. Neduva, and T. J. Gibson, "Globplot: Exploring protein sequences for globularity and disorder," *Nucleic acids research*, vol. 31, no. 13, pp. 3701–3708, 2003.

[141] Z. Dosztányi, V. Csizmok, P. Tompa, and I. Simon, "Iupred: Web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content," *Bioinformatics*, vol. 21, no. 16, pp. 3433–3434, 2005.

[142] O. V. Galzitskaya, S. O. Garbuzynskiy, and M. Y. Lobanov, "Foldunfold: Web server for the prediction of disordered regions in protein chain," *Bioinformatics*, vol. 22, no. 23, pp. 2948–2949, 2006.

[143] P. Romero, Z. Obradovic, C. Kissinger, J. Villafranca, and A. K. Dunker, "Identifying disordered regions in proteins from amino acid sequence," in *Proceedings of International Conference on Neural Networks (ICNN'97)*, IEEE, vol. 1, 1997, pp. 90–95.

[144] K. Peng, S. Vucetic, P. Radivojac, C. J. Brown, A. K. Dunker, and Z. Obradovic, "Optimizing long intrinsic disorder predictors with protein evolutionary information," *Journal of bioinformatics and computational biology*, vol. 3, no. 01, pp. 35–60, 2005.

[145] K. Peng, P. Radivojac, S. Vucetic, A. K. Dunker, and Z. Obradovic, "Length-dependent prediction of protein intrinsic disorder," *BMC bioinformatics*, vol. 7, no. 1, p. 208, 2006.

[146] Z. Obradovic, K. Peng, S. Vucetic, P. Radivojac, and A. K. Dunker, "Exploiting heterogeneous sequence properties improves prediction of protein disorder," *Proteins: Structure, Function, and Bioinformatics*, vol. 61, no. S7, pp. 176–182, 2005.

[147] T. Zhang, E. Faraggi, B. Xue, A. K. Dunker, V. N. Uversky, and Y. Zhou, "Spine-d: Accurate prediction of short and long disordered regions by a single neural-network based method," *Journal of Biomolecular Structure and Dynamics*, vol. 29, no. 4, pp. 799–813, 2012.

[148] M. S. Klausen, M. C. Jespersen, H. Nielsen, K. K. Jensen, V. I. Jurtz, C. K. Sønderby, M. O. A. Sommer, O. Winther, M. Nielsen, B. Petersen, *et al.*, "Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning," *Proteins: Structure, Function, and Bioinformatics*, vol. 87, no. 6, pp. 520–527, 2019.

[149] T. Ishida and K. Kinoshita, "Prdos: Prediction of disordered protein regions from amino acid sequence," *Nucleic acids research*, vol. 35, no. suppl_2, W460–W464, 2007.

[150] L. P. Kozlowski and J. M. Bujnicki, "Metadisorder: A meta-server for the prediction of intrinsic disorder in proteins," *BMC bioinformatics*, vol. 13, no. 1, p. 111, 2012.

[151] X. Fan and L. Kurgan, "Accurate prediction of disorder in protein chains with a comprehensive and empirically designed consensus," *Journal of Biomolecular Structure and Dynamics*, vol. 32, no. 3, pp. 448–464, 2014.

[152] M. J. Mizianty, W. Stach, K. Chen, K. D. Kedarisetti, F. M. Disfani, and L. Kurgan, "Improved sequence-based prediction of disordered regions with multilayer fusion of multiple information sources," *Bioinformatics*, vol. 26, no. 18, pp. i489–i496, 2010.

[153] M. J. Mizianty, Z. Peng, and L. Kurgan, "Mfdp2: Accurate predictor of disorder in proteins by fusion of disorder probabilities, content and profiles," *Intrinsically disordered proteins*, vol. 1, no. 1, e24428, 2013.

[154] D. T. Jones and D. Cozzetto, "Disopred3: Precise disordered region predictions with annotated protein-binding activity," *Bioinformatics*, vol. 31, no. 6, pp. 857–863, 2015.

[155] K. Tamiola, B. Acar, and F. A. Mulder, "Sequence-specific random coil chemical shifts of intrinsically disordered proteins," *Journal of the American Chemical Society*, vol. 132, no. 51, pp. 18 000–18 003, 2010.

[156] Y. Wang and O. Jardetzky, "Investigation of the neighboring residue effects on protein chemical shifts," *Journal of the American Chemical Society*, vol. 124, no. 47, pp. 14 075–14 084, 2002.

[157] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[158] T. Amemiya, R. Koike, A. Kidera, and M. Ota, "Pscdb: A database for protein structural change upon ligand binding," *Nucleic acids research*, vol. 40, no. D1, pp. D554–D558, 2012.

[159] A. Barducci, M. Bonomi, and M. Parrinello, "Metadynamics," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 1, no. 5, pp. 826–843, 2011.

[160] J. Zhang, H. Zhang, C. Xia, and L. Sun, "Graph-bert: Only attention is needed for learning graph representations," *arXiv preprint arXiv:2001.05140*, 2020.

[161] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," *arXiv preprint arXiv:1802.04407*, 2018.

VITA

Ahmadreza Ghanbarpour was born in July of 1988 in Tehran, Iran. He entered the school of pharmacy in Tehran University of Medical Sciences in October 2007 to obtain his Doctor of Pharmacy degree. He finished his degree requirements in January 2013. During this time, he received training on various concepts of drug discovery and development from research and clinical perspectives. His dissertation was on preparing human serum Albumin conjugates of SN38 for the targeted treatment of cancer which led to a publication. In 2015, he was admitted to Purdue University and he moved to the United States to earn a PhD in Medicinal Chemistry and Molecular Pharmacology. Shortly after, he joined Dr. Markus Lill's lab to pursue research in the field of computer-aided drug design.

His initial research work was on development of computational tools for protein fibrillation prediction. During his final years of PhD he shifted towards developing methods based on deep neural networks to address different problems in the field of drug discovery, such as development of methods for prediction of protein hydration properties, target-based de novo design of molecules and prediction of protein disorder using deep neural networks.

After finishing his PhD, Ahmad aims to receive additional training as a postdoctoral research scientist. He has accepted an offer from Eli Lilly and he looks forward to working on methods based on deep neural networks to assist the design of antibodies as therapeutics.