Analysis of Protein-Protein Docking Through Interaction Maps, and a Machine Learning Method for Map Generation

Jared Taylor Ottley

Thesis submitted to the faculty of the School of Physics at
University College Dublin in partial fulfilment
of the requirements for the degree of
MASTER OF SCIENCE
in
Computational Physics

Supervisor: Assoc. Prof. Nicolae-Viorel Buchete

September 2024

Acknowledgments

I would like to thank my advisor, Professor Nicolae Vio Buchete, for advice and mentorship in the writing of this work, as well as Ovidiu Petrisor, a graduate student working under Professor Buchete. I would also like to thank and acknowledge the contributions of the University College Dublin IT Services for the use of the Sonic High Performance Computing cluster for performing the calculations used in this work.

Statement of Original Authorship

I certify that this work is of original authorship and was conducted while a taught master's student at University College Dublin. I also certify that no other degrees are or will be granted based on the research conducted in this work.

Jared Taylor Ottley

09/02/2024

Statement of Data and Code Availability

The code written for this work, as well as the data used in it, are available online in the below GitHub repository.

https://github.com/KoStar2/UCD-Masters-Thesis

Contents

Contents 5

Abstract 7

List of Abbreviations 8

List of Figures 9

List of Tables 12

1. Introduction 13

1.1. Proteins 13

1.2. Protein Structures 14

1.3. Protein Interactions 15

2. Methods 17

2.1. Dataset Selection 17

2.2. Reference Frame Definition 17

2.3. Molecular Docking 18

2.3.1. Docking Procedures: Preprocessing 19

2.3.2. Docking Procedures: Pose Identification 21

2.3.3. Docking Procedures: Scoring 21

2.3.4. Docking in This Work 22

2.4. PPI Maps 22

2.4.1. Optimization of Bin Size 24

2.5. Dataset Splitting Based on Biophysical Properties 26

2.6. Machine Learning 26

2.6.1. Neural Networks 27

2.6.2. Machine Learning in this Work 28

3. Results and Discussion 32

3.1. Dataset Analysis 32

3.2. Biophysical Property Sampling 35

3.3. Bin Size Optimization 37

3.4. Docking Results 40

3.4.1. PPI Maps and Charge 40

3.4.2. PPI Maps and Radius of Gyration 43

3.4.3. PPI Maps and SASAH 45
3.4.4. PPI Maps and Arithmetic Surface Roughness 47
3.4.5. PPI Maps and Mass 50
3.5. Machine Learning Analysis 52
4. Conclusions..58
References 61

Abstract

Modeling protein-protein interactions (PPIs) is fundamental to understanding the functions and structures of protein complexes, diseases and disorders caused by interactions, and changes in biological activity. Modern methods of studying PPIs, such as molecular docking, are hindered by large elaborate proteins, as well as calculation times for large protein datasets, leading to a desire for methods to reduce the calculation complexity.

In this work, two developments are presented. The first is a method of representing the likelihood of a pair of proteins to interact based on the orientation of the proteins. This is accomplished through modifying the heat maps used by Lopez et al. to represent the adsorption energy of proteins onto a nanoparticle surface into "PPI maps" to model the docking scores of protein-protein interaction[1]. The orientational accuracy of these maps is assessed by comparing the predicted orientation of homodimers using Pearson's correlation coefficient and KL divergence. These methods showed that below 30° PPI docking data generated from PatchDock is subject to noise and randomness, while above 45° the topology of the maps have flattened beyond the point of use.

The second development is the creation of a dataset to test the applicability of machine learning (ML) models as tools for the prediction the PPI maps. The dataset has been split into three sets for training, testing, and validation of ML models, based on the goal of conserving the distribution of biophysical properties of proteins across each set. This was done to encourage models trained on the dataset to avoid overfitting to subsets of proteins with specific biophysical factors, such as high sphericity or hydrophobic proteins. A set of small and simple NN architectures using geometric inputs were tested on the dataset, with inconclusive results as to the effectiveness of this dataset due to the model's overgeneralizing to a single output for all proteins.

List of Abbreviations

COM	Center of Mass
GNN	Graphical Neural Network
JS	Jenson-Shannon (Divergence)
KL	Kullback-Liebler (Divergence)
ML	Machine Learning
MPNN	Message Passing Neural Network
NN	Neural Network
PCC	Pearson’s Correlation Coefficient
PPIs	Protein-Protein Interactions
RMS	Root Mean Square
RMSD	Root Mean Square Deviation
SASA	Solvent Accessible Surface Area
SASA$_H$	Hydrophobic Fraction of Solvent Accessible Surface Area

List of Figures

Figure 1. Surface of Anthocyanin Synthase (1GP6), colored by amino acid type. Basic groups are blue, acidic groups are red, polar groups are green, and nonpolar groups are white. 14
Figure 2. The A chain of the 1COZ protein, modelled using (left) all-atoms and (right) the secondary structure of the chain. 15
Figure 3. The 1COZ chain A homodimer found through (A) The experimental resolution[2], (B) PatchDock simulation in this work[4, 5]. (C) Overlap of the two experimental and simulated homodimers. RMSD between the dimers is $6.85 \cdot 10 - 3$ Å. 16
Figure 4. Protein oriented so that the largest moment of inertia is along the z-axis and the smallest is along the x-axis[1]. 18
Figure 5. A diagrammatic sequence of the steps taken in rigid docking[13]. Note that steps (a) and (b) are neglected in many docking programs. (a) The conformation space of the ligand molecule is explored to generate likely geometries. (b) Likely interaction sites are identified. (c) Possible interactions between the receptor and ligand are identified. (d) The solutions are scored, and the best scoring solutions are evaluated to generate a docked model. 20
Figure 6. An example of PPI maps calculated for a homodimer of isovaleryl-CoA dehydrogenase (4O5M)'s chain A; the highest scoring pose is shown in the bottom left. The bins on both maps cover widths and lengths of 30 degrees. The map in the top right shows the direction of the ligand's COM with respect to the COM of the receptor. The map on the top right shows the direction of the receptor's COM with respect to the ligand's COM. The map on the bottom right shows the average of the receptor and ligand maps. 23
Figure 7. (a) An example NN with one hidden layer (b) An example of a NN with many hidden layers[24]. 28
Figure 8. Diagrammatic illustration of the procedure for generating PPI maps using machine learning. 29
Figure 9. PCC between the biophysical properties of the LOMETS3 dataset calculated by Ovidiu A. Petrisor[19]. 32
Figure 10. Distribution between select biophysical properties of the LOMETS3 dataset. The PCC between the properties is displayed in the top right corner. 34
Figure 11. 80-10-10 split of the training (blue), validation (orange), and testing (green) datasets. 36
Figure 12. Pearson's correlation coefficient of ligand and receptor maps at 5° increments, averaged over the entire dataset, as a function of the bin length. The black lines indicate 10, 20, 30, and 45°. 37
Figure 13. KL divergence of the (red) ligand map from the receptor map and (blue) the divergence of the receptor map from the ligand map, averaged over the entire dataset, as a function of the bin length. 38
Figure 14. JS divergence of the receptor and ligand maps, averaged over the entire dataset, as a function of bin length. The black lines indicate 10, 20, 30, and 45°. 39

Figure 15. (A) The distribution of protein charges in the LOMETS3 dataset. The PPI map and highest scoring docking pose for (B) the most negative protein, chain A of 1EXR, (C) a neutral protein, chain A of 1LXJ, (D) the most positive protein, chain U of 2J01. 42
Figure 16. (A) The distribution of protein radius of gyration in the LOMETS3 dataset. The PPI map and highest scoring docking pose for (B) The smallest radius of gyration, chain A of 2BCR, (C) an average radius of gyration, chain X of 2IV2, (D) The largest non-outlier radius of gyration, chain A of 3UMH...44
Figure 17. (A) The distribution of protein $SASA_H$ in the LOMETS3 dataset. The PPI map and highest scoring docking pose for (B) The smallest $SASA_H$, chain A of 2MSP, (C) an average $SASA_H$, chain A of 3O6C, (D) The largest non-outlier $SASA_H$, chain C of 2PD1. 46
Figure 18. Homodimer of the 2PD1 chain C protein, visualize using the residue type of the protein surface. The interface between proteins contains a high amount of matching non-polar groups (white) and paired acidic and basic groups (red and blue, respectively). 47
Figure 19. (A) The distribution of protein arithmetic roughness in the LOMETS3 dataset. The PPI map and highest scoring docking pose for (B) The smoothest protein, chain B of 1J4I, (C) an average roughness, chain A of 2VER, (D) The roughest protein, chain B of 3CKC. 49
Figure 20. (A) The distribution of protein mass, calculated by the number of alpha carbons and the number of heavy atoms in the LOMETS3 dataset. The PPI map and highest scoring docking pose for (B) The lightest protein, chain A of 2BCR, (C) an average mass protein, chain A of 3RK1, (D) The most massive protein chain B of 3CKC...51
Figure 21. Learning loss on the training dataset calculated for the final member of each epoch. The figure on the right presents the same information on a finer scale for better viewing of the 2 sigmoidal and 3 sigmoidal training loss. Circle markers represent models with 3 sigmoidal layers; square markers represent models with 1 ReLU layer followed by 2 sigmoidal layers. Triangular markers represent models with 2 ReLU layers followed by a sigmoidal layer. Blue, orange, and green lines represent learning rates of 0.01, 0.005, and 0.001, respectively. 52
Figure 22. PPI maps (left) generated by the 3 Sigmoidal model with a learning rate of 0.001 using a MSE loss function compared to (right) PPI maps generated by PatchDock for (A) 3PMG chain A and (B) 1JG8 chain D. 53
Figure 23. PPI maps (left) generated by the 2-ReLU, 1 Sigmoidal model with a learning rate of 0.001 using a MSE loss function compared to (right) PPI maps generated by PatchDock for (A) 3PMG chain A and (B) 1JG8 chain D. 54
Figure 24. Training loss for the final member of each epoch for the 3-sigmoidal model using a JS divergence loss function. Blue, orange, and green lines represent learning rates of 0.01, 0.005, and 0.001. 55
Figure 25. PPI maps (left) generated by the 3 Sigmoidal model with a learning rate of 0.001 using a JS divergence loss function compared to (right) PPI maps generated by PatchDock for (A) 3PMG chain A and (B) 1JG8 chain D. 56
Figure 26. PPI maps (left) generated by the large 3 Sigmoidal model with a learning rate of 0.01 using a MSE divergence loss function compared to (right) PPI maps generated by PatchDock for (A) 3PMG chain A and (B) 1JG8 chain D. 57

Figure 27. PPI maps (left) generated by the large 3 Sigmoidal model with a learning rate of 0.01 using a JS divergence loss function compared to (right) PPI maps generated by PatchDock for (A) 3PMG chain A and (B) 1JG8 chain D. 57

List of Tables

Table 1. One-hot encoding indexes used for PDB featurization. 31
Table 2. Average MSE of the Validation Dataset After Training 53

1. Introduction

1.1. Proteins

Proteins are one of the three primary macromolecules of biological systems, along with DNA and RNA. They have the most extensive function set of biological molecules, serving as catalysts (referred to as enzymes), structural units, molecular transporters, and many other essential tools.

From a chemical perspective, proteins are polymers built out of a chain of amino acids, joined by peptide bonds between their C-terminal, a carboxyl group, and their N-terminal, a nitrogen. There are 20 amino acids prevalent in biological systems, with a variety of properties arising from differences in their side chains. Amino acids are typically grouped based on the acidity of the side group, using the presence of charged atom, or if the molecule is polar or non-polar. Figure 1 presents a protein “surface”, with amino acids colored by their residue archetype.

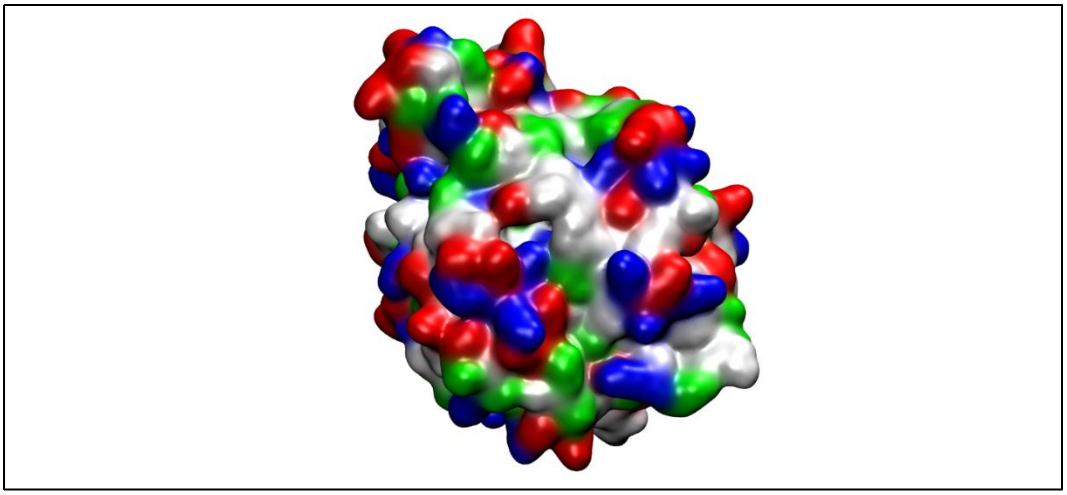


Figure 1. Surface of Anthocyanin Synthase (1GP6), colored by amino acid type. Basic groups are blue, acidic groups are red, polar groups are green, and nonpolar groups are white.

1.2. Protein Structures

The structural landscape of proteins is complex, due to their large size, as well as the number of possible building blocks. Due to this, descriptions of protein structures are split into four levels, the primary, secondary, tertiary, and quaternary structures. The primary structure refers to the amino acid sequence, usually listed as a string of one-letter codes for corresponding amino acids. The secondary structure is the local sequence area, which tend to form into helices or pleated sheets, referred to as α-helices or β-sheets, respectively. The tertiary structure refers to the entirety of a protein molecule, which can fold into one or more conformations, which dictate a protein's ability to perform physiological tasks.

The quaternary structure is the aggregation of multiple subunits; chains, which are single macromolecules, and domains, which are one or more chains whose folding is independent from the rest of the protein. Figure 2 shows a single folded domain of the bacteria Bacillus Subtilis's CTP:glycerol-3-phosphate cytidylyltransferase protein, referred to in the Protein Databank as

1COZ, consisting solely of its “A” chain, modeled using its all-atom representation, as well as its secondary structure[2]. Prediction of the quaternary structure is the goal of protein-protein docking, which will be discussed in section 2.3.

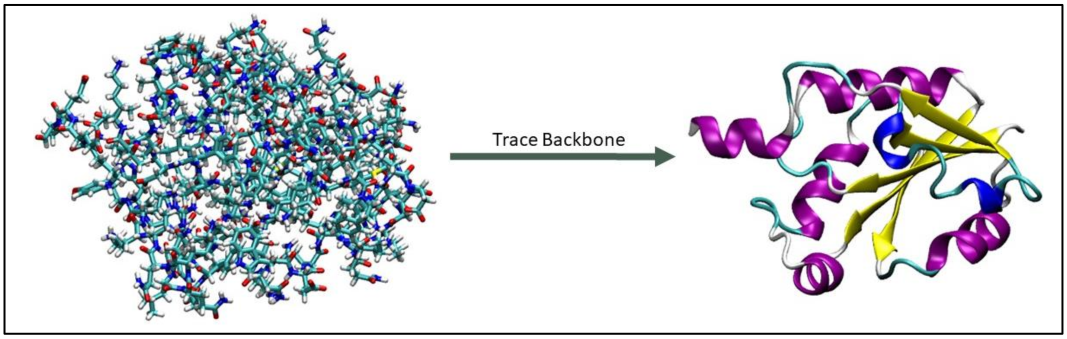


Figure 2. The A chain of the 1COZ protein, modelled using (left) all-atoms and (right) the secondary structure of the chain.

1.3. Protein Interactions

While every protein has a variety of possible interaction sites, these sites aren’t equal in purpose and capability; some sites are more likely to interact with other molecules based on physical compatibility measures. Interaction sites tend to exist either on the surface of the protein or, more likely, inside a “binding pocket”, which leads to specificity in the molecules that can bind to the protein. Prediction of these sites requires information about all four of the protein’s structure levels, as the existence and availability of binding pockets depend on the folding of the protein.

While some proteins have a biological purpose that doesn’t require any cooperating proteins, many work in tandem with each other to better execute their tasks. Examples include complexes such as ATP synthase, whose different components form two molecular rotors which pump protons through a membrane, driving the attachment of a phosphate group to an ADP molecule in order to form an ATP molecule[3].

Figure 3 contains an example of a PPI, the homodimer generated from the interaction of two of the 1COZ protein's chain A, using both an experimental representation of the 1COZ homodimer, as well as the structure predicted by this work.

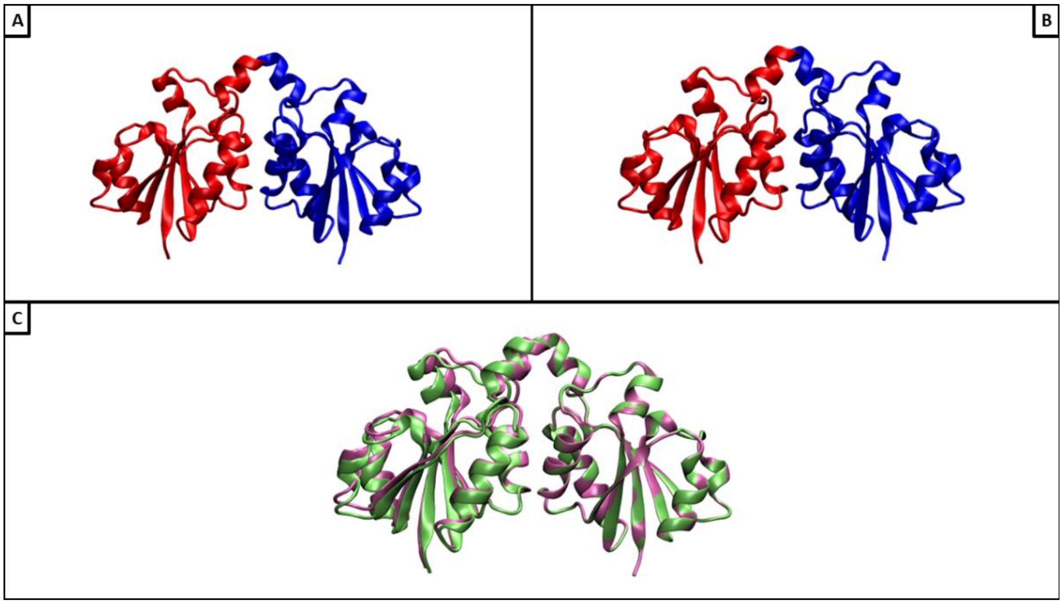


Figure 3. The 1COZ chain A homodimer found through (A) The experimental resolution[2], (B) PatchDock simulation in this work[4, 5]. (C) Overlap of the two experimental and simulated homodimers. RMSD between the dimers is $\mathbf{6.85 \cdot 10^{-3}}$ Å.

2. Methods

2.1. Dataset Selection

The dataset used in this work is the benchmark set used by Zheng et. al for testing of the LOMETS3 protein structure prediction server[6]. While the LOMETS3 dataset is not experimental data, it has several advantages compared to other datasets[7]. The first is that the dataset contains 614 single-domain proteins; in this work, multi-domain proteins have been removed from the dataset. The solutions found by docking single domain proteins together are expected to be similar, allowing for an easier test case for the use of machine learning in docking. Second is the wide distribution of biophysical properties spanned by the dataset, enabling sampling methods that avoid overfitting ML models created on the dataset to a protein subset, such as spherical or hydrophobic proteins. Finally, the dataset is non-homologous, meaning that proteins with a closely related evolutionary ancestor have been removed, and non-redundant, meaning that proteins with identical sequence have been removed from the dataset. These factors lead to a variance between each protein in the dataset, reducing the possibility of overfitting a ML model to a specific set of similar proteins.

2.2. Reference Frame Definition

The asymmetry of proteins introduces inherent complications in comparisons between them. In order to maintain a consistent coordinate system between proteins, we adopted the reference frame defined by Lopez et al[1]. The COM of the protein's alpha carbons is chosen as the reference frame origin. The axes are aligned so that the z-axis is concurrent with the largest principal axis, and the x-axis lies along the smallest. Angles θ and φ are used to represent the direction of points from the origin along the z-y and x-y planes. A graphical representation of an oriented protein system can be seen in Figure 4[1].

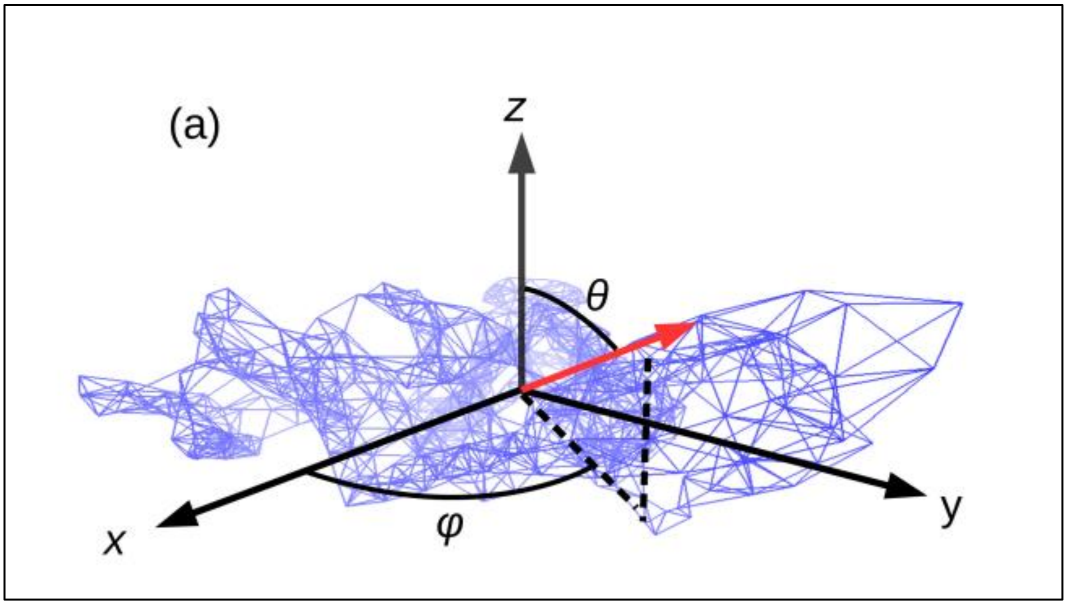


Figure 4. Protein oriented so that the largest moment of inertia is along the z-axis and the smallest is along the x-axis[1].

A two step-procedure was used to fit the proteins in the LOMETS3 dataset to the defined reference frame. First, Hydrogens were added to the proteins using the PyMol library[8]. Then, the open-source python library MDAnalysis was used to center the proteins on the COM of the α-carbons, with the largest principal axis aligned to the z-axis and the smallest aligned to the x-axis[9, 10].

2.3. Molecular Docking

Molecular docking is a computational method used to aid in predicting the most likely interaction sites between two molecules. Early development of molecular docking focused on calculating the binding interactions between drug candidates, labelled the ligand molecules, and proteins, labeled receptor or target molecules. The past decade has seen an increased focus on docking larger molecules together, specifically protein-protein docking and protein-peptide docking. This has led to the development of programs designed specifically for protein-protein docking, such as ZDock and Haddock[11, 12].

2.3.1. Docking Procedures: Preprocessing

Figure 5 illustrates a common pathway for protein-peptide docking; while this is in name a different procedure than protein-protein docking, the process remains the same[13]. The first steps in the diagram, subfigures A and B, are preprocessing steps that respectively increase the accuracy of docking by more fully evaluating the conformational space of the system and reduce the complexity of calculations. In subfigure A, the conformations of the ligand are evaluated, allowing for the flexibility of the system to be introduced to the docking procedure; the same step may be taken for the receptor molecule, leading to a complete conformational analysis of the system. Subfigure B uses knowledge of the receptor molecule to reduce the search space to binding sites that have been previously identified. These steps are commonly neglected in protein-protein docking works, including this work, due to computational expense from the system size, or lack of experimental data.

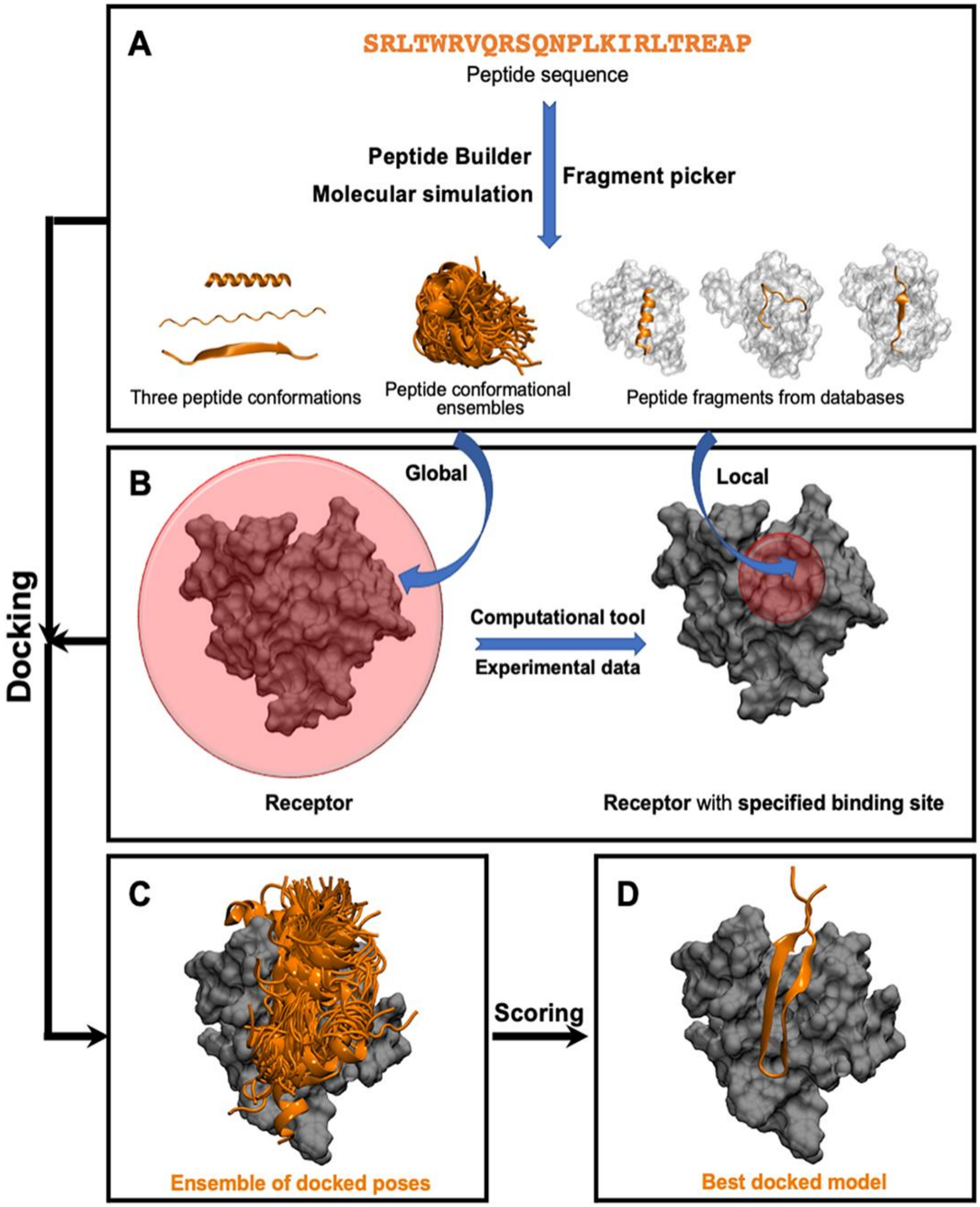


Figure 5. A diagrammatic sequence of the steps taken in rigid docking[13]. Note that steps (a) and (b) are neglected in many docking programs. (a) The conformation space of the ligand molecule is explored to generate likely geometries. (b) Likely interaction sites are identified. (c) Possible interactions between the receptor and ligand are identified. (d) The solutions are scored, and the best scoring solutions are evaluated to generate a docked model.

2.3.2. Docking Procedures: Pose Identification

Following preprocessing, the information gathered is used to calculate docking poses, (subfigures C and D in Figure 5). An extensive review of the docking procedures used, which are summarized in this and section 2.3.3, may be found in the work by Mohanty and Mohanty[14]. Depending on the knowledge of the system, the calculations can be rigid, where both target and receptor are held rigid, semi-flexible, where conformations of the ligand are considered, and flexible, where the full conformation space of both molecules are explored. Docking then takes place by analyzing the possible binding sites between the two molecules.

There are two main methods to do perform the docking analysis. The first method is based in protein geometry, and is used in in PatchDock[4, 5], the docking program used in this work. In the geometric method, the Connolly surfaces of both molecules are calculated and separated into "patches". Depending on the atoms or amino acids present in the patch, it is matched to a partner patch on the other molecule to generate a docking pose, with the many combinations of partners allowing for a vast number of poses to be identified. The second method, used in programs such as Haddock[12], is data-based, where the molecules are matched to known systems that are similar geometrically or sequence-wise.

2.3.3. Docking Procedures: Scoring

After a variety of docking poses have been identified, a docking score is calculated. There are many ways to define docking score, with nearly all docking programs having a unique method of doing so, but the evaluation is typically limited to geometric, energetic, and entropic components.

2.3.4. Docking in This Work

This work used the PatchDock program for docking calculations. The docking procedure was rigid with respect to both molecules. The receptor surface was searched globally for possible PPIs. The fine PatchDock docking method, with explicit calculation of atomic contact energies, was used.

2.4. PPI Maps

The docking score and the number of attempted dockings at a given theta and phi, indicating the direction of the COM of the ligand molecule with respect to the COM of the target molecule, were used to create PPI maps, such as those in Figure 6. These maps are a modification of those proposed by Lopez et al for the modeling the adsorption energy of a protein onto the surface of a nanoparticle[1]. Because nanoparticles are (typically) symmetric, Lopez et al. avoid the need to consider the orientation of the target nanoparticle. Proteins, however, are asymmetric, so a complete prediction of a PPI requires the orientation of both molecules to be calculated. Due to this, a map of the ligand and the receptor are required.

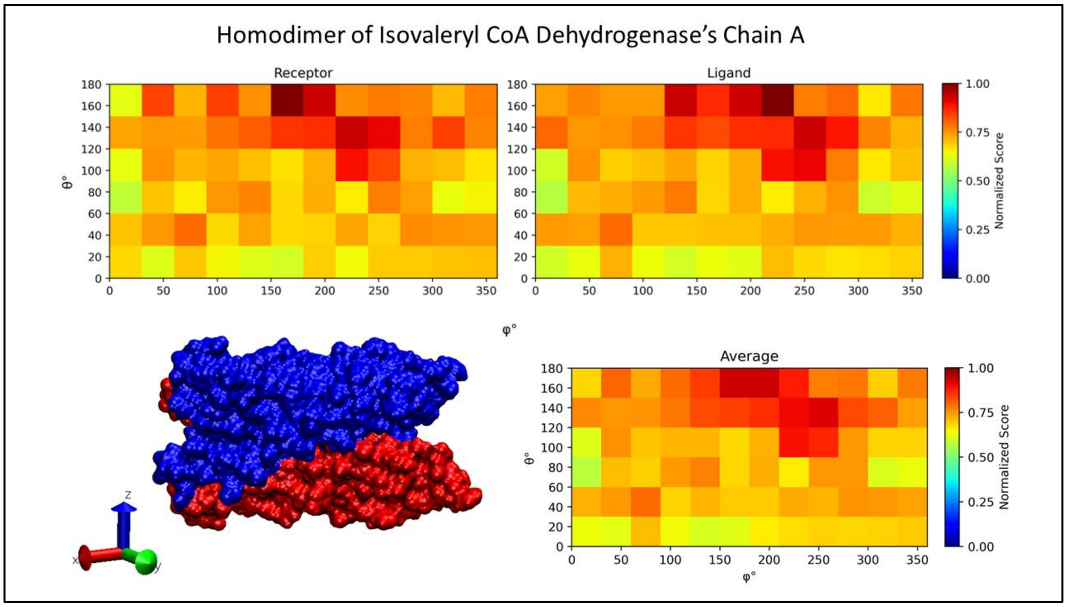


Figure 6. An example of PPI maps calculated for a homodimer of isovaleryl-CoA dehydrogenase (4O5M)'s chain A; the highest scoring pose is shown in the bottom left. The bins on both maps cover widths and lengths of 30 degrees. The map in the top right shows the direction of the ligand's COM with respect to the COM of the receptor. The map on the top right shows the direction of the receptor's COM with respect to the ligand's COM. The map on the bottom right shows the average of the receptor and ligand maps.

When docking homodimers, it would be expected that the receptor and the ligand maps would be identical, as the docked homodimers should create a symmetric solution. This would negate the need for two PPI maps when docking homodimers. However, as seen in Figure 6, this is not the case; while the PPI maps are similar, they are not the same due to imperfect sampling. When docking geometrically, some degree of randomness can be expected in the identified solutions; it is computationally prohibitive to perfectly sample the possible solutions. The receptor and ligand maps in the homodimeric case can be averaged to produce a new map with reduced error, as seen in Figure 6. Because the ligand and receptor maps are ideally identical, studying these maps presents another advantage; the correlation between the maps can be used to optimize the size of each bin.

2.4.1. Optimization of Bin Size

Each square in a PPI map contains information about many calculated docking solutions binned over a range of angles. A question that arises is over what range docked poses should be grouped. Small degrees are subject to noise and randomness, while large degrees lose information about the topology of the docking solutions. This work uses three measures to optimize the bin size.

The first measure used is Pearson's correlation coefficient, a measure of the linear correlation between variables[15]. While the PPI maps themselves are not linear, a baseline correlation measure such as this gives an understanding of the actual differences between maps. The PCC, r, is calculated by the equation

$$\boldsymbol{r} = \frac{\sum(\boldsymbol{x_i} - \overline{\boldsymbol{x}})(\boldsymbol{y_i} - \overline{\boldsymbol{y}})}{\sqrt{\sum(\boldsymbol{x_i} - \overline{\boldsymbol{x}})^2 \sum(\boldsymbol{y_i} - \overline{\boldsymbol{y}})^2}}. \tag{1}$$

The second measure is the KL divergence, also referred to as relative entropy[16]. KL Divergence is a measure of how well one probability distribution is represented by another. It is defined by the equation

$$D_{KL}(P||Q) = \sum_{x}^{S} P(x)\, ln\left(\frac{P(x)}{Q(x)}\right). \tag{2}$$

Where P and Q are probability distributions and x is a variable in sample space S. Due to the scaled scores not being a complete probability distribution, the use of KL divergence in this work is not a truly representative measure. Its usage here, however, does allow us to compare differences in topologies of the maps, a factor that is ignored by Pearson's correlation coefficient due to its reliance on each variable's average.

The last measure used is the JS divergence, also known as the information radius. a measure related to, and calculated with, the KL divergence. It uses the same probability distributions as the KL Divergence, P and Q, as well as a mixture probability distribution, $M,$ defined by

$$M = \frac{1}{2}(P + Q). \tag{3}$$

The equation for the JS divergence is

$$D_{JS} = \frac{1}{2} D_{KL}(P||M) + \frac{1}{2} D_{KL}(Q||M). \tag{4}$$

Jenson-Shannon divergence is a symmetrized version of KL divergence that has been gives the distance between the KL divergence of both distributions, centered at their average[17]. While this loses some of the information about changes between the distributions, it is advantageous for two reasons. The first is that the value indicates differences between the distributions and the average values. As the best maps for the homodimers will be the maps created by averaging the receptor and ligand maps, the JS divergence presents a measure of the total topological changes between both maps and the average map.

The second reason involves cases where one of the distribution's bins has zero probability. In KL divergence, these bins will contribute an infinite term to the divergence, causing the whole divergence to approach zero. Due to this, this work neglects bins where either distribution has a zero when calculating the KL divergence. The JS divergence, on the other hand, can consider zero-probability bins, as the mixture distribution is never zero unless both bins are zero. In this case neglection of the bin is still used.

2.5. Dataset Splitting Based on Biophysical Properties

Consideration of new methods is dependent on the usage of those methods over a wide range of inputs; in the development of statistical methods, three datasets are commonly used. The first dataset, the training data, is used to generate the model. The second dataset, the validation data, is used to validate the parameters used to generate the model. The final dataset, the testing data, is used to evaluate the model in comparison to other methods. While the size of these datasets can vary, this work uses an ansatz common to statistical methods, splitting the data into training, validation, and testing sets containing 80%, 10%, and 10% of the data, respectively.

A variety of splitting measures have been used for biophysical datasets, most commonly randomly splitting, and scaffold splitting, which ensures that each dataset contains a variety of molecular "backbones"[18]. The use of these methods on proteins, however, often leads to different distributions of biophysical properties in the split datasets. This work considers a variety of biophysical properties, calculated by Ovidiu A. Petrisor, and splits the dataset using the multi-pivot QuickSort algorithm in the Python module Pandas as a method of retaining the distribution of biophysical properties in each dataset[19, 20].

2.6. Machine Learning

Due to the intricacies inherent in biophysical systems, whether it be due to the large size of molecules, the large number of different molecules, etc, there is a large desire to develop methods that reduce the complexity of problems and identify and reduce variables to the most important factors. In many cases, such as molecular docking, the physics of the systems are well known, but the calculations remain computationally expensive and time-consuming. Due to this, ML has become a standard tool in the study of biophysical interactions. The most famous use of this is

Google DeepMind's AlphaFold model, which has shown to predict protein folding exceedingly well[21]. Other use cases include identifying descriptors for specific methods of protein folding, and prediction of chemical properties from molecular dynamics data[22, 23].

2.6.1. Neural Networks

The prevailing ML model type are neural networks, which are modelled after the connections between brain neurons. A neural network is a series of parallel linear equations that convert a set of inputs to an output or set of outputs. To optimize these equations, a member of the training dataset is given to the model, and the information loss between the predicted output and the actual output is calculated using a method such as maximum likelihood, mean-squared error (MSE), or KL divergence. A gradient descent algorithm is then used to modify each variable to minimize the loss. This is iterated over all or part of the training dataset.

The build of the NN model creates another set of optimization parameters, which is corrected using a validation dataset. Some of the parameters that can be modified are the layer size, or the number of linear equations in parallel; the number of hidden layers, or the number of layers in series; the learning rate, or the step-size taken in each gradient descent iteration; and the type of equation used in each layer (commonly used equations include tanh, sigmoidal and rectified linear-unit equations). Two example NN architectures can be seen in Figure 7[24].

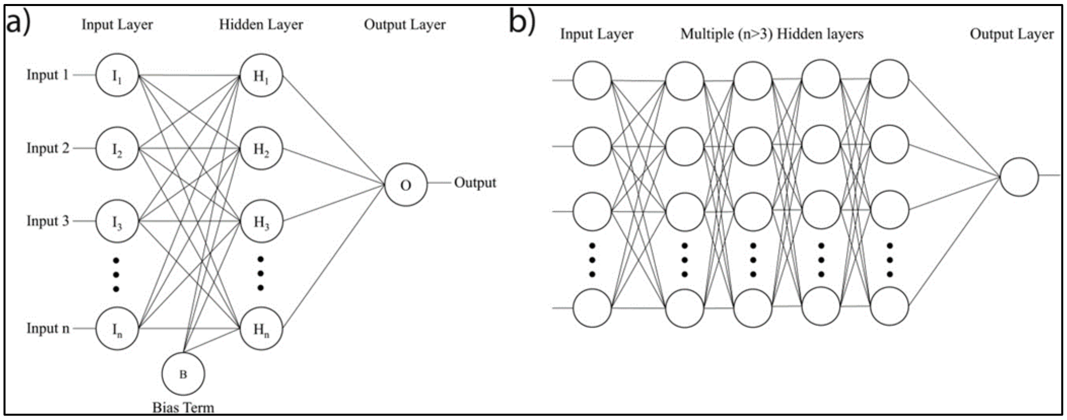


Figure 7. (a) An example NN with one hidden layer (b) An example of a NN with many hidden layers[24].

2.6.2. Machine Learning in this Work

This work constructs a simple NN for predictions of homodimer PPI maps from PDB files. A diagrammatic outline of the ML scheme used in this work is presented in Figure 10. This NN uses the docking outputs generated by this work as output labels. As inputs, this work created a featurization method for the conversion of a protein's PDB files into a 10,000 by 22 matrix representing the protein.

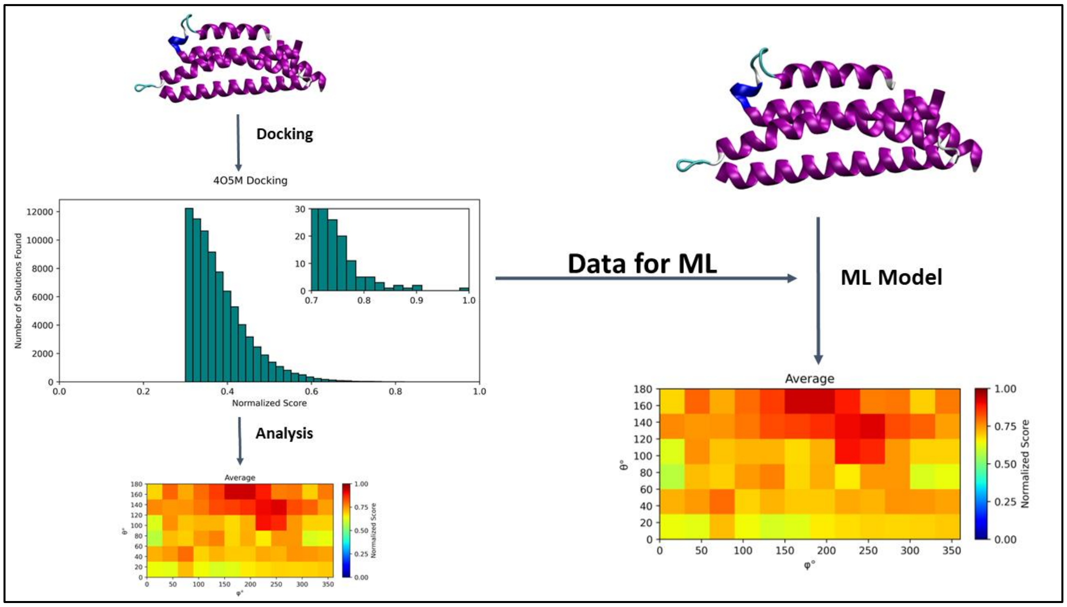


Figure 8. Diagrammatic illustration of the procedure for generating PPI maps using machine learning.

The first 19 columns are one-hot encoding representing the atom type of the protein, with indexes shown in Table 1. The first column is 1 if there is no atom, and 0 if there is. Columns 2 through 18 are labeled using the simplified indexes used in PatchDock, covering 133 different atom types, found in the chem.lib file[4, 5]. The residue type was omitted, and atom types found in several indexes due to different residue types were set to the lowest PatchDock index. This reducing the indexing required from 19 to 17 different indexes, as the 5 and 8 indexes contain only duplicates. The sulfur atom type, SE, which is not present in PatchDock, was added to index 17. Column 19 is 1 if the atom is hydrogen, which is omitted from consideration in PatchDock. The last three columns are the x, y, and z coordinates of the atom. For non-existent atoms, this is set to 999.9999 to label as omitted from consideration.

Several atom types present in the LOMETS3 dataset are not explicitly labeled in the PatchDock chem.lib file, instead being pseudo-labelled, such as CX atoms being considered by the C?? atom type. Both the pseudo-labelling and the explicit labelling added by this work are listed in Table 1.

This work constructs NN models using the PyTorch Python library[25]. Models consisted of three hidden layers, with layer sizes of either 360,180, 72 nodes, or a larger model of 720, 360, 72 nodes. These layers take the form of a rectified linear unit (ReLu), defined as

$$f(x) = \begin{cases} x \\ 0 \end{cases} \quad \text{if} \quad \begin{cases} x > 0 \\ \text{otherwise}' \end{cases} \tag{5}$$

or a sigmoidal function,

$$f(x) = \frac{1}{1+e^{-x}}. \tag{6}$$

Models were trained using two different loss functions. The first loss function is the MSE loss function provided in the PyTorch library. The second loss function was a custom JS divergence loss function for the calculation of the JS divergence of non-probabilistic arrays. Learning rates for models were varied as 0.01, 0.005, and 0.001.

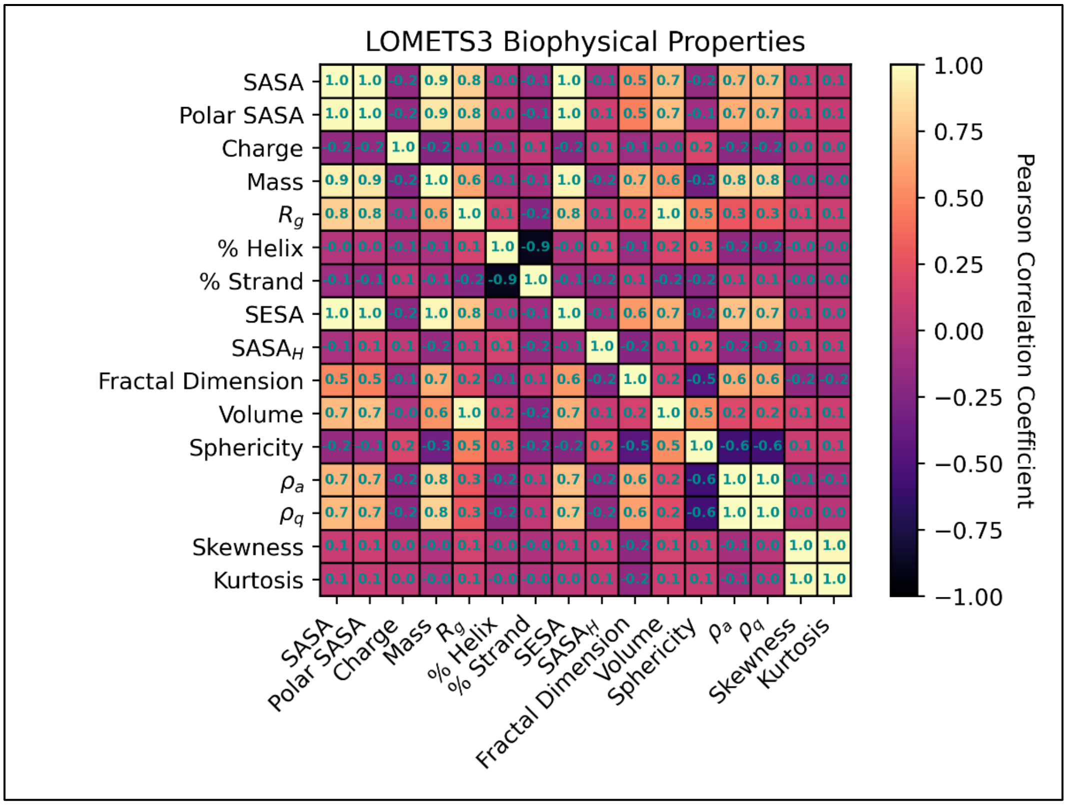
Table 1. One-hot encoding indexes used for PDB featurization.

Index	PatchDock Atom Type Index	Atom Types
0	N/A	No atom
1	0	XXX
2	1	N, N3B, N??
3	2	CA, CHD, CHC, CHA, CHB, CAB, CAC
4	3	C, CX, C3C, C2D, C3D, C2B, C3B, C2C, C4B, C1D, C4C, C2A, C3A, PA, C4A, C1C, C1A, C4D, C1B, PB, C??
5	4	O4*, O2B, O5*, O1A, O2A, O, O6, O3A, O1B, O??, OQ1, OQ2, OP1, OP2, OP3, OP4
6	6	C1, C2, C3, C4, C5, C1’, C2’, C3’, C4’, C5’, C1*, C2*, C3*, C4*, CB, CBD, CBA, CBB, CAA, CBC, CAD, CD, C5*, CG
7	7	NZ, N2, CE
8	9	OE, OE1, OXT, OT2, OE2, O1D, CGD, OD1, O3G, O2G, O1G, PG, OD2, OT1, P??, O3B, O3, CGA, O4, O2, O1, S, O2D
9	10	F??, BR?, FE?, MG?, CAL, FE, E, NE2, ND1, CZ, NH1, NH2
10	11	ND2
11	12	NE
12	13	OG1, OD, O2*, OG, OH, O3*
13	14	N9, N1, CE1, CD2, C6, NE1, C8, N7, C2, N3, C4, C5, NB, ND, NA, NC
14	15	CE2
15	16	CZ3, CH2, CE3, CG2, CZ2, CD1, CG1, SD
16	17	CMD, CMB, CMC, CH1, CMA, CH3
17	18	SG, S??, SE
18	N/A	All hydrogens

3. Results and Discussion

3.1. Dataset Analysis

Figure 11 presents a heatmap of the PCC between various biophysical properties of the LOMETS3 dataset, calculated by Ovidiu A. Petrisor[19]. Using this, four properties with low PCCS in relation to each other were selected to split the LOMETS3 dataset into training, validation, and testing sets. The four properties selected were the charge, radius of gyration, SASA$_H$, and the arithmetic roughness.



	SASA	Polar SASA	Charge	Mass	R_g	% Helix	% Strand	SESA	SASA$_H$	Fractal Dimension	Volume	Sphericity	ρ_a	ρ_q	Skewness	Kurtosis
SASA	1.0	1.0	-0.2	0.9	0.8	-0.0	-0.1	1.0	-0.1	0.5	0.7	-0.2	0.7	0.7	0.1	0.1
Polar SASA	1.0	1.0	-0.2	0.9	0.8	0.0	-0.1	1.0	0.1	0.5	0.7	-0.1	0.7	0.7	0.1	0.1
Charge	-0.2	-0.2	1.0	-0.2	-0.1	-0.1	0.1	-0.2	0.1	-0.1	-0.0	0.2	-0.2	-0.2	0.0	0.0
Mass	0.9	0.9	-0.2	1.0	0.6	-0.1	-0.1	1.0	-0.2	0.7	0.6	-0.3	0.8	0.8	-0.0	-0.0
R_g	0.8	0.8	-0.1	0.6	1.0	0.1	-0.2	0.8	0.1	0.2	1.0	0.5	0.3	0.3	0.1	0.1
% Helix	-0.0	0.0	-0.1	-0.1	0.1	1.0	-0.9	-0.0	0.1	-0.1	0.2	0.3	-0.2	-0.2	-0.0	-0.0
% Strand	-0.1	-0.1	0.1	-0.1	-0.2	-0.9	1.0	-0.1	-0.2	0.1	-0.2	-0.2	0.1	0.1	-0.0	-0.0
SESA	1.0	1.0	-0.2	1.0	0.8	-0.0	-0.1	1.0	-0.1	0.6	0.7	-0.2	0.7	0.7	0.1	0.0
SASA$_H$	-0.1	0.1	0.1	-0.2	0.1	0.1	-0.2	-0.1	1.0	-0.2	0.1	0.2	-0.2	-0.2	0.1	0.1
Fractal Dimension	0.5	0.5	-0.1	0.7	0.2	-0.1	0.1	0.6	-0.2	1.0	0.2	-0.5	0.6	0.6	-0.2	-0.2
Volume	0.7	0.7	-0.0	0.6	1.0	0.2	-0.2	0.7	0.1	0.2	1.0	0.5	0.2	0.2	0.1	0.1
Sphericity	-0.2	-0.1	0.2	-0.3	0.5	0.3	-0.2	-0.2	0.2	-0.5	0.5	1.0	-0.6	-0.6	0.1	0.1
ρ_a	0.7	0.7	-0.2	0.8	0.3	-0.2	0.1	0.7	-0.2	0.6	0.2	-0.6	1.0	1.0	-0.1	-0.1
ρ_q	0.7	0.7	-0.2	0.8	0.3	-0.2	0.1	0.7	-0.2	0.6	0.2	-0.6	1.0	1.0	0.0	0.0
Skewness	0.1	0.1	0.0	-0.0	0.1	-0.0	-0.0	0.1	0.1	-0.2	0.1	0.1	-0.1	0.0	1.0	1.0
Kurtosis	0.1	0.1	0.0	-0.0	0.1	-0.0	-0.0	0.0	0.1	-0.2	0.1	0.1	-0.1	0.0	1.0	1.0

Figure 9. PCC between the biophysical properties of the LOMETS3 dataset calculated by Ovidiu A. Petrisor[19].

Once the properties were selected, the property distributions were plotted against each other to create a visual identification of the dataset splitting. This is displayed in Figure 12. As can be seen in this dataset, while there is a wide distribution between most of the properties, the $SASA_H$ has a tight distribution, concentrated between 0.4 and 0.6, except for one outlier. Inclusion of the $SASA_H$ as a splitting measure can then be used to coral outliers in the other properties into a better split dataset.

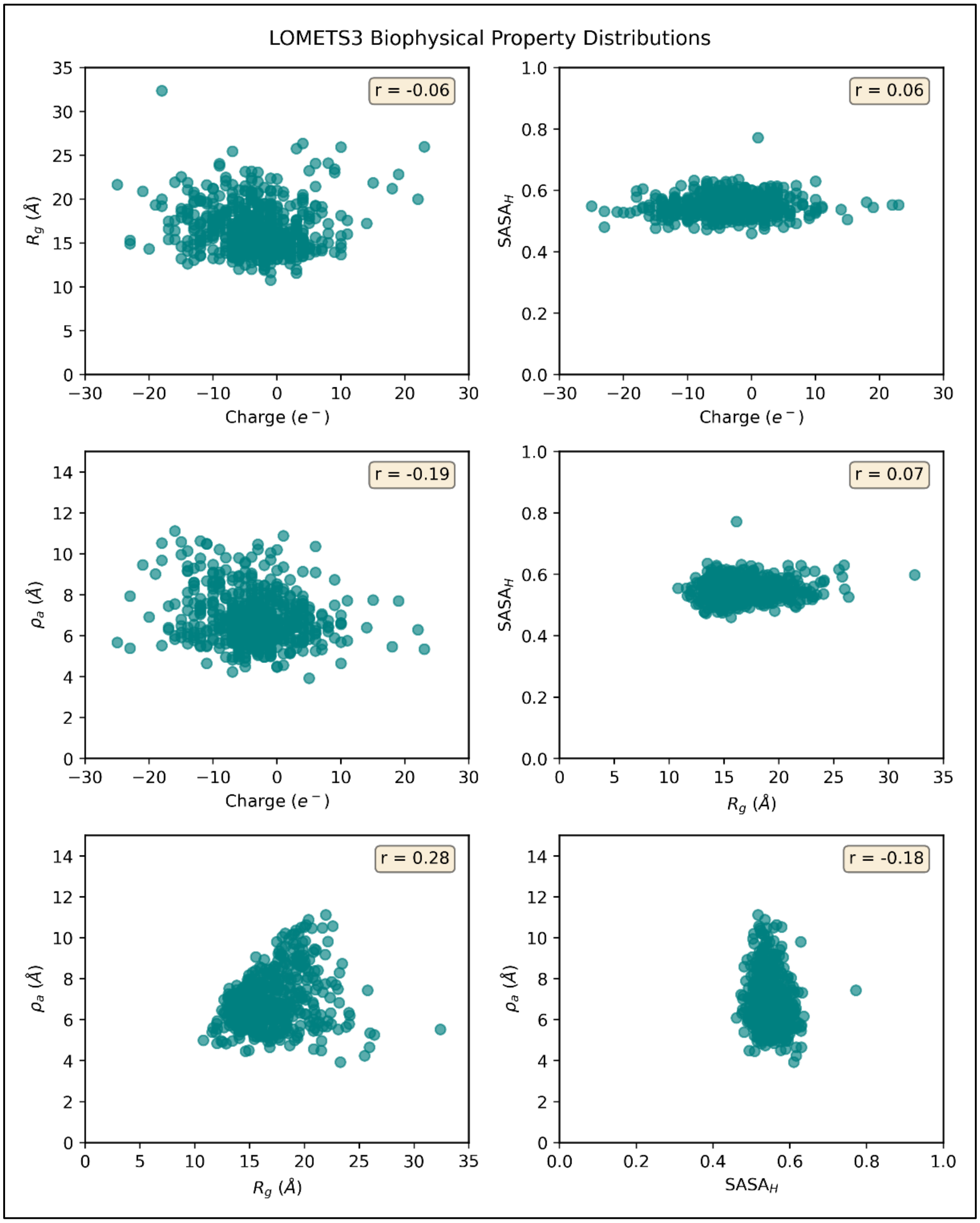


Figure 10. Distribution between select biophysical properties of the LOMETS3 dataset. The PCC between the properties is displayed in the top right corner.

3.2. Biophysical Property Sampling

The dataset was split into training, validation, and testing data, using the commonly used ansatz of 80-10-10. The Python module Pandas was used to perform a multi-pivot quicksort of the four biophysical properties chosen for splitting, the overall charge, radius of gyration, $SASA_H$, and the arithmetic roughness[20]. After sorting, 2 out of every 10 proteins were chosen, with 1 sent to the validation dataset, and the other sent to the testing dataset. The distributions of the split data are shown in Figure 13. As can be seen in the figure, the training data (blue), validation data (orange), and testing data (green) hold distributions close to those of the whole dataset seen in Figure 12.

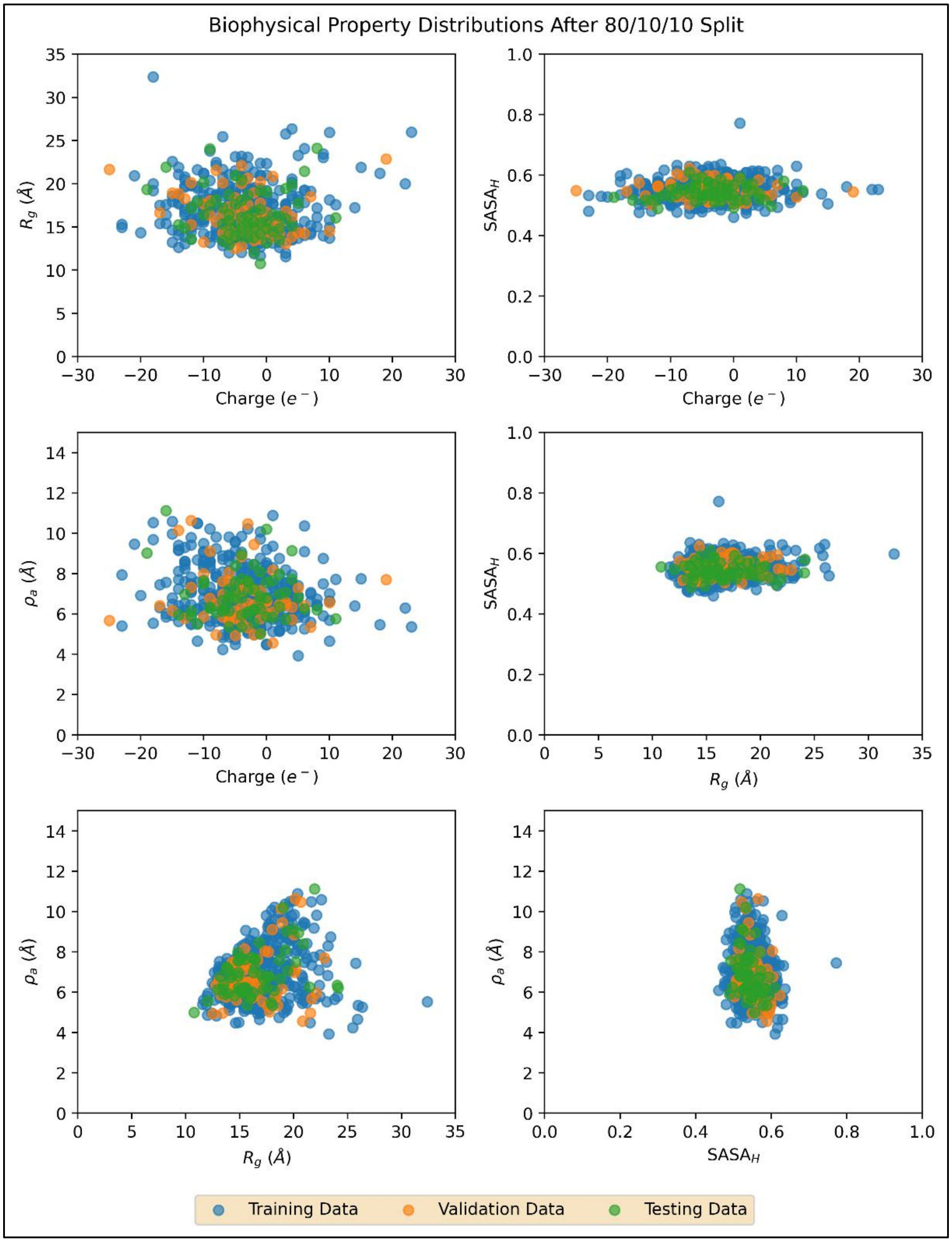


Figure 11. 80-10-10 split of the training (blue), validation (orange), and testing (green) datasets.

3.3. Bin Size Optimization

Figure 11 displays the average PCC of the PPI maps in 5° increments, starting at 5° and ending at 180°. The PCC rapidly increases to a value just over 0.8, with convergence between resolutions of 30° and 45°. The divergence not reaching a perfect value of 1 is due to the PPI maps being non-square. At the maximum resolution there are still two bins spanning over the same θ values, $[0,180]$, but different ϕ values, $[0,180]$ and $[180, 360]$.

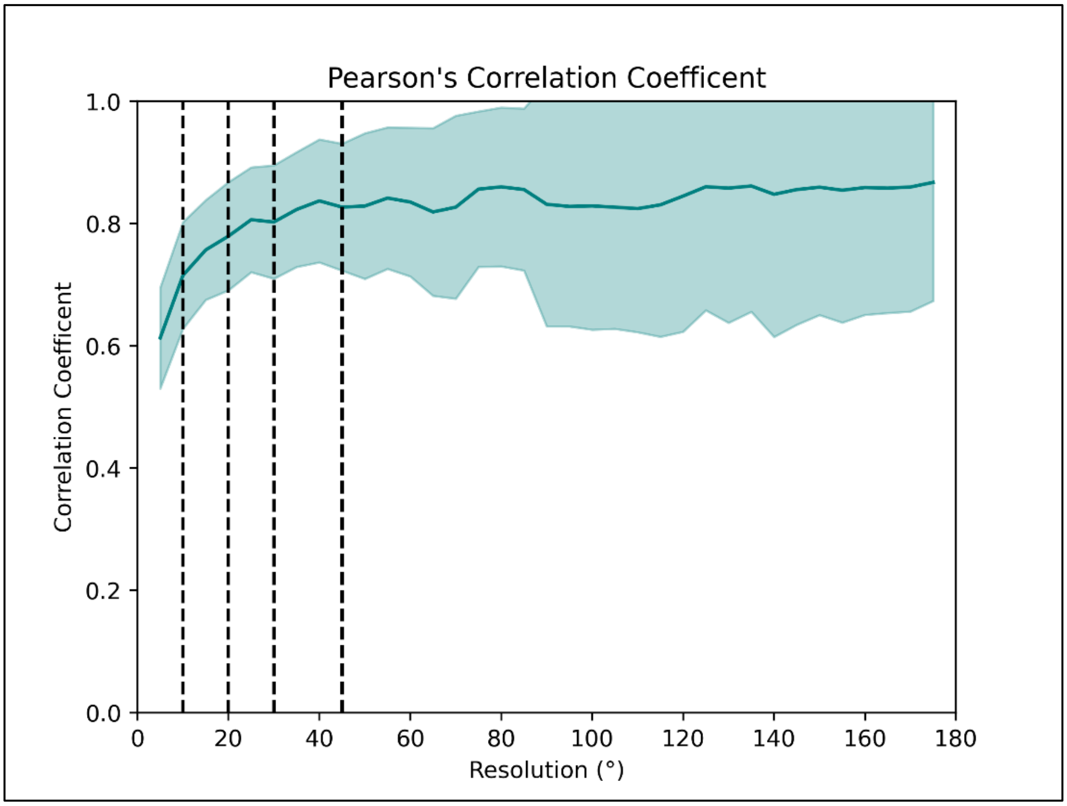


Figure 12. Pearson's correlation coefficient of ligand and receptor maps at 5° increments, averaged over the entire dataset, as a function of the bin length. The black lines indicate 10, 20, 30, and 45°.

The KL divergence of the receptor and ligand maps is plotted in Figure 12. While the KL divergence is not a symmetric measure, the divergence of the maps from each other is close enough to appear so in the figure. As with the PCC, the KL divergence rapidly decreases to a minimum value instead of a perfect score, in this case 0.001. The KL divergence converges at a larger resolution than the PCC, between 45° and 60°, indicating that there are still some topological distinctions between those two points, even the linear trends in the PPI maps are well coordinated.

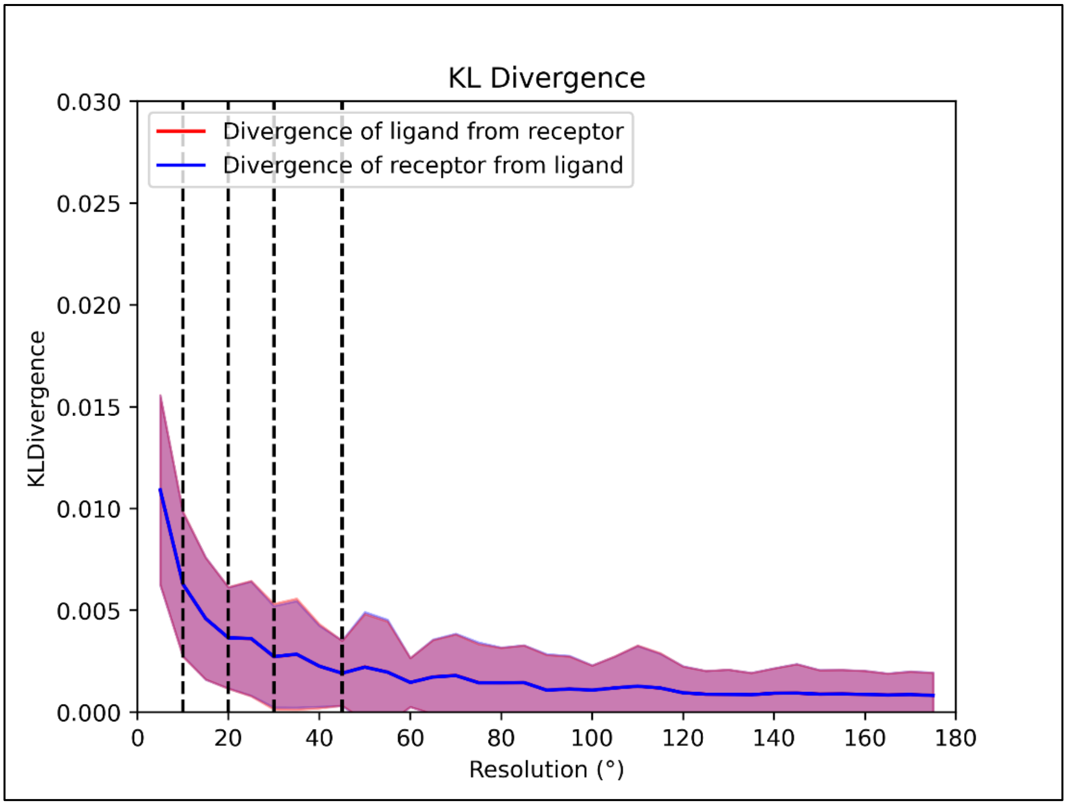


Figure 13. KL divergence of the (red) ligand map from the receptor map and (blue) the divergence of the receptor map from the ligand map, averaged over the entire dataset, as a function of the bin length.

Analysis of the JS divergence, shown in Figure 13, reveals that use of the average distribution of the maps yields little difference to either of them, as even at the smallest resolution the JS divergence is smaller than the KL divergence. These differences are more pronounced at higher resolutions relative to the lower distributions, with the JS divergence having the largest convergence resolution of the analysis measures at 90°.

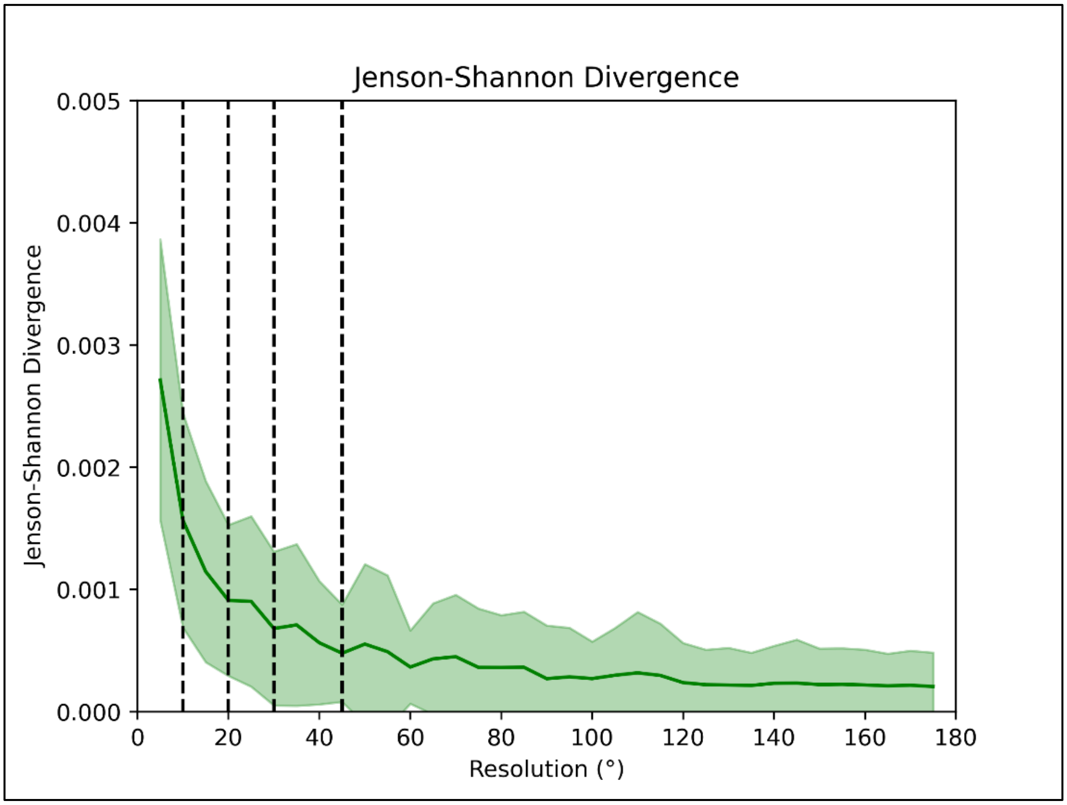


Figure 14. JS divergence of the receptor and ligand maps, averaged over the entire dataset, as a function of bin length. The black lines indicate 10, 20, 30, and 45°.

This work proposes an optimal bin size of 30°. This bin size allows for high correlation of linear trends between the two maps, while also conserving some of the noise from the imperfect sampling present in docking programs. While data noise is best removed in most cases, it is useful for creations of PPI maps, as it allows orientations of the ligand and receptor molecules to be considered over a larger range in the map than a perfectly square search space.

3.4. Docking Results

To ensure that the molecular docking and PPI maps methods used hold across proteins with various properties, this work analyzes the proteins with the smallest, the most average (or a neutral protein in the case of charge), and the largest value of each of the biophysical properties for splitting, excluding outliers. Additionally, this work performed the same analysis for protein masses, as one goal of improving methods for PPI docking is better consideration of large proteins.

3.4.1. PPI Maps and Charge

It's important to note that the high variance protein structure leads to a corresponding wide variance in overall protein charge, with a tendency away from neutrality. Due to this it is common to represent the electrostatic properties of proteins using more complex factors, including multipole moments, isoelectric points, and surface charge densities[26, 27]. This work chooses to use consider the overall charge as a biophysical factor for simplicity and ease of calculation. Further consideration of higher order electrostatic factors may lead to altered understandings of the relationships between electrostatic properties of proteins and PPI maps

While the overall charge of a molecule has important effects for docking, this is not something that can be used to predict the effect of charge on the PPI map, as can be realized from the above mentioned of higher order factors. PPI maps are not expected to have a large change from differences in overall charges, and changes in the maps are more likely to be better attributed to other biophysical factors. These biophysical properties, however, may have some relationship to the charge. The radius of gyration and sphericity, for example, are expected to increase with the magnitude of charge, as the unpaired charges repel each other, leading to a rod-like shape.

In Figure 17, the PPI maps for the most negative, most positive, and a neutral protein examine this. The most negative protein, 1EXR chain A, has a representatively average radius of gyration, which leads to an expected elliptical shape of high scoring regions, which is seen in the PPI map. The neutral protein, 1LXJ, has a small radius of gyration, with a slight extension along one axis, leading to the expectation of a slightly squeezed spherical zone of high scoring regions, which agrees with the generated PPI map. The most positive protein, 2J01, has a high radius of gyration, so the expectation, which is seen in the PPI map, is a highly heterogeneous map with a spread of high scoring zones.

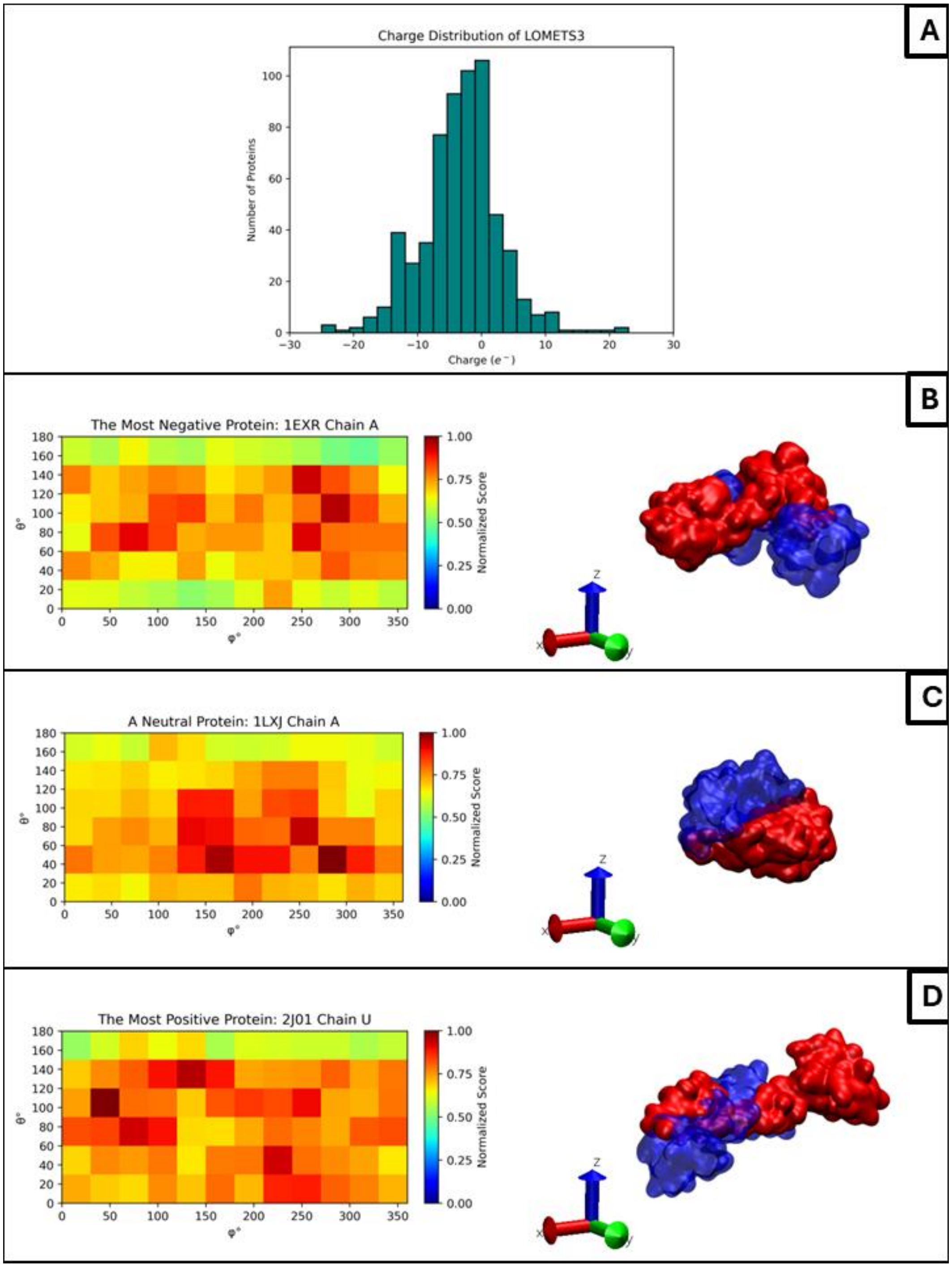


Figure 15. (A) The distribution of protein charges in the LOMETS3 dataset. The PPI map and highest scoring docking pose for (B) the most negative protein, chain A of 1EXR, (C) a neutral protein, chain A of 1LXJ, (D) the most positive protein, chain U of 2J01.

3.4.2. PPI Maps and Radius of Gyration

The radius of gyration of a protein is a multifold descriptor of other protein biophysical properties. The radius of gyration increases with protein size and decreases with protein sphericity. Due to this, the radius of gyration was selected as a singular parameter in the dataset splitting indicative of both properties. The expectation of PPI maps as the radius of gyration increases is a transformation from a circular spread of high scoring zones at low radii, as the contact is relegated to the small concurrent surface interactions between spherical particles, to a more oblong spread, from larger rod-like particles with an increased amount of surface contact leading to a variety of interactions along the length of the surface contacts.

Figure 18 contains PPI maps for the lowest, an average, and the highest non-outlier radius of gyration proteins. Comparison of these figures shows that as the radius of gyration increases, the spread of high docking scores changes from a contained spread, seen center of the 2BCR PPI map, to a rectangular spread, in the 2IV2 PPI map, to a non-regular spread in the 3UMH PPI map, holding with the expectation of PPI maps as the radius of gyration increases.

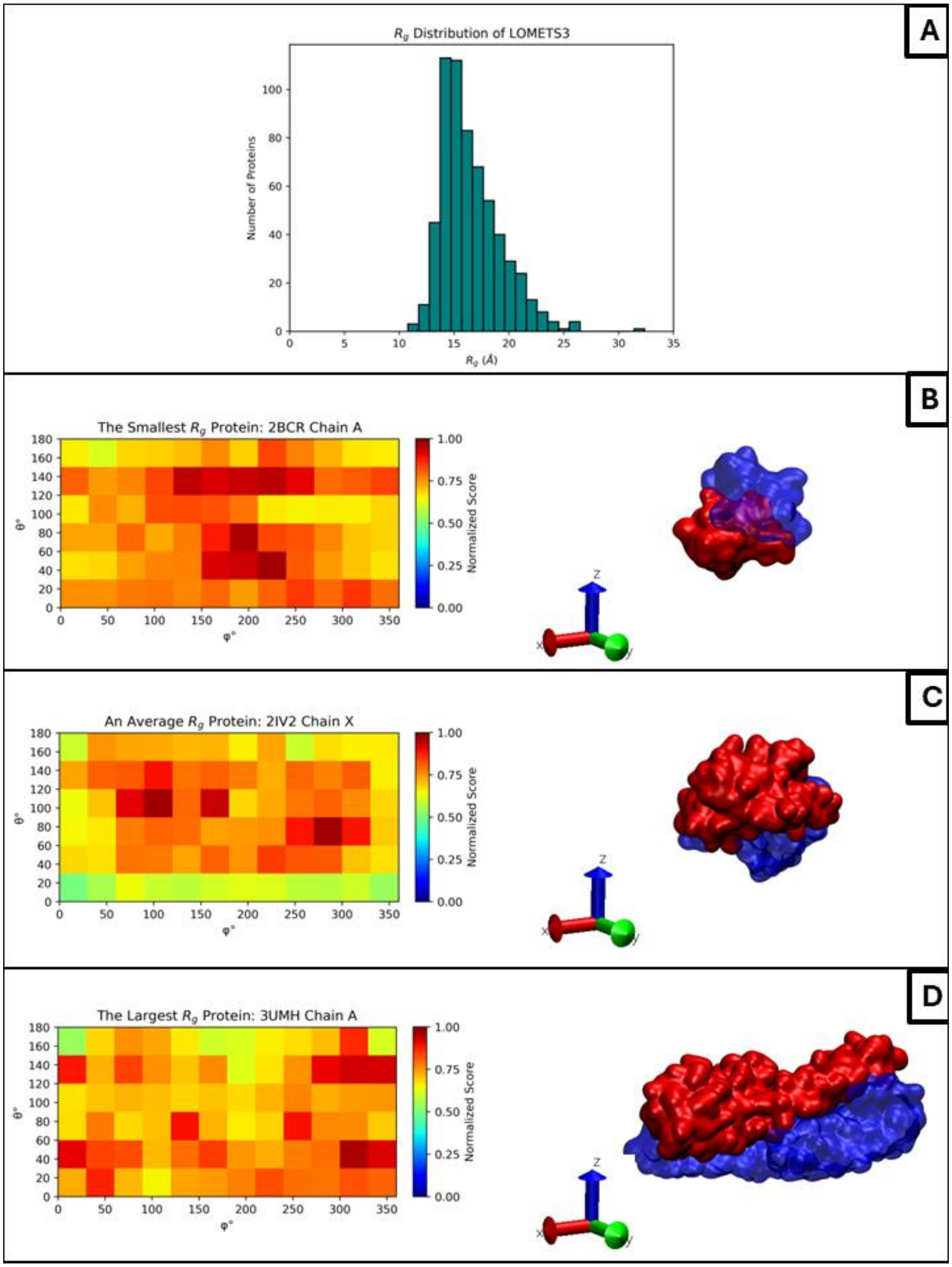


Figure 16. (A) The distribution of protein radius of gyration in the LOMETS3 dataset. The PPI map and highest scoring docking pose for (B) The smallest radius of gyration, chain A of 2BCR, (C) an average radius of gyration, chain X of 2IV2, (D) The largest non-outlier radius of gyration, chain A of 3UMH.

3.4.3. PPI Maps and SASAH

Hydrophobic patches on protein surfaces have low contact energies with polar, positive, and negative surface patches, and are thus high likelihood candidates for PPI interaction sites[28]. This leads to the prediction that proteins with a low $SASA_H$ will have a PPI map with a large amount of variation between scores, as there of areas with a mixture of contact areas. A high $SASA_H$, on the other hand, would indicate a homogeneous PPI map, as much of the surface area will have low contact energies.

The PPI maps shown in Figure 15, with the lowest, an average, and the highest non-outlier $SASA_H$ proteins, show this expectation holding true with PPI maps generated from docking the LOMETS3 dataset. An important note on the PPI map of the highest $SASA_H$ protein, 2PD1, is the highest scoring region outperforming the rest of the map by a large margin. Analysis of the surface residue types in this pose, seen in Figure 20, shows that the interface contains a large amount of matching non-polar (hydrophobic) groups, as well as paired acidic and basic groups, indicating that the docking pose itself is a good match. Further investigation of this was considered outside the scope of this work, with difficulty arising from the absence of a publication describing the protein in the PDB.

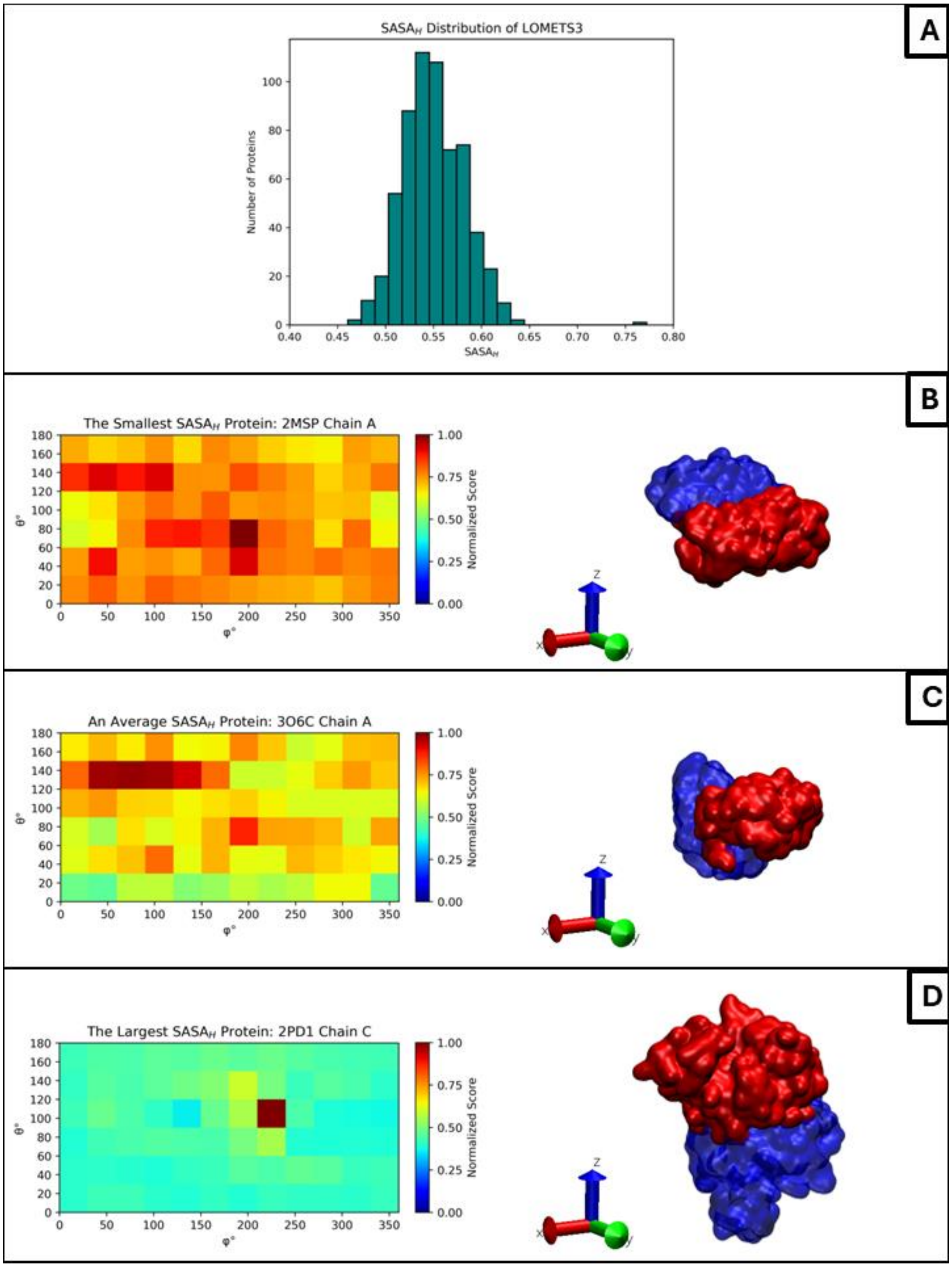


Figure 17. (A) The distribution of protein SASA_H in the LOMETS3 dataset. The PPI map and highest scoring docking pose for (B) The smallest SASA_H, chain A of 2MSP, (C) an average SASA_H, chain A of 3O6C, (D) The largest non-outlier SASA_H, chain C of 2PD1.

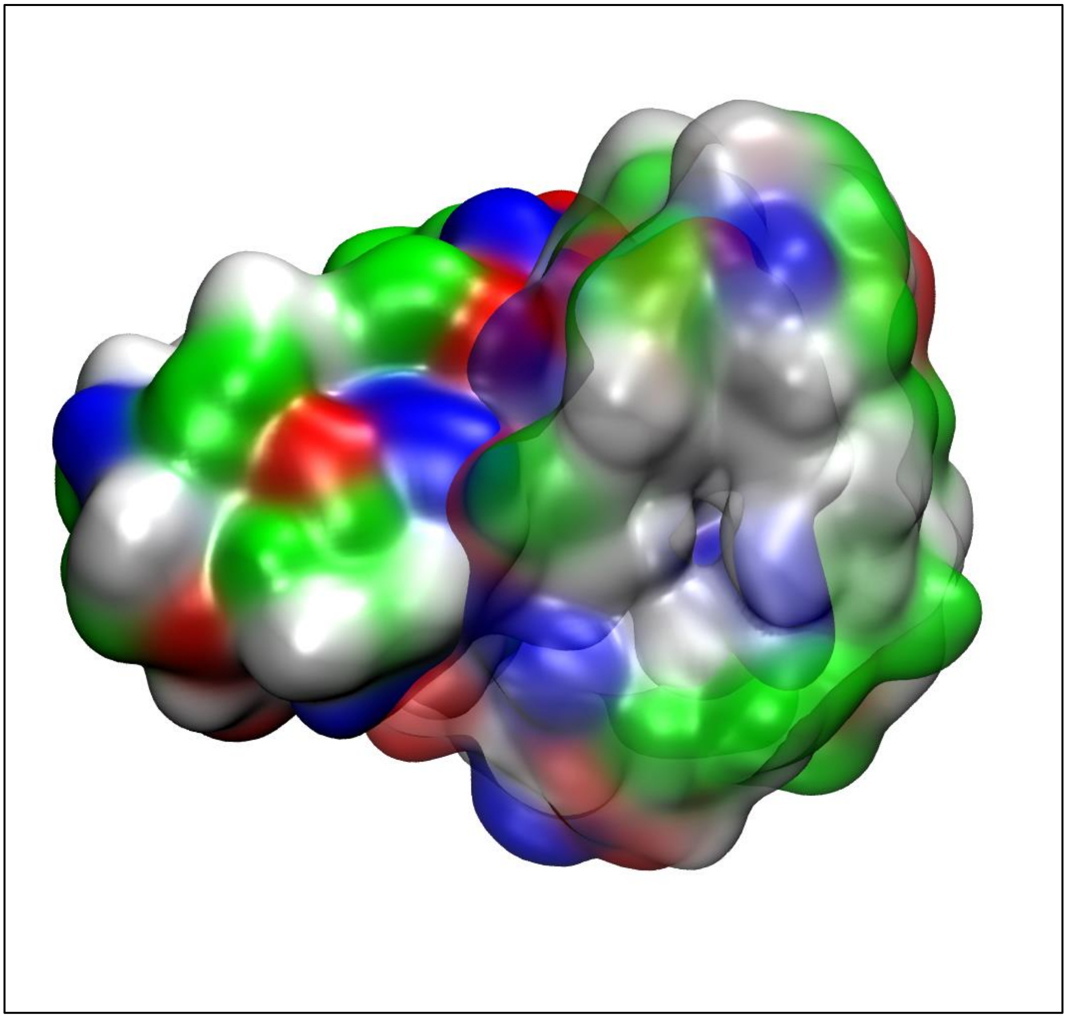


Figure 18. Homodimer of the 2PD1 chain C protein, visualize using the residue type of the protein surface. The interface between proteins contains a high amount of matching non-polar groups (white) and paired acidic and basic groups (red and blue, respectively).

3.4.4. PPI Maps and Arithmetic Surface Roughness

As the name indicates, surface roughness is a measure of the topological differences of a protein surface, specifically its peaks and valleys. Complex topological areas on macromolecular surfaces are indicative of possible binding sites in that region[29]. Smooth surfaces tend to be poor binders for other molecules, whether macromolecular or small, while rough surfaces allow for easy capture, and more stabilizing interactions between specific regions of the molecules, such as binding pockets for small molecules.

There are many commonly used methods for analyzing surface roughness, such as the fractal dimension, arithmetic or mean roughness, and root-mean square (RMS) roughness, which are commonly used in conjunction to analyze properties of the protein surface. The fractal dimension indicates the complexity of surface patches on the protein. The RMS roughness is an indicator of whether there are large peaks or valleys on the surface. The arithmetic roughness is an indicator of whether there are many peaks or valleys on the surface. In this work, the arithmetic surface roughness was chosen to split the dataset, as it contains more general knowledge about the protein surface instead of the magnitude and differences in the surface topology.

The expectation for increasing surface roughness is that a rougher surface would have more high scoring regions, but the variation between neighbors would be more varied overall. Figure 16, which displays PPI maps for the smoothest, the roughest, and an average roughness protein, shows that this expectation is met.

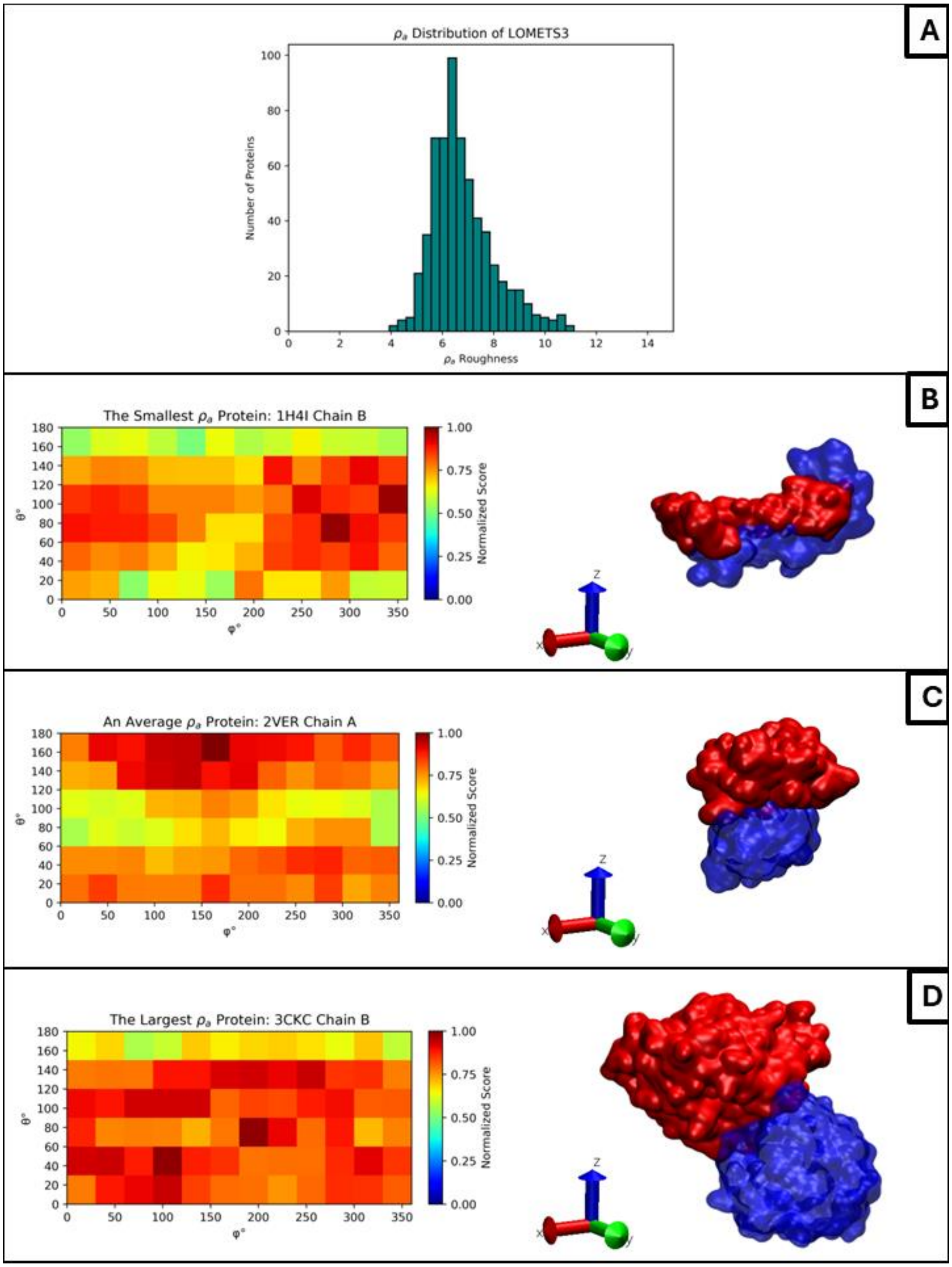


Figure 19. (A) The distribution of protein arithmetic roughness in the LOMETS3 dataset. The PPI map and highest scoring docking pose for (B) The smoothest protein, chain B of 1J4I, (C) an average roughness, chain A of 2VER, (D) The roughest protein, chain B of 3CKC.

3.4.5. PPI Maps and Mass

The last analysis biophysical property analyzed is the mass, which was considered as a factor for dataset splitting. While protein mass is indicative of a larger number of amino acids, and thus a larger possibility of variety in the biophysical factors, the property itself does not give information about these factors. The mass is thus considered only as an analysis of the PPI maps, and not of the variance of the LOMETS3 dataset. While a higher mass allows for the possibility of a higher heterogeneity on PPI maps, it doesn't contain any knowledge of this without additional information, besides an increase in radius of gyration, like overall protein charge. However, unlike the charge, trends between the mass and other biophysical properties are much more difficult to consider, as it has no significant relationship to the types of amino acids. As such, PPI maps for proteins of increasing weight are expected to have no clear trend of hetero or homogeneity in the maps.

Figure 17 displays this expectation in the PPI maps generated by the LOMETS3 dataset, using the lightest, an average mass, and the heaviest protein in the dataset. While these proteins have an increasing radius of gyration, they are all generally spherical. As such, each map has a close-to-circular zone where the score decreases as the distance from the center increases. As the radius of gyration increases, these zones increase in size.

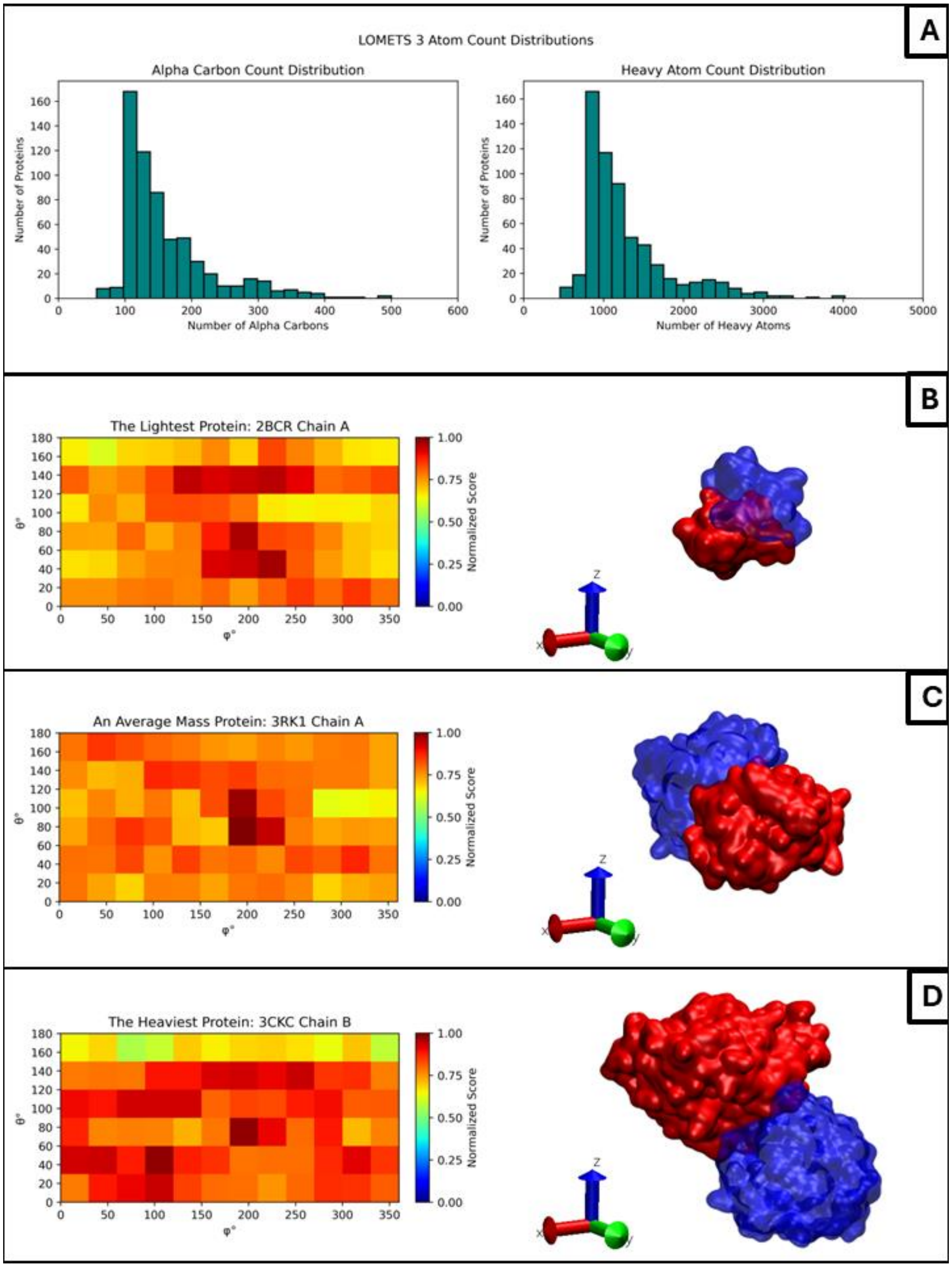


Figure 20. (A) The distribution of protein mass, calculated by the number of alpha carbons and the number of heavy atoms in the LOMETS3 dataset. The PPI map and highest scoring docking pose for (B) The lightest protein, chain A of 2BCR, (C) an average mass protein, chain A of 3RK1, (D) The most massive protein chain B of 3CKC.

3.5. Machine Learning Analysis

Figure 21 shows the training learning loss for the MSE models. Notably, the models did not show signs of fitting better to the data after more epochs, instead converging after a single epoch. Models with two sigmoidal functions vastly outperformed the two ReLu function models. Switching the first layer to another sigmoidal function did not show signs of improvement, except in the high learning rate case. The lowest average MSE on the validation dataset, as can be seen in Table 2, was the low learning rate model with three sigmoidal functions.

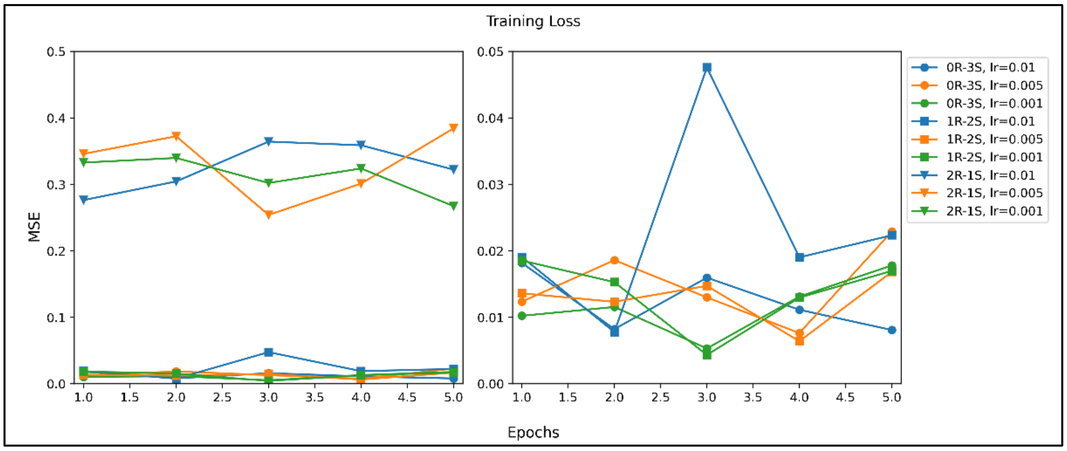


Figure 21. Learning loss on the training dataset calculated for the final member of each epoch. The figure on the right presents the same information on a finer scale for better viewing of the 2 sigmoidal and 3 sigmoidal training loss. Circle markers represent models with 3 sigmoidal layers; square markers represent models with 1 ReLU layer followed by 2 sigmoidal layers. Triangular markers represent models with 2 ReLU layers followed by a sigmoidal layer. Blue, orange, and green lines represent learning rates of 0.01, 0.005, and 0.001, respectively.

Table 2. Average MSE of the Validation Dataset After Training

	Learning Rate = 0.01	Learning rate = 0.005	Learning Rate = 0.001
2 ReLu, 1 Sigmoid	0.3183	0.3183	0.3183
1 ReLu, 2 Sigmoid	0.0371	0.214	0.0185
0 ReLu, 3 Sigmoid	0.0192	0.0189	0.0184

Two test PPI maps generated by the three sigmoid, low learning rate model are presented in Figure 22. As can been seen in the Figure, the model does not generate PPI maps based on the input data. Instead, the model has learned a generalized best representation for all proteins in the dataset. This result also occurs in the models with ReLu functions, as can be seen in Figure 23, which shows maps generated by a 2-ReLu 1-Sigmoid model for the same proteins.

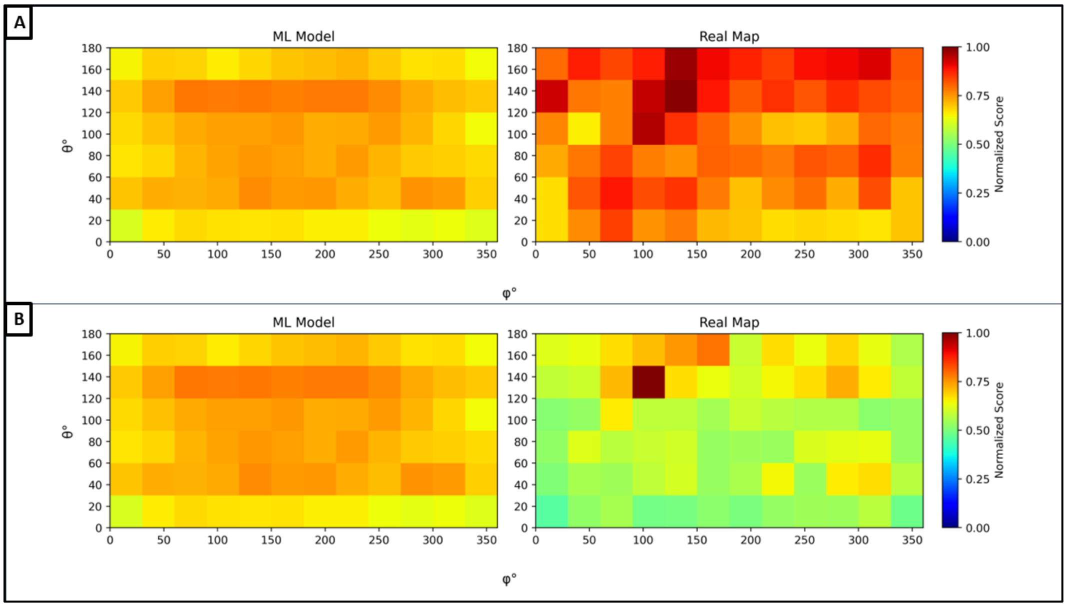


Figure 22. PPI maps (left) generated by the 3 Sigmoidal model with a learning rate of 0.001 using a MSE loss function compared to (right) PPI maps generated by PatchDock for (A) 3PMG chain A and (B) 1JG8 chain D.

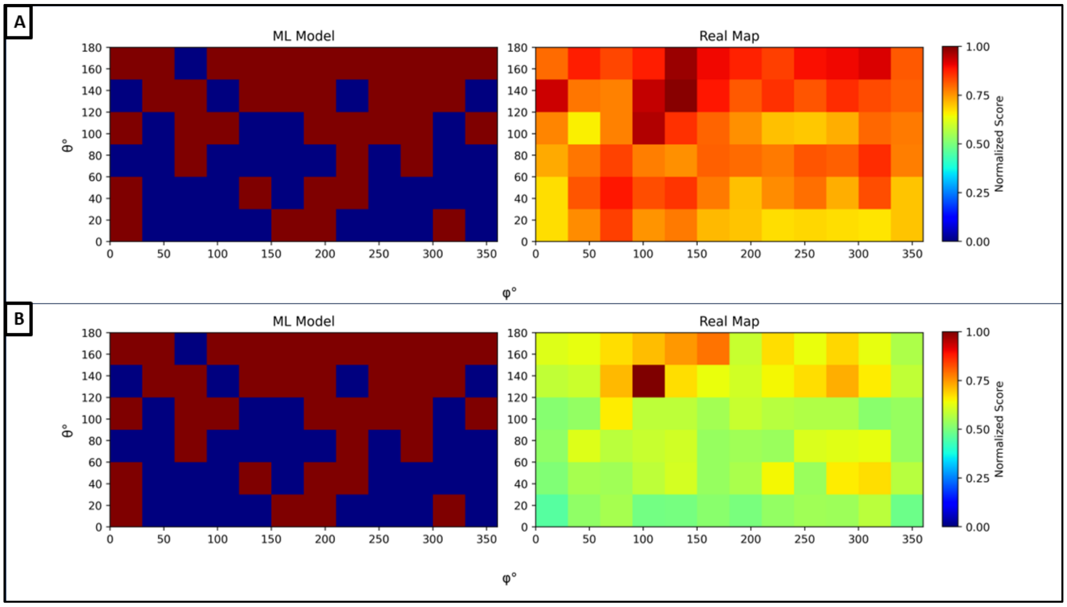


Figure 23. PPI maps (left) generated by the 2-ReLU, 1 Sigmoidal model with a learning rate of 0.001 using a MSE loss function compared to (right) PPI maps generated by PatchDock for (A) 3PMG chain A and (B) 1JG8 chain D.

In an attempt to prevent the model from reaching an overgeneralized solution, the same model parameters were used with a JS divergence loss function, due to the JS divergence being more specified to the specific matrix coordinate differences value. Figure 24 shows the training loss for 3-sigmoidal models trained using a JS divergence loss function. The training loss was minimized at 3 epochs. The 3-sigmoid, low learning rate model's output for the same chains is shown in Figure 25. Once again, a generalized representation can be seen. In this case, most of the values are around a scaled score of 0.5, with an I shape centered in the middle of the map containing higher scores.

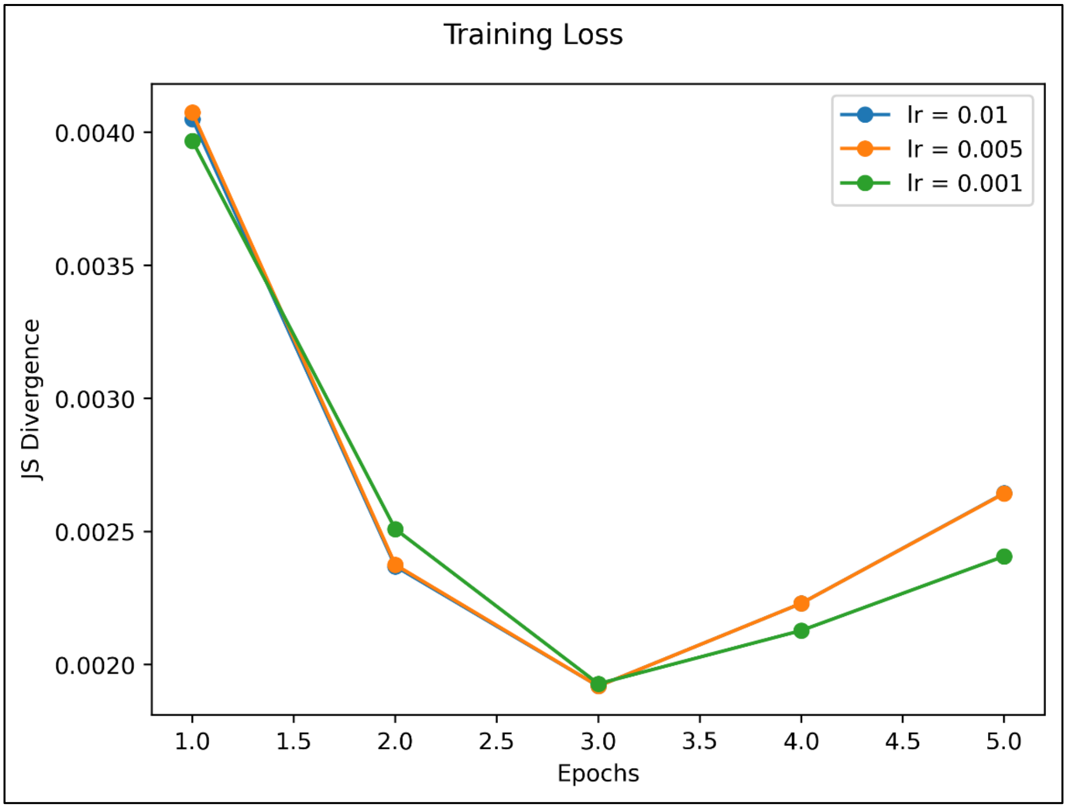


Figure 24. Training loss for the final member of each epoch for the 3-sigmoidal model using a JS divergence loss function. Blue, orange, and green lines represent learning rates of 0.01, 0.005, and 0.001.

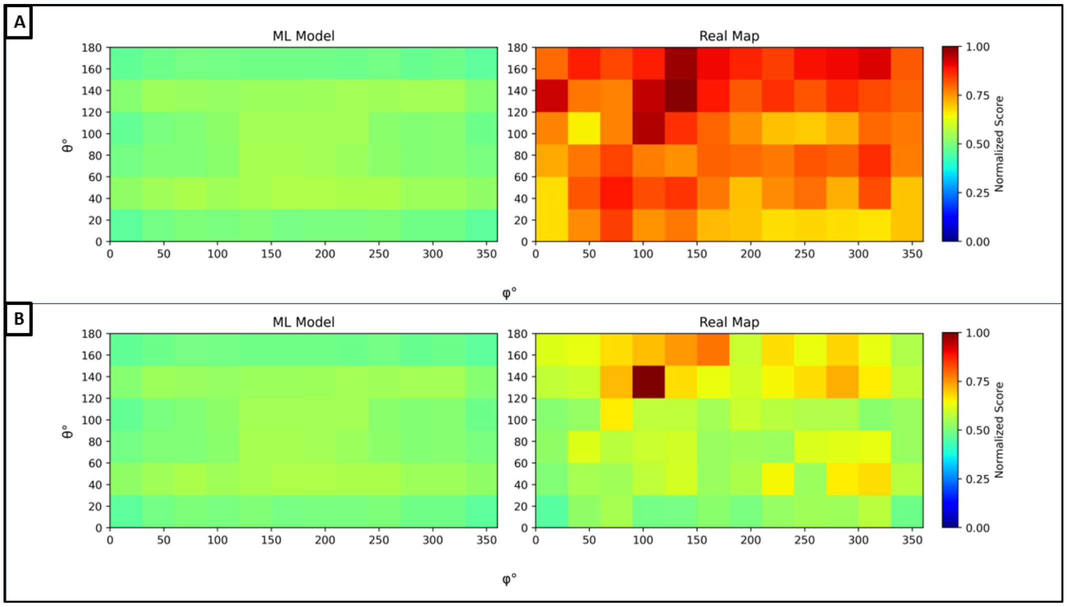


Figure 25. PPI maps (left) generated by the 3 Sigmoidal model with a learning rate of 0.001 using a JS divergence loss function compared to (right) PPI maps generated by PatchDock for (A) 3PMG chain A and (B) 1JG8 chain D.

The next method used to try to lessen the model's generalization was increasing the number of nodes used in the model's hidden layers. The number of nodes layer 1 was increased to 720, and the number of nodes in layer 2 was increased to 360. The 3-sigmoid, high learning rate outputs for the same proteins previously considered are shown in Figures 26 and 27, for model's trained using a MSE and a JS divergence loss rate, respectively. While these models still overgeneralized, the differences between the high and low scores increase in these models, causing the general shape of the higher scoring zones to appear more stark in the PPI maps.

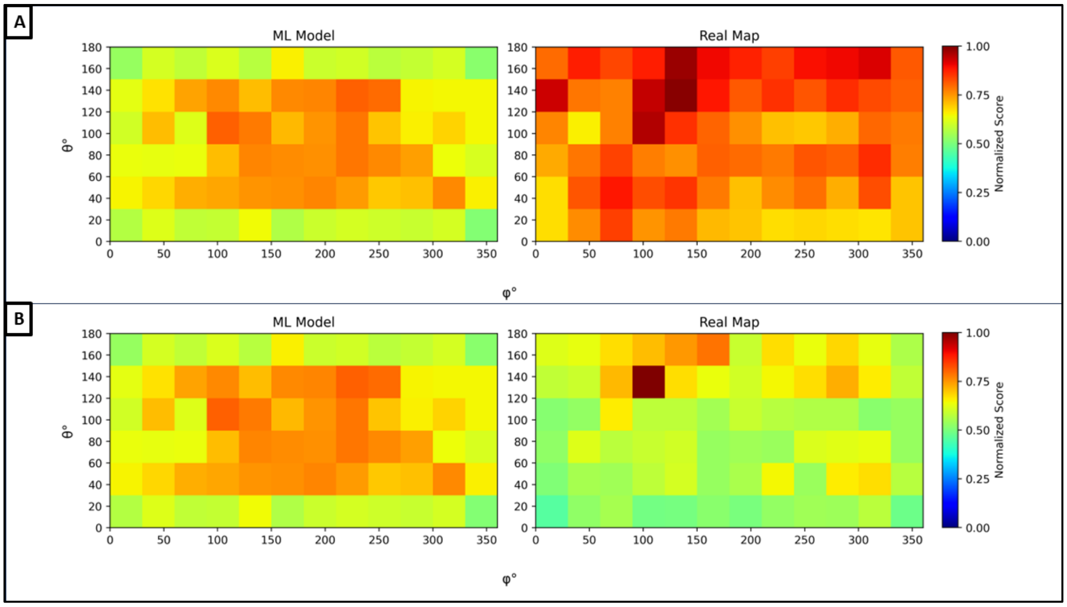


Figure 26. PPI maps (left) generated by the large 3 Sigmoidal model with a learning rate of 0.01 using a MSE divergence loss function compared to (right) PPI maps generated by PatchDock for (A) 3PMG chain A and (B) 1JG8 chain D.

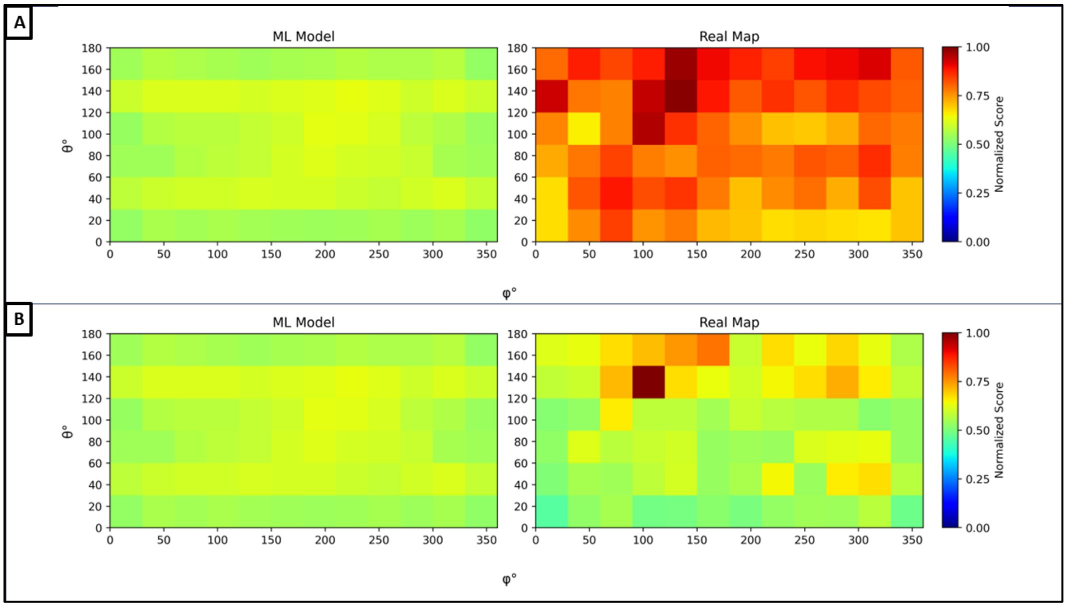


Figure 27. PPI maps (left) generated by the large 3 Sigmoidal model with a learning rate of 0.01 using a JS divergence loss function compared to (right) PPI maps generated by PatchDock for (A) 3PMG chain A and (B) 1JG8 chain D.

4. Conclusions

This work proposes a representation of scaled PPI docking scores based on protein orientation, rooted in the work of Lopez et al.[1], which we refer to as PPI maps. To do so, each protein was matched to a frame of reference that aligns their inertial axes and α-carbon COM to the reference axes and origin of a cartesian coordinate system, pictured in Figure 4. The α-carbon COM of a second protein, docked to the first using PatchDock, was used to identify two variables, θ and ϕ, that indicate where the second protein docks on the first protein. Transformation of the coordinate system so that the second protein's COM and reference frames are aligned allowed for the orientation of both proteins in the docked system to be found.

The coordinate system generated for each protein was used to create specialized heat maps for the study of PPI interactions. By binning the highest score in a range of θ and ϕ, information about the landscape of possible PPIs for a specified protein pair could be found. This landscape can be used as a method of analysis for protein surface features, as well as relationship between these features and PPIs. This analysis is complicated however, due to the orientation space of a heterodimer interaction. Each heterodimer has a unique coordinate frame, leading to two PPI maps being required, one for each protein, to completely analyze the PPI topology. This requirement is eased in the case of homodimers. An ideal homodimer PPI is symmetric, and as such, the PPI maps for both monomers should be the same. Due to simplifications and limitations in molecular docking, the generated PPI maps tend to have differences. By taking the arithmetic average of these maps, some of this error may be reduced, leading to a better PPI map.

Because of the computational infeasibility of completely exploring the PPI conformation space, docking programs must utilize some method to imperfectly sample the conformation space. This limits the resolution of accurate data presented in PPI maps. This work utilized the symmetric docking of homodimers to account for this by constructing a PPI map consisting of the average scaled score of the homodimer's constituents, then analyzing the convergence of several optimization criteria at various resolutions, as seen in section 3.3. The criteria used were Pearson's correlation coefficient, KL divergence, and JS Divergence.

It was found that the three optimization values converged between $30° - 45°$ (PCC), $45° - 60°$ (KL divergence), and at $90°$ (JS divergence). This work proposes an optimal bin size of $30°$, as it allows for linear convergence of the maps, ensuring that the receptor and ligand docking maps agree, while also keeping a small amount of topological noise in the average maps. This allows for some of the randomness in the docking program, due to the conformation space not being fully explored, to remain, as a measure for further exploration of the PPI space.

The aligned protein PDBs and the optimized PPI maps were used to create a dataset for use in training a ML model for prediction of PPI maps. The dataset was separated based on the biophysical properties of the protein to avoid overfitting of the model onto a subset of proteins with similar biophysical properties. The properties used for splitting in this work were overall charge, the radius of gyration, the arithmetic surface roughness, and the $SASA_H$ of each protein. The distributions of each set after splitting are displayed in Figure 11.

The split datasets were tested using several small NN models. Each model contained three hidden layers, with the activation functions being either a sigmoidal function or a ReLu function, as well as two loss functions, a MSE loss and a JS divergence loss function It was found that they performed better when a larger number of sigmoidal functions were used. However, analysis found that all models generated in this work overgeneralized to a single generalized output, instead of fitting to the generated protein.

Following these results, this work proposes the next steps for analyzing and improving the use of PPI maps, as well as the use of machine learning for aiding PPI docking. New steps in the use of PPI maps should focus on studying their applicability to heterodimers. While the use of homodimers in this study was advantageous for studying useful resolutions for PPI maps, the increased complexity of heterodimers is vastly more complex, and approaches will require the use of Euler angles, quaternions, or the like, to be used for matching the reference frame of each protein to each other.

Future works focused on the use of machine learning methods for the generation of PPI maps, barring the generation of new data for ML purposes, may be split into two parts. First is the use of increasingly complex and larger model architectures, beyond the small models used in this work. Second is the use of graph-based ML methods including chemical data, as a replacement for the geometric approach used in this work.

References

(1) Lopez, H.; Brandt, E. G.; Mirzoev, A.; Zhurkin, D.; Lyubartsev, A.; Lobaskin, V. Multiscale Modelling of Bionano Interface. In *Modelling the Toxicity of Nanoparticles*, Tran, L., Bañares, M. A., Rallo, R. Eds.; Springer International Publishing, 2017; pp 173-206.

(2) Weber, C. H.; Park, Y. S.; Sanker, S.; Kent, C.; Ludwig, M. L. A prototypical cytidylyltransferase: CTP:glycerol-3-phosphate cytidylyltransferase from Bacillus subtilis. *Structure* **1999**, *7* (9), 1113-1124. DOI: 10.1016/s0969-2126(99)80178-6.

(3) Junge, W.; Lill, H.; Engelbrecht, S. ATP synthase: an electrochemical ransducer with rotatory mechanics. *Trends in Biochemical Sciences* **1997**, *22* (11), 420-423. DOI: https://doi.org/10.1016/S0968-0004(97)01129-8.

(4) Duhovny, D.; Nussinov, R.; Wolfson, H. J. Efficient Unbound Docking of Rigid Molecules. In *Algorithms in Bioinformatics*, Berlin, Heidelberg, 2002//, 2002; Guigó, R., Gusfield, D., Eds.; Springer Berlin Heidelberg: pp 185-200.

(5) Schneidman-Duhovny, D.; Inbar, Y.; Nussinov, R.; Wolfson, H. J. PatchDock and SymmDock: servers for rigid and symmetric docking. *Nucleic Acids Research* **2005**, *33* (suppl_2), W363-W367. DOI: 10.1093/nar/gki481 (acccessed 3/27/2024).

(6) Zheng, W.; Zhang, C.; Wuyun, Q.; Pearce, R.; Li, Y.; Zhang, Y. LOMETS2: improved meta-threading server for fold-recognition and structure-based function annotation for distant-homology proteins. *Nucleic Acids Res* **2019**, *47* (W1), W429-w436. DOI: 10.1093/nar/gkz384 From NLM.

(7) Zheng, W.; Wuyun, Q.; Zhou, X.; Li, Y.; Freddolino, P. L.; Zhang, Y. LOMETS3: integrating deep learning and profile alignment for advanced protein template recognition and function annotation. *Nucleic Acids Research* **2022**, *50* (W1), W454-W464. DOI: 10.1093/nar/gkac248 (acccessed 5/28/2024).

(8) Schrödinger, L. L. C. The PyMOL Molecular Graphics System, Version 1.8.

(9) Michaud-Agrawal, N.; Denning, E. J.; Woolf, T. B.; Beckstein, O. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *Journal of Computational Chemistry* **2011**, *32* (10), 2319-2327. DOI: https://doi.org/10.1002/jcc.21787 (acccessed 2024/03/26).

(10) R. J. Gowers, M. L., J. Barnoud, T. J. E. Reddy, M. N. Melo, S. L. Seyler, D. L. Dotson, J. Domanski, S. Buchoux, I. M. Kenney, and O. Beckstein. MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations. Benthall, Ed.; 2016///; pp 98 - 105. DOI: 10.25080/Majora-629e541a-00e.

(11) Pierce, B. G.; Hourai, Y.; Weng, Z. Accelerating Protein Docking in ZDOCK Using an Advanced 3D Convolution Library. *PLoS ONE* **2011**, *6* (9), e24657. DOI: 10.1371/journal.pone.0024657.

(12) Honorato, R. V.; Trellet, M. E.; Jiménez-García, B.; Schaarschmidt, J. J.; Giulini, M.; Reys, V.; Koukos, P. I.; Rodrigues, J. P. G. L. M.; Karaca, E.; van Zundert, G. C. P.; et al. The HADDOCK2.4 web server for integrative modeling of biomolecular complexes. *Nature Protocols* **2024**. DOI: 10.1038/s41596-024-01011-0.

(13) Mondal, A.; Chang, L.; Perez, A. Modelling peptide–protein complexes: docking, simulations and machine learning. *QRB Discovery* **2022**, *3*, e17. DOI: 10.1017/qrd.2022.14 From Cambridge University Press Cambridge Core.

(14) Mohanty, M.; Mohanty, P. S. Molecular docking in organic, inorganic, and hybrid systems: a tutorial review. *Monatshefte für Chemie - Chemical Monthly* **2023**, *154* (7), 683-707. DOI: 10.1007/s00706-023-03076-1.

(15) Lane, D. Values of the Pearson Correlation. In *Introductory Statistics*, Rice University, 2022.

(16) Thomas M. Cover, J. A. T. Entropy, Relative Entropy, and Mutual Information. In *Elements of Information Theory*, 2005; pp 13-55.

(17) Nielsen, F. On a Generalization of the Jensen–Shannon Divergence and the Jensen–Shannon Centroid. In *Entropy*, 2020; Vol. 22.

(18) Hu, Y.; Stumpfe, D.; Bajorath, J. Computational Exploration of Molecular Scaffolds in Medicinal Chemistry. *Journal of Medicinal Chemistry* **2016**, *59* (9), 4062-4076. DOI: 10.1021/acs.jmedchem.5b01746.

(19) Petrisor, O. A. University College Dublin Research Master's Thesis. University College Dublin, 2024.

(20) *pandas-dev/pandas: Pandas*; Zenodo: Apr. 10, 2024, 2024. (accessed Aug. 4th, 2024).

(21) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596* (7873), 583-589. DOI: 10.1038/s41586-021-03819-2.

(22) da Hora, G. C. A.; Oh, M.; Nguyen, J. D. M.; Swanson, J. M. J. One Descriptor to Fold Them All: Harnessing Intuition and Machine Learning to Identify Transferable Lasso Peptide Reaction Coordinates. *The Journal of Physical Chemistry B* **2024**, *128* (17), 4063-4075. DOI: 10.1021/acs.jpcb.3c08492.

(23) Ash, J.; Fourches, D. Characterizing the Chemical Space of ERK2 Kinase Inhibitors Using Descriptors Computed from Molecular Dynamics Trajectories. *Journal of Chemical Information and Modeling* **2017**, *57* (6), 1286-1299. DOI: 10.1021/acs.jcim.7b00048.

(24) Goh, G. B.; Hodas, N. O.; Vishnu, A. Deep learning for computational chemistry. *Journal of Computational Chemistry* **2017**, *38* (16), 1291-1307. DOI: https://doi.org/10.1002/jcc.24764 (acccessed 2024/05/22).

(25) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv [cs.LG]* **2019**.

(26) Grant, M. L. Nonuniform Charge Effects in Protein−Protein Interactions. *The Journal of Physical Chemistry B* **2001**, *105* (14), 2858-2863. DOI: 10.1021/jp0039823.

(27) Jifeng, Z. Protein-Protein Interactions in Salt Solutions. In *Protein-Protein Interactions*, Weibo, C., Hao, H. Eds.; IntechOpen, 2012; p Ch. 18.

(28) Jernigan, R. L.; Khade, P.; Kumar, A.; Kloczkowski, A. Using Surface Hydrophobicity Together with Empirical Potentials to Identify Protein–Protein Binding Sites: Application to the Interactions of E-cadherins. In *Computer Simulations of Aggregation of Proteins and Peptides*, Li, M. S., Kloczkowski, A., Cieplak, M., Kouza, M. Eds.; Springer US, 2022; pp 41-50.

(29) Todoroff, N.; Kunze, J.; Schreuder, H.; Hessler, G.; Baringhaus, K.-H.; Schneider, G. Fractal Dimensions of Macromolecular Structures. *Molecular Informatics* **2014**, *33* (9), 588-596. DOI: https://doi.org/10.1002/minf.201400090 (acccessed 2024/08/03).