ALGEBRAIC TOPOLOGY AND MACHINE LEARNING FOR BIOMOLECULAR MODELING

By

Zixuan Cang

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Applied Mathematics - Doctor of Philosophy

2018

ABSTRACT

ALGEBRAIC TOPOLOGY AND MACHINE LEARNING FOR BIOMOLECULAR MODELING

By

Zixuan Cang

Data is expanding in an unprecedented speed in both quantity and size. Topological data analysis provides excellent tools for analyzing high dimensional and highly complex data. Inspired by the topological data analysis's ability of robust and multiscale characterization of data and motivated by the demand of practical predictive tools in computational biology and biomedical researches, this dissertation extends the capability of persistent homology toward quantitative and predictive data analysis tools with an emphasis in biomolecular systems.

Although persistent homology is almost parameter free, careful treatment is still needed toward practically useful prediction models for realistic systems. This dissertation carefully assesses the representability of persistent homology for biomolecular systems and introduces a collection of characterization tools for both macromolecules and small molecules focusing on intra- and inter-molecular interactions, chemical complexities, electrostatics, and geometry. The representations are then coupled with deep learning and machine learning methods for several problems in drug design and biophysical research.

In real-world applications, data often come with heterogeneous dimensions and components. For example, in addition to location, atoms of biomolecules can also be labeled with chemical types, partial charges, and atomic radii. While persistent homology is powerful in analyzing geometry of data, it lacks the ability of handling the non-geometric information. Based on cohomology, we introduce a method that attaches the non-geometric information

to the topological invariants in persistent homology analysis. This method is not only useful to handle biomolecules but also can be applied to general situations where the data carries both geometric and non-geometric information.

In addition to describing biomolecular systems as a static frame, we are often interested in the dynamics of the systems. An efficient way is to assign an oscillator to each atom and study the coupled dynamical system induced by atomic interactions. To this end, we propose a persistent homology based method for the analysis of the resulting trajectories from the coupled dynamical system.

The methods developed in this dissertation have been applied to several problems, namely, prediction of protein stability change upon mutations, protein-ligand binding affinity prediction, virtual screening, and protein flexibility analysis. The tools have shown top performance in both commonly used validation benchmarks and community-wide blind prediction challenges in drug design.

To my parents, for their love.

ACKNOWLEDGMENTS

I would particularly like to thank my advisor Professor Guowei Wei for his enlightening guidance and tremendous support which have shaped the result in this thesis. His passion in research, visionary insights, and bold ideas have greatly encouraged me and influenced my path.

I want to thank Professor Elizabeth Munch and Professor Yiying Tong for the mathematical instructions, the fruitful discussions, and their encouragement.

The interdisciplinary research experience I obtained in the collaboration with Professor Ke Dong, Professor Heedeok Hong, and Professor Jian Hu is invaluable to the application part of this thesis. I would like to thank Professor Gunnar Carlsson for useful discussions and support. In addition, I want to express my gratitude to Professor Peter Bates and Professor Changyi Wang for guidance, to Dr. Lin Mu for offering me the opportunity of collaborating at ORNL, and to Professor Di Liu for serving in my committee. I thank my undergraduate advisors Professor Zhong Tan and Professor Jianyong Wang at Xiamen University for motivating and supporting me. I would also like to thank my friends and colleagues who have supported me and enriched my life during my study here.

Finally yet importantly, I would like to thank my parents Wei Shi and Ping Cang for their unchanging love and trust and my girlfriend Yiqing Yang for her support and understanding throughout my research career.

TABLE OF CONTENTS

LIST OF TABLES . . . . . **x**

LIST OF FIGURES . . . . . **xii**

Chapter 1 Introduction **1**
1.1 Applied algebraic topology 1
1.2 Machine learning and deep learning 5
1.3 Biomolecular modeling 8
1.4 Motivation 10
1.5 Outline 11

Chapter 2 Background **13**
2.1 Applied algebraic topology 13
2.1.1 Simplicial homology 13
2.1.2 Filtration and persistence 15
2.1.3 Barcode space metrics 19
2.2 Coupled dynamical systems 20
2.2.1 Oscillators and coupling 20
2.2.2 Stability and controllability 21
2.3 Machine learning 21
2.3.1 K-nearest neighbor algorithm 22
2.3.2 Decision tree 23
2.3.3 Ensemble of trees 25
2.3.4 Deep learning 26
2.4 Biomolecular modeling 29
2.4.1 Proteins and small molecules 29
2.4.2 Physical modeling 29
2.4.3 Sequence tools 32

Chapter 3 TopologyNet: Deep convolutional neural networks based on topology for biomolecular property predictions. **33**
3.1 Introduction 33
3.2 Methods 38
3.2.1 Persistent homology 38
3.2.2 Topological representation of biomolecules 39
3.2.3 Neuron for persistence barcode 49
3.2.4 Multichannel topological convolutional neural network 50
3.3 Results 56
3.3.1 Deep learning prediction of protein-ligand binding affinities 56
3.3.2 Deep learning prediction of protein folding free energy changes upon mutation 59

3.3.3 Multi-task deep learning prediction of membrane protein mutation impacts . . . 61
3.4 Discussion and conclusion . . . 63

Chapter 4 Persistent cohomology for data with heterogeneous dimensions 68
4.1 Introduction . . . 68
4.2 Methods . . . 72
4.2.1 Cohomology . . . 72
4.2.2 Smoothed cocycle . . . 73
4.2.3 Enriched persistent barcode . . . 75
4.2.4 Preprocessing of the input function . . . 77
4.2.5 Modified Wasserstein distance . . . 79
4.3 Examples and results . . . 81
4.3.1 A minimalist example . . . 81
4.3.2 Example datasets . . . 82
4.3.3 Wasserstein distance based similarity . . . 86
4.3.4 Analysis of molecules . . . 87
4.3.5 An application to protein-ligand binding . . . 89
4.4 Discussion and conclusion . . . 95

Chapter 5 Evolutionary homology for coupled dynamical systems . . . 97
5.1 Introduction . . . 97
5.2 Methods . . . 99
5.2.1 Coupled dynamical systems . . . 100
5.2.1.1 Oscillators and coupling . . . 100
5.2.1.2 Stability and controllability . . . 101
5.2.1.3 Topological learning . . . 104
5.2.2 Evolutionary homology (EH) and the EH barcodes . . . 106
5.2.2.1 Filtration function defined for coupled dynamical systems . 107
5.2.2.2 Definition of evolutionary homology . . . 110
5.2.3 Protein residue flexibility analysis . . . 110
5.3 Results . . . 114
5.3.1 Disordered and flexible protein regions . . . 114
5.3.2 Protein B-factor prediction . . . 115
5.4 Conclusion . . . 119

Chapter 6 Topological characterization of static macrobiomolecules and small molecules . . . 121
6.1 Introduction . . . 121
6.2 Biological considerations . . . 122
6.3 Methods . . . 125
6.3.1 Element specific persistent homology . . . 125
6.3.2 Construction of distance matrix . . . 126
6.3.2.1 Multi-level persistent homology. . . . 126
6.3.2.2 Interactive persistent homology. . . . 128

6.3.2.3 Correlation function based persistent homology. 129
6.3.2.4 Electrostatic persistence 130
6.3.3 Feature generation from topological invariants 132
6.3.4 Machine learning algorithms 136
6.4 Results 141
6.4.1 Ligand based protein-ligand binding affinity prediction 143
6.4.2 Complex based protein-ligand binding affinity prediction 145
6.4.3 Structure-based virtual screening 148
6.5 Discussion 156
6.5.1 Ligand based protein-ligand binding affinity prediction 156
6.5.2 Complex based protein-ligand binding affinity prediction 163
6.5.2.1 Robustness of GBT algorithm against redundant element combination features and potential overfitting. 163
6.5.3 Structure-based virtual screening 171
6.6 Conclusion 173

Chapter 7 Dissertation contribution 176

BIBLIOGRAPHY 180

LIST OF TABLES

Table 3.1: Topological representations of protein-ligand complexes. 46

Table 3.2: Topological representations for protein mutation problem. 47

Table 3.3: Performance comparisons of TNet-BP and other methods 58

Table 3.4: Performance comparisons of TNet-MP and other methods. 61

Table 3.5: Performance comparisons of TNet-MMP and other methods. 63

Table 4.1: Candidate values for hyper-parameters of the gradient boosting trees model. 95

Table 4.2: The predictor performance is evaluated by training on PDBBind refined set excluding the core set and testing on the core set of a certain year's version. The median Pearson's correlation coefficient (root mean squared error) among 10 repeated experiments is reported. 95

Table 5.1: The averaged Pearson correlation coefficients (R_P) between the computed values (blind prediction for the topological features and regression for the rest of the models) and the experimental B-factors for a set of 364 proteins [146] (Left: Prediction R_Ps based on EH barcodes. Right: A comparison of the R_Ps of predictions from different methods.). Here, EH is the linear regression using $EH^{\infty,0}$, $EH^{\infty,1}$, $EH^{1,0}$, $EH^{1,1}$, $EH^{2,0}$, and $EH^{2,1}$ within each protein. For a few large and multi-chain proteins (i.e., 1F8R, 1H6V, 1KMM, 2D5W, 3HHP, 1QKI, and 2Q52), to reduce the computational time and as a good approximation, we compute their EH barcodes on separated (protein) chains. We see from the table at right that the proposed EH barcode method outperforms other methods in this application. 118

Table 6.1: Pearson correlation coefficients (RMSE in kcal/mol) of ligand based topological model on the S1322 dataset. 144

Table 6.2: Description of the PDBBind datasets. 146

Table 6.3: Pearson correlation coefficients (RMSE in kcal/mol) of different protein-ligand complex based approaches on PDBBind datasets. 146

Table 6.4: Parameters used in machine learning. 151

Table 6.5: Performance on each protein in DUD dataset. 154

Table 6.6: AUC comparison of different methods on DUD dataset. 155

Table 6.7: Experiments for ligand-based protein-ligand binding affinity prediction of 7 protein clusters and 1322 protein-ligand complexes. 161

Table 6.8: Performance of different approaches on the S1322 dataset. 162

Table 6.9: Experiments for protein-ligand-complex-based protein-ligand binding affinity prediction for the PDBBind datasets. 164

Table 6.10: Performance of different protein-ligand complex based approaches on the PDBBind datasets. 167

Table 6.11: The AUC for autodock vina, TopVS-ML with only compound features, TopVS-ML with only protein-compound complex features, and TopVS-ML with all features. The targets with high quality results by Autodock Vina are reported (AUC > 0.8) 172

Table 6.12: The AUC for autodock vina, TopVS-ML with only compound features, TopVS-ML with only protein-compound complex features, and TopVS-ML with all features. The targets with low quality results by Autodock Vina are reported (AUC < 0.5) 173

LIST OF FIGURES

Figure 2.1: Persistence barcodes of alpha complex filtration (bottom left) and Vietoris-Rips complex filtration (bottom right) for the point cloud (top). The top and bottom panels of barcodes are H_1 and H_0 barcodes. 18

Figure 2.2: Example regression decision tree. 24

Figure 2.3: a. Dense layer. b. Convolution layer. 27

Figure 2.4: A practical neural network architecture for multitask learning. 28

Figure 3.1: An illustration of barcode changes from wild type to mutant proteins.[26] 41

Figure 3.2: Energy cycle of protein-ligand binding free energy modeling.[26] 45

Figure 3.3: Mutation induced protein folding free energy changes.[26] 48

Figure 3.4: An illustration of the 1D convolutional neural network.[26] 51

Figure 3.5: The deep learning architecture for the application to globular proteins.[26] 53

Figure 3.6: The multi-task deep learning architecture for membrane proteins.[26] . . 54

Figure 3.7: A comparison of behaviors of the GBT based method and the neural network based method.[26] 66

Figure 4.1: A simple example loop.[25] 82

Figure 4.2: a: A point cloud sampled from two adjacent annulus. b: The corresponding H_1 barcode using alpha complex filtration.[25] 83

Figure 4.3: Example of smoothed H^1 cocycle.[25] 83

Figure 4.4: a and b: Two datasets with similar geometry but different information given on the nodes. c and d: The differences are revealed in the enriched H_1 barcodes.[25] 84

Figure 4.5: Persistent cohomology enriched barcode example of data points sampled from porous cuboid.[25] 85

Figure 4.6: D3 dataset sampled from an annulus with randomly assigned values on the points and corresponding H_1 enriched barcode. 86

Figure 4.7: Wasserstein characteristics curve. 87

Figure 4.8: a: The cucurbit[8]uril molecule viewed from two different angles. The hydrogen, carbon, nitrogen, and oxygen atoms are colored in white, grey, blue, and red. b, c, and d: The H_1 enriched barcodes obtained by assigning 1 to nodes of the selected atom types (carbon, nitrogen, and oxygen) and 0 elsewhere.[25] 88

Figure 4.9: **a**: A structure of the $B_{24}N_{24}$ cage. The nitrogen and boron atoms are colored in blue and grey. **b**: The enriched barcodes obtained by assigning 1 to Boron atoms and 0 elsewhere. H_1 and H_2 barcodes are plotted in bottom and top panels.[25] 89

Figure 4.10: Enriched barcodes focusing on atomic partial charges. [25] 91

Figure 5.1: a: Chaotic trajectory of one oscillator without coupling. b: The 70 synchronized oscillators associated with the carbon C_α atoms of protein PDB:1E68 are plotted together.[22] 103

Figure 5.2: The filtration of the simplicial complex associated to three 1-dimensional trajectories.[22] 109

Figure 5.3: An example of the construction of the evolutionary homology barcode.[22] 111

Figure 5.4: The result of perturbing residue 31 in protein (PDB:1ABA).[22] 112

Figure 5.5: Left: partially disordered protein, model 1 of PDB:2RVQ. Right: well folded protien, PDB:1UBQ.[22] 114

Figure 5.6: (a) Models 1-3 of PDB:2ME9 with the disordered region colored in blue, red, and yellow for the three models. (b) Similar plot as (a) for PDB:2MT6. (c) Topological features for PDB:2ME9 whose large disordered region is from residue 28 to residue 85. (d) Topological features for PDB:2MT6 whose large disordered region is from residue 118 to residue 151. [22] . . 116

Figure 5.7: Barcode plots for two residues. (a) Residue 6 of PDB:2NUH with a B-factor of 12.13 $Å^2$. (b) Residue 49 of PDB:2NUH with a B-factor of 33.4 $Å^2$.[22] 117

Figure 5.8: B-factors and the computed topological features. EH shows the linear regression with $EH^{\infty,0}$, $EH^{\infty,1}$, $EH^{1,0}$, $EH^{1,1}$, $EH^{2,0}$, and $EH^{2,1}$ within each protein. (a) PDB:3PSM with 94 residues. (b) PDB:3SZH with 697 residues.[22] 119

Figure 6.1: Multi-level persistent homology on simple small molecules.[20] 128

Figure 6.2: The network architecture of TopBP-DL.[20] 139

Figure 6.3: The network architecture of TopVS-DL.[20] 140

Figure 6.4: An illustration of the topology based machine learning algorithms used in scoring and virtual screening.[20] 142

Figure 6.5: Statistics of ligands in 7 protein clusters in S1322 dataset.[20] 157

Figure 6.6: An illustration of similarities between ligands measured by their barcode space Wasserstein distances.[20] 158

Figure 6.7: Plot of performance against number of element combinations used.[20] . 160

Figure 6.8: Feature robustness tests on PDBBind datasets.[20] 166

Figure 6.9: Assessment of performance of the model on samples with elements that are rare in the data sets.[20] 170

Chapter 1

Introduction

1.1 Applied algebraic topology

Topology delivers an abstraction of a space by studying the properties that persist as the space deforms continuously. A concise description of a space can be obtained by using topology and the geometric information of different levels of details can be kept by using methods in algebraic topology. Because of the brevity and the ability of information preservation, algebraic topology methods have been applied to data analysis leading to the advancement in an emerging field called topological data analysis. Topological data analysis is especially powerful at handling high-dimensional and highly complex data sets and has been applied to many fields such as image processing, network analysis, and genomics study. The emphasis on the application side of this thesis is given to molecular biology where quantitative and predictive models are developed.

Homology can distinguish topological spaces by assigning algebraic structures to the spaces to characterize the holes of various dimensions. Intuitively speaking, the 0-dimensional homology characterizes connected components, the 1-dimensional homology counts circular structures, the 2-dimensional homology addresses voids or cavities, and the higher-dimensional homology concerns the higher-dimensional holes. Given a data set with a chosen parameter reflecting the geometric scale, one can compute the homology by building a com-

plex upon the data and representing the algebraic structures as vector spaces. There are also different ways for data representation. For example, a simplicial complex consisted of points, edges, triangles, and the higher-dimensional counterparts can be built upon a point cloud and a cubical complex can naturally represent volumetric data. Efficient algorithms have been proposed together with several implementations enabling the analysis of large datasets. Computational homology has been applied in both mathematical applications and applications in other fields. For example, the Conley index for real-world systems can be computed using computational homology [98, 208]. It has also been applied to topological characterizations of spatial-temporal chaos [76].

The ability to automatically analyze data at multiple scales is important for a practical data analysis method. Fewer assumptions of the given data need to be made if such property is met which indicates higher robustness. Persistent homology adds another dimension to the conventional homology by introducing a filtration of the topological space to achieve a multiscale characterization of data. Instead of computing homology of a fixed topological space, persistent homology scans along a filtration of a space where the building blocks are added sequentially to the space ordered by their associated filtration parameter values. Then, the computation is done for not only homology of each frame of filtration but also how the homology generators appear, disappear, and persist along the course of filtration. A method named size function was introduced for applications in computer vision which is a 0th dimensional version of persistent homology [72, 163]. Persistent homology theory and practical algorithms were formulated and developped by Edelsbrunner et al. [62]. A more general theory was later introduced by Zomorodian and Carlsson [216]. The outputs of persistent homology computation are collections of homology generators of different dimensions paired with its "birth" and "death" values along the course of filtration and are

usually visualized via barcode plots [33, 79] where horizontal line-segments are stacked each corresponds to a homology generator or persistence diagrams [43, 36] where each generator is plotted as a point in the 2-dimensional plane. A quantitative description of two persistent homology computation results can be realized by computing their bottle-neck distance or more generally, their pth Wasserstein distance [44, 29]. As a topological method, persistent homology massively reduces the original dimension of data to one retaining crucial geometric information through the filtration. These features make persistent homology a suitable tool for data analysis especially for complex and high-dimensional data. Persistent homology also has a potential to be paired with predictive models such as deep learning and manifold learning.

Since the introduction of persistent homology, there have been tremendous advancements in algorithms, implementations, and theories. The theory of multidimensional persistent homology has been formulated to address the situation where multiple parameters are involved in building the increasing spaces in filtration [32]. A practical implementation for 2-dimensional case was developed enabling interactive visualization [113]. Zigzag persistent homology together with practical algorithm were introduced which allow traveling in both directions along filtration facilitating the analysis of real-valued functions on topological spaces by associating homology groups to levelsets [30]. Introduced to study persistence diagram stability, vinyards can be used to analyze time-varying data [45]. Many systems can be recorded as graphs. Clique complex can be efficiently constructed for undirected graphs [215] and path complex was introduced for directed graphs [84]. The theory of path complex was extended with persistence theory formulating persistent path homology for the analysis of directed graphs [41]. Efforts have also been made on the dual side of persistent homology to make use of the richer information carried by cohomology. A more detailed description of

1-dimensional homology generators was derived by assigning circular coordinates to the input space using persistent cohomology [54]. Cohomology was also used for the coordination in higher dimensional cases [155].

There is a collection of software for persistent homology with a wide range of utilities. Dionysus [129] is able to compute zigzag persistence, persistent cohomology, vineyards, alpha complex filtration, circular coordinates, and bottleneck and Wasserstein distances. Ripser [8] focuses on Rips complex filtration and is extremely fast. Perseus [136] speeds up computation by using discrete Morse theory [127] and provides utilities for cubical complex filtration. DIPHA provides efficient algorithms enabling distributed computing [9] and PHAT provides fast matrix reduction implementations [11]. JavaPlex is easy to use through MATLAB and can conveniently illustrate the concepts [2]. It also includes some approximation constructions for faster computation. Gudhi [122] is another comprehensive library with Python interface. There is also efficient implementations of both exact and approximate algorithms for computing the bottleneck and Wasserstein distances [101]. Both computer science techniques and mathematical properties are used to accelerate computations and one of the most important theoretical foundations is the duality between persistent homology and persistent cohomology [53].

Another major method in topological data analysis is mapper which builds a graph to represent the topology of a dataset [171]. Given a point cloud dataset, each data point is first assigned a value which can be its eccentricity or one of many other choices and this assignment is called a filter function. Then, a cover of the range of the filter function is decided based on user defined parameters such as cover length and overlap percentage. Finally, data points in the same cover are grouped and are represented by a node in the graph while an edge in the graph means there is an overlap of the intervals associated to the

two nodes. Compared to persistent homology, mapper reduces the data dimension but still keeps a relatively explicit representation of the topology of data making it a good method for visualization especially for high dimensional datasets. More quantitative descriptions can be derived by running persistent homology upon mapper graphs. A library that can be called in Python with GUI is available [132]. Mapper has found its applications in biological and biomedical sciences [142, 143, 206].

Computational homology and persistent homology have been applied to various fields, including image and signal analysis [31, 150, 172, 14, 73, 157], chaotic dynamics characterization [126, 98], sensor networks [80], complex networks [112, 91], shape recognition [57, 66], and computational biology [99, 75, 48, 156].

1.2 Machine learning and deep learning

Machine learning models are able to automatically extract information from data and subsequently make predictions or inferences. There are mainly supervised, semi-supervised, and unsupervised learning. In supervised learning, collections of data entries with descriptions and labels are given to the model and once the model has learned from the data it makes predictions on labels of new data given only descriptions. A simple example of supervised learning is the linear regression model. There are also situations where large dataset is given but only a small portion of data has known labels. Instead of learning on the small amount of labeled data using supervised learning techniques, semi-supervised learning models also utilize unlabeled data which usually help determine the underlying distribution of the data. One example is manifold learning where unlabeled data are used to approximate the underlying manifold and the similarities between data entries are measured by their distance

on the manifold. Unsupervised learning derives useful information from unlabeled data by inferring the underlying structure of data. In this thesis, we mainly focus on supervised learning with an emphasis on molecular biology.

There are many competitive machine learning methods and we just list a few here. Support vector machine [46] builds a hyperplane to separate samples by solving an optimization problem. It can also be extended to nonlinear cases by using a nonlinear kernel which tends to separate data in higher dimensional spaces without the need to explicitly embed the data in the higher dimensional space. Support vector machines can be used for both supervised learning and clustering. Another popular class of machine learning methods is ensemble learning. Ensemble learning relies on the assumption that combining multiple weak learner can improve the prediction performance. A well-known model of this kind is random forest [19] where independent decision trees are built and an average of these tress is used as the final model. Since the decision trees are constructed independently, random forest is usually good at lowering bias. Another way of ensemble is by gradient boosting such as gradient tree boosting where one decision tree is added to the model at a time according to the error at this stage [70, 71]. Due to the nature of gradient boosting, it is good at reducing variance. Inspired by biological neural networks, artificial neural networks [125] were proposed to mimic the arrangement of biological neurons by stacking nonlinear functions forming a complicated composite function with tunable parameters. A common problem for machine learning models especially those with large amount of tunable parameters is the overfitting problem where models can have worse performance on new data while their performance on training data is getting better. Many techniques have been developed to prevent overfitting. For example, each weak learner in the ensemble learning can be trained on randomly selected parts of training data introducing randomness to the training process. Also, a regularization

term containing norms of model parameters can be added to the objective function avoiding making a model too specialized.

Neural network has a potential for complicated and complex situations with its highly flexible structure. After the advancement in dealing with the vanishing gradient problem where impacts of back propagated error on parameters decay too fast along layers, deep learning has flourished. Many different models have been introduced and these models have state-of-the-art performance in many realistic applications. A stacking of regular neural network layers makes a deep neural network and the number of layers can be as many as hundreds. Convolutional neural networks significantly reduce the number of parameters compared to deep neural networks by taking advantage of locality properties of input data and applying the same filter to only local connections [111]. Convolutional neural network is very good at processing image data [107]. Recurrent neural network allows directed connections between neurons in the same layer following the flow of an input sequence and is good at processing sequential data such as gene sequences, natural languages, and time series. In addition to the models that learn functions of inputs, there are also generative models that learn to generate data. When there are related learning tasks, multitask learning or transfer learning take advantage of the shared properties of the tasks to improve the predictive power. Practical multitask learning models can be constructed due to the flexibility and the hierarchical structure of neural networks. In general, one of the major advantages of machine learning and especially deep learning methods is their ability of handling large and diverse datasets. Deep learning methods have established state-of-the-art in many computational biology problems. For example, one of the most accurate sequence-based protein structure prediction tools assessed by blind prediction challenges is based on residual network with tens of convolutional layers [191]. The winner of a blind prediction challenge on toxicity pre-

diction (Tox21) that tens of groups participated was based on deep neural network paired with multitask learning technique [124].

1.3 Biomolecular modeling

Understanding structure-function relationships is a major challenge in the molecular level computational biology. Studying the relationship between structures of biomolecules and their functions not only helps us understand nature but also aids biomedical research such as drug design. For a biomolecular system, there are many important properties that are related to the function of the system. For example, the binding affinity of protein-ligand complexes, the stability changes induced by amino acid mutations in protein, and the flexibility of protein residues. The ability of modeling these basic physical properties further leads to the studying of functions more related to real-world applications such as whether a candidate small molecule potentially binds to a protein target and the impact on drug effect when a specific mutation happens in the target protein.

Physics-based methods build models according to physical laws and are indispensable for molecular modeling which provide predictions and reveal underlying mechanisms. Examples of such kind include quantum mechanics calculation, molecular dynamics simulation, and Monte Carlo sampling. There are also more efficient approximations to the atomic systems by using a continuum for part of the systems. For example, the Poisson-Boltzmann model delivers efficient description of the electrostatics in the solvation processes of molecules by using a continuum solvent. A more complete and affordable parameterization of complex biomolecular systems can be achieved by using force fields usually derived from quantum chemistry computations. Such models are able to model larger realistic systems such as

protein-ligand binding [147, 207]. There are also descriptions of the systems that are potentially related to the target property but rigorous connections to the target property may be hard to derive. In this case, an empirical model can be constructed combining each component with weights determined from experimental data or more expensive but accurate models. A widely used setup for empirical models is to combine molecular mechanics energies with polar part of solvation process modeled by Poisson-Boltzmann model or Generalized Born model and nonpolar part of solvation process reflected by surface areas which are usually called MM/PBSA or MM/GBSA models [189]. MM/PBSA and MM/GBSA models have been applied to the energy modeling of various biomolecular systems including protein-ligand binding [77] and protein-DNA binding [153].

When there are more detailed descriptions of the systems, possibly hundreds or thousands of descriptors of various types, it is hard to effectively put them in a linear regression based empirical model. Machine learning models with more capacity may help with this situation. High capacity usually comes with higher number of model parameters to be determined which requires more data. Generally, the performance of knowledge-based model heavily depends on the quantity and quality of available data. Knowledge-based models using machine learning methods can achieve better performance compared to physics-based and empirical methods given sufficient data. At the same time, the descriptors derived from physical models can be crucial for knowledge-based methods.

In this work, we are interested in several biological applications, structure-based protein-ligand binding affinity prediction where binding affinity is to be predicted given the structure of protein-ligand complexes, prediction of protein stability changes upon mutation where structures of the wild-type proteins are given, virtual screening which aims to determine if a pair of target protein and a candidate small molecule could potentially bind, and protein

flexibility analysis where relative flexibility of protein residues or atoms are to be determined.

Persistent homology has been applied to computational biology [99, 75, 48], in the mathematical modeling and prediction of nano particles, proteins, and other biomolecules [198, 195, 75]. Quantitative topological analysis has been cultivated to predict the curvature energy of fullerene isomers [195, 186], characterize protein folding [198], and quantify immunohistochemical effects [176]. Differential geometry based persistent homology [186] and multiresolutional persistent homology [201] have been proposed to better characterize biomolecular data, detect protein cavities [117], and resolve ill-posed inverse problems in cryo-EM structure determination [200].

1.4 Motivation

The exponential growth of biological data has set the stage for data-driven discovery of structure-function relationships. Indeed, the Protein Data Bank (PDB) has accumulated near 130,000 tertiary structures. The availability of 3D structural data enables knowledge-based approaches to offer complementary and competitive predictions of structure-function relationships. Recent advances in machine learning algorithms have made data driven approaches more competitive and powerful than ever. Arguably, machine learning is one of the most important developments in data analysis. Machine learning has become an indispensable tool in biomolecular data analysis and prediction. Virtually every computational problem in computational biology and biophysics, such as the prediction of solvation free energies, protein-ligand binding affinities, mutation impacts, pKa values, etc, has a class of knowledge based approaches that are either parallel or complementary to physics based approaches.

On the other hand, persistent homology as a data analysis tool can be applied to both data that are sampled from underlying manifolds or naturally discrete data such as molecular structures. Actually, persistent homology has been applied to both qualitative and quantitative analysis of complex molecular structures [99, 75, 48, 198, 195, 176].

Encouraged by persistent homology's ability of information extraction and dimension reduction and the advantage on predictive modeling of machine learning and especially deep learning methods, we aim at developing competitive predictive models using persistent homology and machine learning with focuses on molecular biology. Despite the excellent out-of-box performance of powerful machine learning methods in many applications, proper featurization of biomolecular systems is crucial to the success of a model. To this end, persistent homology can serve as an appropriate featurization tool. Though persistent homology is good at describing geometry and is especially powerful compared to other methods in the case of high dimension, special treatment is still needed to address the chemical and biological complexities when applied to predictive modeling of biomolecules.

1.5 Outline

In Chapter 2, we review the background of persistent homology, coupling of dynamical systems, some relevant machine learning and deep learning methods, and several biological applications we worked on. In Chapter 3, we develop a persistent homology based deep learning model using convolutional neural networks and multitask learning for the prediction of protein-ligand binding affinity and mutation induced protein stability change. A construction of neuron which can directly take persistence barcodes as input is also introduced. We extend the capacity of the topological characterizations in Chapter 4 by embedding physical

properties of the molecules to persistence barcodes using cohomology. This extension is also useful for general situations where data come with heterogeneous dimensions. In Chapter 5, we further consider dynamical properties of molecules using coupled dynamical systems and introduce a construction of persistent homology to analysis the resulting trajectories. In Chapter 6, we introduce several descriptions based on persistent homology for both macrobiomolecules and small molecules and we discuss in detail about the representability of persistent homology for biomolecular systems. The dissertation contribution is summarized in Chapter 7.

Chapter 2

Background

2.1 Applied algebraic topology

2.1.1 Simplicial homology

Topological spaces can be approximated, represented, and discretized by simplicial complexes. An (abstract) simplicial complex is a (finite) collection of sets $K = \{\sigma_i\}_i$ where each σ_i is a subset of a (finite) set K^0 called the vertex set. We require that this collection satisfies the following condition: if $\sigma_i \in K$ and τ is a face of σ_i (that is, if $\tau \subseteq \sigma_j$ commonly denoted $\tau \leq \sigma_i$), then $\tau \in K$. If σ_i has $k+1$ vertices, $\{v_0, v_1, \cdots, v_k\}$ where every pair of vertices is nonequivalent, σ_i is called a k-simplex. The k-skeleton of a simplicial complex K is the subcomplex of K consisting of simplicies of dimension k and below. While the simplices for the abstract simplicial complex we will build will not have an obvious geometric meaning, there is a more geometric viewpoint from which we often reference simplicies. A geometric k-simplex can be regarded as the convex hull of $k+1$ affinely independent points in $\mathbb{R}^d$, and because of this we often call a 0-simplex a point, a 1-simplex an edge, a 2-simplex a triangle, and a 3-simplex a tetrahedron. Without confusion, we will use the same symbols for geometric and abstract simplices.

The homology group for a fixed simplicial complex gives a topological characterization which encodes holes of different dimensions. Homology groups are built using linear transfor-

mations called boundary operators. A k-chain of the simplicial complex K is a finite formal sum of the k-simplices in K, $\alpha = \sum a_i \sigma_i$ with coefficients $a_i \in G$ where G is a chosen group, for example, the widely used one $\mathbb{Z}_2$ or more generally $\mathbb{Z}_p$ with a prime p. The group of all k-chains with addition given by the addition of the coefficients is called the k-th chain group and is denoted by $C_k(K)$ or simply C_k when the choice of complex is obvious. Note that when $\mathbb{Z}_2$ is used, since it is a field, $C_k(K)$ is, in fact, a vector space.

The boundary operator $\partial_k : C_k \to C_{k-1}$ is the linear transformation generated by mapping any k-simplex to the sum of its codim-1 faces; namely,

$$\partial_k(\{v_0, v_1, \cdots, v_k\}) = \sum_{i=0}^{k} (-1)^i \{v_0, \cdots, \widehat{v_i}, \cdots, v_k\},$$

where $\widehat{v_i}$ means that v_i is absent. The kth cycle group, $Z_k(K)$, is the kernel of the boundary operator ∂_k with elements called k-cycles. The kth boundary group, $B_k(K)$, is the image of the boundary operator ∂_{k+1} and its elements are called k-boundaries. Since $\partial_k \circ \partial_{k+1} = 0$, $B_k(K)$ is a subgroup of $Z_k(K)$. Thus we can define the kth homology group, $H_k(K)$, to be the quotient group $Z_k(K)/B_k(K)$. Two k-cycles are called homologous if they differ by a boundary; equivalently if they are in the same equivalence class of $H_k(K)$. Intuitively, if two k-cycles differ from each other by the boundary of a subcomplex, they can roughly be deformed from one to another continuously through the subcomplex. Each equivalence class in $H_k(K)$ can be thought of as corresponding to a k-dimensional "loop" in K going around a $k+1$-dimensional "hole": 1-dimensional classes give information about loops going around 2D voids, 2-dimensional classes give information about enclosures of 3D voids, etc. While the analogy is not as nice, 0-dimensional classes give information about connected components of the space.

2.1.2 Filtration and persistence

We now turn to the case where we have a changing simplicial complex and want to understand something about its structure. Consider a finite simplicial complex K and let f be a real-valued function on the simplices of K which satisfies the following: $f(\tau) \leq f(\sigma)$ for all $\tau \leq \sigma$ simplices in K. We will refer to this function as the filtration function. For any $x \in \mathbb{R}$, the sublevelset of K associated to x is defined as

$$K(x) = \{\sigma \in K \mid f(\sigma) \leq x\}.$$

Note first that because of our assumptions on f, $K(x)$ is always a simplicial complex, and second that $K(x) \subseteq K(y)$ for any $x \leq y$. Further, as x varies, $K(x)$ only changes at the function values defined on the simplices. Since K is assumed to be finite, let $\{x_1 < x_2 < \cdots < x_\ell\}$ be the sorted range of f. The filtration of K with respect to f is the ordered sequence of its subcomplexes,

$$\emptyset \subset K(x_1) \subset K(x_2) \subset \cdots \subset K(x_\ell) = K. \tag{2.1}$$

The filtration of a simplicial complex sets the stage for a thorough topological examination of the space under multiple scales of the filtration parameter which is the output value of the filtration function f.

The definition of homology is valid for a fixed simplicial complex, however we are interested in studying the structure of a filtration like that of Eq. (2.1). Functoriality of homology means that such a sequence of inclusions induces linear transformations on the sequence of

vector spaces

$$H_k(K(x_1)) \to H_k(K(x_2)) \to \cdots \to H_k(K(x_n)). \tag{2.2}$$

Persistent homology not only characterizes each frame in the filtration $\{K(x_i)\}_i$, but also tracks the appearance and disappearance (commonly referred to as births and deaths) of nontrivial homology classes as the filtration progresses. A collection of vector spaces $\{V_i\}$ and linear transformations $f_i : V_i \to V_{i+1}$ is called a persistence module, of which Eq. (2.2) is an example. It is a special case of a much more general theorem of Gabriel [74] that sufficiently nice persistence modules can be decomposed uniquely into a finite collection of interval modules[37, 149]. An interval module $\mathbb{I}_{[b,d)}$ is a persistence module for which $V_i = \mathbb{Z}_2$ if $i \in [b, d)$ and 0 otherwise; and f_i is the identity when possible, and 0 otherwise.

So, given the persistence module of Eq. (2.2), we can decompose it as $\bigoplus_{[b,d)\in B} \mathbb{I}_{[b,d)}$, and thus fully represent the algebraic information by the discrete collection B. These intervals exactly encode when homology classes appear and disappear in the persistence module. The collection of such intervals can be visualized by plotting points in the 2D half plane $\{(x, y) \mid y \geq x\}$ which is known as a persistence diagram; or by stacking the horizontal intervals, which is known as a barcode. In this paper, for no reason other than convenience, we represent our information using barcodes. We call the barcode resulting from a sequence of trivial homology groups the empty barcode and denote it by $\emptyset$. Thus, for every interval $[b, d) \in B$, we call b the birth time and d the death time.

We review two widely used constructions of filtration. Vietoris-Rips (VR) complex is built upon the 1-skeleton induced by pairwise distances among given points. Given a distance function and a threshold distance, a simplex is in the VR complex if the distance between any pair of vertices in the simplex is smaller than or equal to the threshold. More formally,

VR complex on a finite point set X at threshold δ is defined as

$$VR(X, \delta) = \{\sigma | v \in X,\, d(v, v') \leq \delta,\, \forall v, v' \in \sigma\}. \tag{2.3}$$

The distance function can naturally be the Euclidean distance but can also be other user defined distances tailored for specific applications. Since it does not directly rely on the exact geometry, a more abstract usage is possible with distance functions that do not satisfy triangular inequality. Another complex construction is alpha complex which is closely related to geometric modeling. On the finite point set X in the Euclidean space, we can build a Voronoi diagram and let $V(v)$ be the Voronoi cell associated to $v \in X$. Then the alpha complex associated to the parameter ϵ is defined as

$$A(X, \epsilon) = \{\sigma | \cap_{v \in X} (V(v) \cap B(v, \epsilon)) \neq \emptyset\}, \tag{2.4}$$

where $B(v, \epsilon)$ is the ϵ-ball centered at v. It should be noted that alpha complex is a subset of Delaunay triangulation and thus is very efficient in terms of computation. Alpha complex is a faithful representation of geometry. Though usually more computationally costly, Vietoris-Rips is useful when the input point set does not have natural coordinates (though geometric embedding can be done) or some interaction scores are of interest instead of geometric distances. Since $A(X, \epsilon) \subseteq A(X, \epsilon')$ for $\epsilon \leq \epsilon'$ and $VR(X, \delta) \subseteq VR(X, \delta')$ for $\delta \leq \delta'$, these constructions of simplicial complexes can induce proper filtrations and ϵ and δ are called the filtration parameters. An example of persistence barcodes is shown in Figure 2.1.

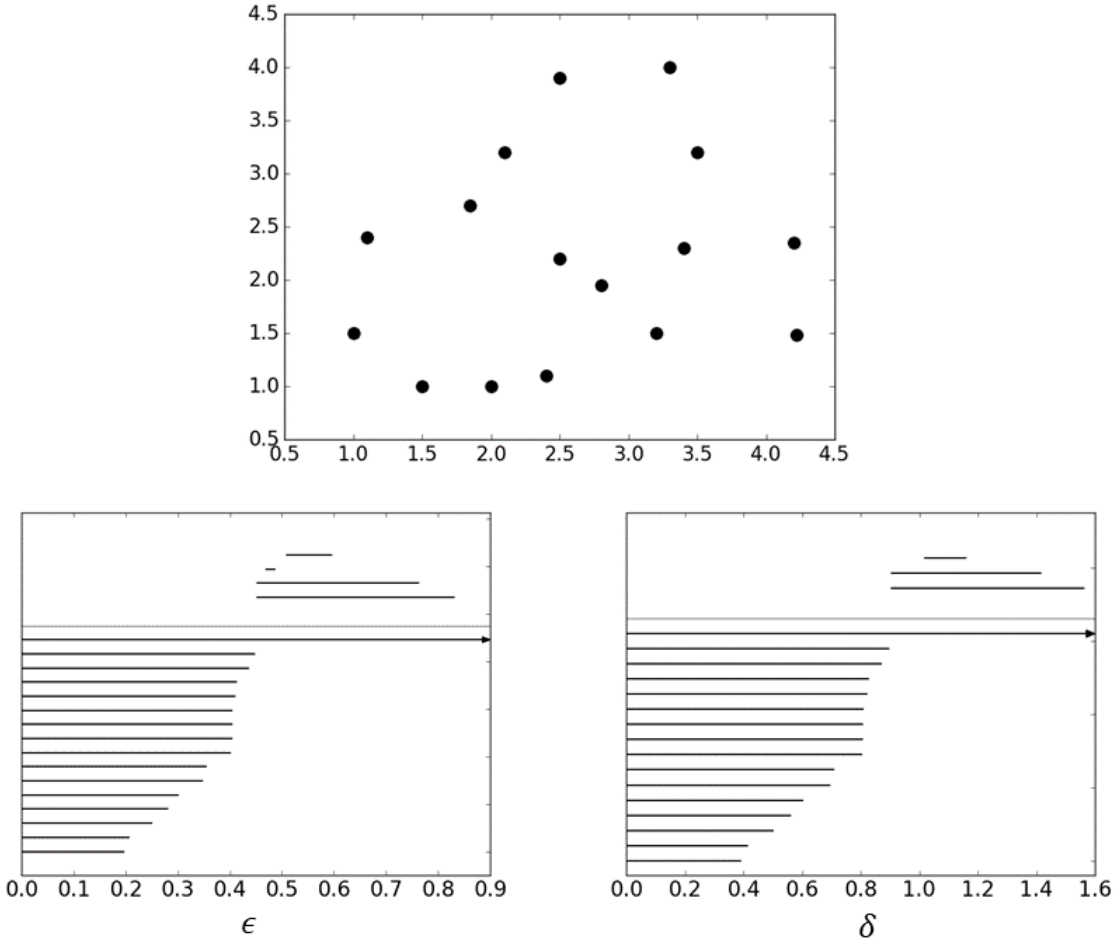

Figure 2.1: Persistence barcodes of alpha complex filtration (bottom left) and Vietoris-Rips complex filtration (bottom right) for the point cloud (top). The top and bottom panels of barcodes are H_1 and H_0 barcodes.

2.1.3 Barcode space metrics

The similarity between persistence barcodes can be quantified by barcode space distances. The most commonly used metrics are the bottleneck distance [43] and the p-Wasserstein distances [44]. The definitions of the two distances are summarized as follows.

The l^∞ distance between two persistence bars $I_1 = [b_1, d_1)$ and $I_2 = [b_2, d_2)$ is defined to be

$$\Delta(I_1, I_2) = \max\{|b_2 - b_1|, |d_2 - d_1|\}.$$

The existence of a bar $I = [b, d)$ is measured as

$$\lambda(I) := (d - b)/2 = \min_{x \in \mathbb{R}} \Delta(I, [x, x)).$$

This can be interpreted as measuring the smallest distance from the bar to the closest degenerate bar whose birth and death values are the same.

For two finite barcodes $B_1 = \{I^1_\alpha\}_{\alpha \in A}$ and $B_2 = \{I^2_\beta\}_{\beta \in B}$, a partial bijection is defined to be a bijection $\theta : A' \to B'$ where $A' \subseteq A$ to $B' \subseteq B$. In order to define the p-Wasserstein distance, we have the following penalty for θ

$$P(\theta) = \left(\sum_{\alpha \in A'} \Delta(I^1_\alpha, I^2_{\theta(\alpha)})^p + \sum_{\alpha \in A \setminus A'} \lambda(I^1_\alpha)^p + \sum_{\beta \in B \setminus B'} \lambda(I^2_\beta)^p \right)^{1/p}.$$

Then the p-Wasserstein distance is defined as

$$d_{W,p}(B_1, B_2) = \min_{\theta \in \Theta} P(\theta),$$

where Θ is the set of all possible partial bijections from A to B. Intuitively, a partial bijection

θ is mostly penalized for connecting two bars with large difference measured by $\Delta(\cdot)$, and for connecting long bars to degenerate bars, measured by $\lambda(\cdot)$.

The bottleneck distance is an L_∞ analogue to the p-Wasserstein distance. The bottleneck penalty of a partial matching θ is defined as

$$P(\theta) = \max\left\{\max_{\alpha\in A'}\left\{\Delta\left(I_\alpha^1, I_{\theta(\alpha)}^2\right)\right\}, \max_{\alpha\in A\setminus A'}\left\{\lambda(I_\alpha^1)\right\}, \max_{\beta\in B\setminus B'}\left\{\lambda(I_\beta^2)\right\}\right\}.$$

The bottleneck distance is defined as

$$d_{W,\infty}(B_1, B_2) = \min_{\theta\in\Theta} P(\theta).$$

2.2 Coupled dynamical systems

The general theory of control of coupled dynamical systems has been well-studied in the literature [148, 92, 192, 197]. A brief review is given in this section.

2.2.1 Oscillators and coupling

We consider a collection of N n-dimensional dynamical systems originally governed by the same equation

$$\frac{d\mathbf{u}_i}{dt} = g(\mathbf{u}_i),\ i = 1, 2, \cdots, N,$$

where $\mathbf{u}_i = \{u_{i,1}, u_{i,2}, \cdots, u_{i,n}\}^T$ is a column vector of size n.

The individual dynamical systems can be coupled with an $N \times N$ coupling matrix A by building an $(N \times n)$-dimensional system. We denote $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_N\}^T$ by $\mathbf{u}$ where $\mathbf{u}_i = \{u_{i,1}, u_{i,2}, \cdots, u_{i,n}\}^T$. The coupled system is an $(N\times n)$-dimensional dynamical system

modeled as

$$\frac{d\mathbf{u}}{dt} = \mathbf{G}(\mathbf{u}) + \epsilon(A \otimes \Gamma)\mathbf{u}, \tag{2.5}$$

where $\mathbf{G}(\mathbf{u}) = \{g(\mathbf{u}_1), g(\mathbf{u}_2), \cdots, g(\mathbf{u}_N)\}^T$, ϵ is a parameter reflecting the coupling strength, and Γ is an $n \times n$ predefined linking matrix specifying how the variables of individual systems are coupled across the systems.

2.2.2 Stability and controllability

The coupled systems are said to be in synchronous state if

$$\mathbf{u}_1(t) = \mathbf{u}_2(t) = \cdots = \mathbf{u}_N(t) = \mathbf{s}(t).$$

The stability can be analyzed using $\mathbf{v} = \{\mathbf{u}_1 - \mathbf{s}, \mathbf{u}_2 - \mathbf{s}, \cdots, \mathbf{u}_N - \mathbf{s}\}^T$ with the following equation obtained by linearizing Eq. (2.5)

$$\frac{d\mathbf{v}}{dt} = [I_N \otimes Dg(\mathbf{s}) + \epsilon(A \otimes \Gamma)]\mathbf{v}, \tag{2.6}$$

where I_N is the $N \times N$ unit matrix and $Dg(\mathbf{s})$ is the Jacobian of g on $\mathbf{s}$. The stability of can be studied by eigenvalue analysis of the coupling matrix A.

2.3 Machine learning

In this section, we review a few supervised learning methods. For a data entry, we assume we have a description of it and it is called the feature. We also assume that there is a label associated to an entry which might be known or unknown. An example scenario is that a

drug can have different effects on patients with the same disease and we would like to predict the efficiency of a drug on some new patients before applying the drug. Here, each patient is an entry and drug effect is the label that we are interested in. What we have is the labeled data on a set of past patients which means that we know the drug outcome on them. And for both past and new patients, we have descriptions of them such as height, body weight, and age. These descriptions are the features which are usually much easier to measure than the label. Then, based on the experience of past patients, we can build a model that can predict labels for new patients. Basically, given a collection of pairs of features and labels $\{(x_i, y_i)\}_{i=1}^{n}$ usually called the training data, a machine learning method tries to build a function that is able to predict the label for a newly given entry based on its feature. The feature can be in structured and unstructured forms. The data can also be in the form where only a metric measuring similarities between data entries is given.

The basic assumption is that entries sharing similar features tend to have similar labels. Generally, a supervised machine learning model have hyperparameters that determines the basic model structure and tunable parameters that are to be optimized with training data. Usually, there is a loss function which measures how well the model is performing upon the training data. Then, the tunable parameters of the model are optimized to minimize the loss function.

2.3.1 K-nearest neighbor algorithm

The k-nearest neighbor algorithm is a direct usage of the assumption that entries sharing similar features tend to have similar labels and it relies on a reliable distance function in the input space. Let X be the input (feature) space and Y be the output (label) space, there is a given distance function or more loosely speaking, a function reflecting distance,

$d(x, x')$, $x, x' \in X$. Given a collection of input output pairs $\{(x_i, y_i)\}_{i=1}^n$, we can predict the output of a newly given input x by comparing it to the inputs of all the training examples. First, an ordering of training data with respect to the newly given input $\{(x_{\alpha_i}, y_{\alpha_i})\}_{i=1}^n$ is determined such that

$$d(x, x_{\alpha_1}) \leq d(x, x_{\alpha_2}) \leq \cdots \leq d(x, x_{\alpha_{n-1}}) \leq d(x, x_{\alpha_n}) \tag{2.7}$$

Here, we consider the regression case where the labels are real numbers. Then, the output y paired with x can be predicted as

$$\frac{1}{k} \sum_{i=1}^{k} y_{\alpha_i},$$

where k is a positive integer chosen by the user. Weights can be added to accommodate the hierarchy in the ordering,

$$\sum_{i=1}^{k} w_i y_{\alpha_i},$$

and a simple choice of weights is the inverse of distance $w_i = \frac{1}{d(x,x_{\alpha_i})} / (\sum_{i=1}^{k} \frac{1}{d(x,x_{\alpha_i})})$. There are also other more complicated weights that are more robust and stable.

2.3.2 Decision tree

We review the idea of decision learning tree in the case of regression. When the feature is in the form of a vector, say $\mathbf{x} \in \mathbb{R}^d$. If the impacts of elements of $\mathbf{x}$ on the output y are assumed to be additive, a linear regression model can be used. A generalization of linear model to higher order polynomials can accommodate higher order interactions among the elements of the input. However, model complexity increases dramatically as higher order interactions are considered. Decision tree model provides an alternative to address higher

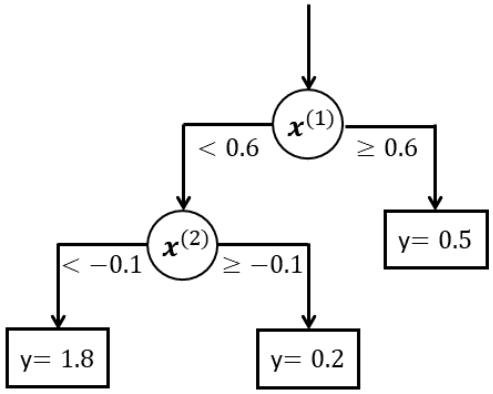

Figure 2.2: Example regression decision tree.

order interactions by partition the feature space instead of exploring the space of high order polynomials with many variables.

The nodes that are not leaves in a decision tree are associated to elements of the input feature vector and the leaf nodes are assigned prediction values of the label. When flowing from the root node to a leaf node, the value of the corresponding feature vector element is evaluated against predefined criteria to determine which child node to go to. Once reached Fig. 2.2 gives an example regression decision tree.

Decision trees can be constructed with greedy algorithm from top down. At each construction stage, a feature element with a splitting criteria is selected to most effectively reduce the loss function [166]. There are techniques to prevent overfitting by restricting model complexity. For example, one may set an upper bound for number of branches or tree depth. One can also set a lower bound for number of training examples associated to each leaf node. One advantage of decision tree is that one element of feature vector is used at each node so that the features can have completely different meanings and magnitudes.

2.3.3 Ensemble of trees

In many problems, there can be thousands of features and large amount of data that are too diverse and complicated for one decision tree to handle. A group of weak learners can be combined to increase model capacity and this is called ensemble method. Here, a set of decision trees can be used instead using one single tree. There are different ensemble methods for decision trees and we review two widely used constructions here, random forest and gradient boosting trees.

Random forest relies on bootstrap aggregation where independent trees are built on random resampling of the original training data with replacement. Then, a consensus model is constructed by taking the average output of the independent trees. There are techniques designed against overfitting such as using the resampling of only a subset of training data for each tree. Since the trees are trained independently, random forest can be efficiently constructed by constructing the trees in parallel.

Another approach is by gradient boosting where one tree is added at a time minimizing the current loss. Let $L(y, F(x))$ be a loss function measuring the quality of prediction $F(x)$ against true label y. A general gradient boosting procedure is to first initialize a model denoted F_0 which can be a constant function. Then, F_ℓ is updated to $F_{\ell+1}$ by training a base model h_ℓ against the gradient of current loss $\{(x_i, -\frac{\nabla L(y_i, F(x_i)}{\nabla F(x_i)}\big|_{F=F_\ell})\}$. The model in the next iteration takes the form of $F_{\ell+1} = F_\ell + \gamma_\ell h_\ell$ and γ_ℓ is obtained by solving the minimization problem

$$\min_{\gamma} \sum_i L(y_i, F_\ell(x_i) + \gamma h_\ell(x_i)).$$

A real number between 0 and 1 called the learning rate is usually multiplied to γ_ℓ when updating the model for robustness.

2.3.4 Deep learning

A neural network is a generally nonlinear composite function constructly mainly with linear transformations and nonlinear activation functions.

A regular neuron is a processing unit that applies a nonlinear activation function on top of a linear function of the outputs of neurons in the previous layer that are connected to it. A regular neural network is constructed by stacking densely connected layers where every pair of neurons from adjacent layers are connected. Assume that we have a collection of n neurons in layer i and denote the output of the jth neuron by $\mathcal{N}_j^{(i)}$, then the output of the kth neuron in layer $i+1$ is computed as

$$\mathcal{N}_k^{(i+1)} = \phi\left(\sum_{j=1}^{n}(w_{j,k}^{(i+1)}\mathcal{N}_j^{(i)} + b_k^{(i+1)})\right),$$

where $w_{j,k}^{(i+1)}$ and $b_k^{(i+1)}$ are tunable parameters and ϕ is the chosen activation function. Commonly used activation functions include rectified linear unit ($\max(0,x)$), sigmoid ($\frac{1}{1+e^{-x}}$), and hyperbolic tangent ($\frac{e^x+e^{-x}}{e^x-e^{-x}}$). The number of parameters increase rapidly when more neurons and layers are added to the model.

Originally inspired by vision system in biology, convolutional neural network is a widely used architecture especially good at dealing with image data or image-like analogues. Only local connections are allowed and the weights are shared across different locations. The setup of a single neuron is similar to the regular neural network but the information carried by a neuron can be a vector instead of a scalar and the weight and bias parameters are then higher dimensional counterparts. The activation function is applied elementwise on the vector. Fig. 2.3 shows the architecture of a dense layer and a convolution layer. We discuss more details of convolutional neural network in the 1-dimensional case. Consider an $n \times m$

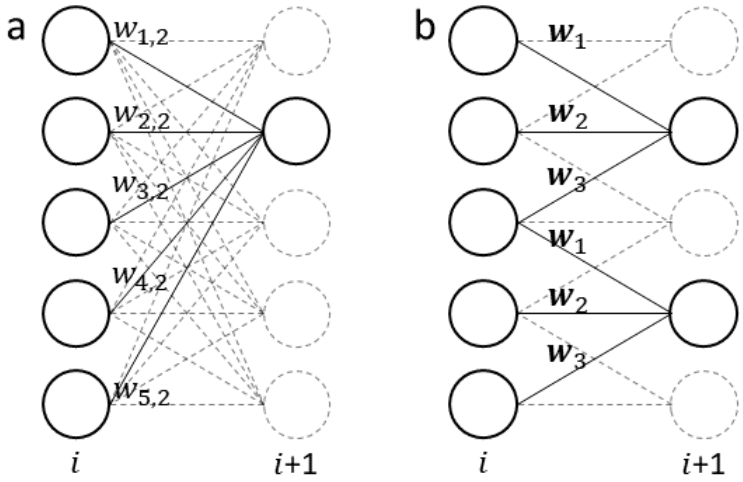

Figure 2.3: a. Dense layer. b. Convolution layer.

second order tensor $\mathbf{V}$, where n is the number of digits along that one dimension and m is number of channels for each digit. With a predefined window size w, a convolutional filter $\mathbf{F}$ can be represented by a $w \times m$ second order tensor. By moving the window of size w along the one dimension of $\mathbf{V}$, a sequence of N_f second order tensors, which are subtensors of V, are obtained and can be concatenated to form an $N_f \times w \times m$ third order tensor $\mathbf{T}$. The filter $\mathbf{F}$ operated on $\mathbf{T}$ results in a first order tensor $\mathbf{T}_{ijk}\mathbf{F}_{jk}$ by tensor contraction. Concatenating the outputs of n_f filters gives an $N_f \times n_f$ second order tensor. Generally speaking, a 1D convolution layer takes an $n \times m$ tensor and outputs an $N_f \times n_f$ tensor. The bias terms and activation functions are omitted in the illustration above for simplicity.

Another technique we are going to make use of is multitask learning. A practical multitask learning model can be achieved by branching out from the shared lower layers. An example architecture of multitask learning neural network is shown in Fig. 2.4.

Feedforward neural networks are usually trained by backpropagation where the error of the output layer is calculated and is propagated backward through the network to update its weights. One popular approach of training a neural network is the stochastic gradient decent

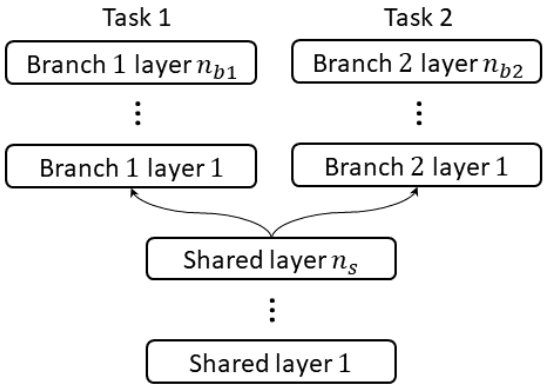

Figure 2.4: A practical neural network architecture for multitask learning.

(SGD) method. Let Θ be the parameters in the network and $L(\Theta)$ be the loss function that is to be minimized. SGD method updates Θ_i to Θ_{i+1} from step i to step $i+1$ as

$$\Theta_{i+1} = \Theta_i - \tau \nabla_\Theta L(\Theta_i; X_s, Y_s),$$

where τ is the learning rate, X_s and Y_s are the input and target of the sth example of the training set. In practice, the training set (X, Y) is often split into mini-batches $\{(X_s, Y_s)\}_{s \in S}$. SGD method then goes through each mini-batch instead of going through only one example at a time. When the landscape of the objective function is like a long steep valley, momentum is added to accelerate convergence of the algorithm. The updating scheme can therefore be changed to

$$\Delta\Theta_i = \Theta_i - \Theta_{i-1},$$
$$\Theta_{i+1} = \Theta_i - (1-\eta)\tau \nabla_\Theta \mathcal{L}(\Theta_i; X_s^i, Y_s^i) + \eta \Delta\Theta_i,$$

where $0 \leq \eta \leq 1$ is a scalar coefficient for the momentum term. Many techniques have been proposed to address the overfitting problem such as dropout and regularization term.

2.4 Biomolecular modeling

We review some basics of biomolecular modeling relevant to this dissertation in this section involving both bioinformatics methods and physical models.

2.4.1 Proteins and small molecules

Proteins are made of sequences of amino acids and play very important roles in living things. Just listing a few of their functions, they serve as structural supporting elements, drive biological processes, and sense chemical changes. The amino acid sequences determine local secondary structures including alpha helix, beta sheet, and random coil. For regular proteins, they usually fold into a relatively stable structure which we call the folded state. There are also disordered or partially disordered proteins that do not have stable structures in certain conditions.

Small molecules also play an important role in biological processes often by interacting with other macromolecules such as proteins and nucleic acids. They may serve as triggering signals for certain processes or inhibit certain processes. Actually, many working drugs are small molecules. Being able to accurately describe small molecules and small molecule-macromolecule complexes is important for computational molecular biology or drug design.

2.4.2 Physical modeling

Solvation free energy

Electrostatic forces play an essential role in protein folding. Electrostatic solvation free energy is an important component of electrostatic interactions. Among all kinds of approaches, continuum solvation models achieve a favorable trade-off between accuracy and

efficiency. Poisson-Boltzmann model is used to quantify the polar part of energy change in solvation process, that is the process of moving a molecule from vacuum to solvent environment.

Poisson-Boltzmann model uses Boltzmann distribution to model ion density and the equation is formulated as

$$\begin{aligned} -\nabla \cdot (\epsilon(\mathbf{x})\nabla\phi(\mathbf{x})) + \overline{k}^2(\mathbf{x})\sinh(\phi(\mathbf{x})) &= \sum_{i=1}^{N} q_i \delta(\mathbf{x}-\mathbf{x}_i), \\ \phi(\infty) &= 0, \end{aligned} \tag{2.8}$$

where ϕ is the electrostatic potential, ϵ is the dielectric constant, $\overline{k}$ is the modified Debye-Hückel screening factor, δ is the Dirac delta function, and q_i and $\mathbf{x}_i$ are the partial charge and location of each fixed atom in the molecule. The second term and the right hand side in equation 2.8 model charge density of mobile ion and fixed molecule respectively. The value of the partial charges can be assigned by precomputed force fields. In computation, a bounding box Ω is taken as computation domain and is composed of solvent domain Ω_s and solute domain Ω_m. One definition of the dielectric constant ϵ is to take constant values on Ω_s and Ω_m. The solution ϕ to equation 2.8 should satisfy the jump condition across the interface Γ between Ω_s and Ω_m,

$$\begin{aligned} [\phi]_\Gamma &:= \phi|_{\Omega_s}(\mathbf{x}) - \phi|_{\Omega_m}(\mathbf{x}) = 0, \quad \mathbf{x} \in \Gamma \\ [\epsilon\nabla\phi]_\Gamma &:= \epsilon_s \nabla\phi|_{\Omega_s}(\mathbf{x}) \cdot \mathbf{n} - \epsilon_m \nabla\phi|_{\Omega_m}(\mathbf{x}) \cdot \mathbf{n} = 0, \quad \mathbf{x} \in \Gamma \end{aligned} \tag{2.9}$$

The solvation free energy modeled by Poisson-Boltzmann model can then be computed as

$$\Delta G_{pol}^{sol} = \frac{1}{2}\sum_{i=1}^{N} q_i(\phi(\mathbf{x}_i) - \phi_{vac}(\mathbf{x}_i)), \tag{2.10}$$

where ΔG_{pol}^{sol} is the polar component of the solvation free energy and ϕ_{vac} is the electrostatic potential of the molecule in vacuum environment which is obtained by solving a Poisson equation with uniform dielectric constant for vacuum.

Electrostatic interaction

The electrostatics of pairwise interactions in homogeneous media can be modeled with the Coulomb model. Given two atoms with partial charges q_i and q_j with a distance of r_{ij} apart from each other. The electrostatic force F_{clb} between these two atoms can be described by Coulomb's law

$$F_{\text{clb}} = k_e \frac{q_i q_j}{r_{ij}^2},$$

where k_e is the Coulomb's constant.

Van der Waals interaction

Van der Waals forces describe interaction between atoms that are irrelevant to covalent bond or electrostatic interaction. Lennard-Jones potential is used to model the Van Der Waals interaction energy between atoms and the energy between the i^{th} and the j^{th} atom is defined as

$$G_{i,j} = \epsilon_{ij}\left(\left(\frac{r_i + r_j}{|\mathbf{x}_i - \mathbf{x}_j|}\right)^{12} - 2\left(\frac{r_i + r_j}{|\mathbf{x}_i - \mathbf{x}_j|}\right)^{6}\right), \tag{2.11}$$

where r_i, r_j are Van der Waals radii of the atoms and ϵ_{ij} is the depth of the potential well which differs for different atoms. Since atomic features are constructed and the interactions between different pairs of atom types are modeled separately, the determination of the

parameter ϵ_{ij} is left with the machine learning algorithm.

Molecule surface area

While the polar part of solvation free energy is modeled with the Poisson-Boltzmann theory, the nonpolar part is affected by several factors. The surface area of the molecule is believed to relate to nonpolar part of solvation free energy.

2.4.3 Sequence tools

BLOSUM matrix

A BLOSUM matrix is a substitution matrix for protein sequence alignments. It scores the switch of any pair of amino acids and can thus describe the favorability of a certain point mutation. In this work, BLOSUM62 which models moderately related proteins is used.

PSSM matrix

The BLOSUM matrix is a general scoring matrix which does not consider residue position or the alignment of the entire sequence. A position-specific scoring matrix (PSSM) is used to fill this information gap. BLAST searches for the query sequence is performed iteratively and the substitution scores are updated according to the matched sequences found which is position-specific along the sequence.

Chapter 3

TopologyNet: Deep convolutional neural networks based on topology for biomolecular property predictions.

3.1 Introduction

Understanding the structure-function relationships of biomolecules is fundamentally important in computational biophysics and experimental biology. As such, methods that can robustly predict biomolecular properties, such as protein-ligand binding affinity and protein stability change upon mutation from three-dimensional (3D) structures are important tools to help us understand this relationship. Numerous approaches have been developed to unveil the structure-function relationship. Physics based models make use of fundamental laws of physics, i.e., quantum mechanics, molecular mechanics, continuum mechanics, multiscale modeling, statistical mechanics, thermodynamics, etc, to investigate structure-function relationships and predict function from structure. Physical methods provide important insights and are indispensable for understanding the relationships between protein structure and function.

Physical models for complex biomolecular systems are made computationally tractable

often by making simplifications to the real-world systems. However, there are so many factors that could affect the properties of biomolecular systems and a factor that is crucial to one type of systems can have minor effect on another. Therefore, it is hard to tune a physical model to handle diverse systems. On the other hand, a model with the capability of handling diverse systems is desirable in many realistic applications. For example, the computational screening of drug candidates involves the scanning of a library containing millions of diverse molecules. The large and diverse datasets bring the opportunity of enabling the model learn from data to automatically extract the relevant factors and the unknown relationships. The machine learning models driven by data make relatively fewer assumptions about the importance of the factors to the systems and there is usually high flexibility in the models so that nonlinear and high-order interactions among features can be recognized. In this chapter, we focus on deep learning models for its excellent ability of handling complex datasets and its flexible architectures enabling the combination of relevant learning tasks. Deep learning has fueled the rapid growth in several areas of data science [110, 138]. Notably, deep learning can automatically extract optimal high level features and discover intricate structures in large data sets. The capability of handling data with underlying spatial dimensions hierarchically has lead to breakthroughs in deep convolutional neural networks in image processing, video, audio and computer vision [107, 170].

Given multiple learning tasks, multi-task learning (MTL) [34] provides a powerful tool to exploit the intrinsic relatedness among learning tasks, transfer predictive information among tasks, and achieve better generalized performance than single task learning. During the learning stage, MTL algorithms seek to learn a shared representation (e.g., shared distribution of a given hyper-parameter [65], shared low-rank subspace [64], shared feature subset [118] and clustered task structure [213]), and use the shared representation to bridge

between tasks and transfer knowledge. MTL has applications to the bioactivity of small molecular drugs [180, 120, 184] and genomics [49]. Linear regression based MTL heavily depends on well crafted features, while neural network based MTL allows more flexible task coupling and is able to deliver satisfactory results with a large number of low level features provided such features have the representative power of the problem.

For complex 3D biomolecular data, the physical features used in machine learning vary greatly in their nature. Typical features are generated from geometric properties, electrostatics, atom types, atomic partial charges, and graph theory based properties [188]. Such manually extracted features can be used in a deep neural network, but the performance heavily relies on feature engineering. In contrast, convolutional neural networks learn to extract high level representations hierarchically from low level features while maintaining the underlying spatial relationships. However, the cost is huge for directly applying convolutional neural network to the 3D biomolecules, especially if long-range interactions are included. A major obstacle in the development of deep learning nets for 3D biomolecular data is their entanglement between geometric complexity and biological complexity.

Most theoretical models for the study of structure-function relationships of biomolecules are based on geometric modeling techniques. Mathematically, these approaches exploit local geometric information, i.e., coordinates, distances, angles, areas, and sometimes curvatures [139] for the physical modeling of biomolecular systems. Indeed, the importance of geometric modeling for structural biology [67], and biophysics cannot be overemphasized. However, geometry based models often contain too much structural detail and are frequently computationally intractable for large structures or datasets. In many biological problems, such as the opening or closing of ion channels, the association or dissociation of binding ligands, the folding or unfolding of proteins, and the symmetry breaking or formation of virus capsids, ob-

vious topological changes exist. In fact, one only needs qualitative topological information to understand many physical and biological functions. In short, *topology-function relationships* exist in many biomolecular systems.

Topology offers entirely different approaches and could provide significant simplification of biomolecular data [168, 216, 175, 51, 88, 56, 52, 169]. The study of topology deals with the connectivity of different components in a space, and characterizes independent entities, rings and higher dimensional faces within the space [98]. Topological methods produce a high level of abstraction to many biological processes. For example, the opening and closing of ion channels, the assembly or disassembly of virus capsids, the folding and unfolding of proteins, and the association or dissociation of ligands are reflected by topological changes. The fundamental task of topological data analysis is to extract topological invariants, namely the intrinsic features of the underlying space, of a given data set without additional structure information. Examples include covalent bonds, hydrogen bonds, van der Waals interactions, etc. In this work, we use persistent homology which is a relatively new branch of algebraic topology that embeds multiscale geometric information in topological invariants to achieve an interplay between geometry and topology. It creates a variety of topologies of a given object by varying a filtration parameter, such as the radii of balls centered at the nodes or the level set of a surface function. As a result, persistent homology can capture topological structures continuously over a range of spatial scales.

The objective of this chapter is to introduce a new framework for the structure based biomolecular property predictions using element-specific persistent homology, and convolutional and multi-task neural networks. In this framework, element-specific persistent homology reduces geometric and biological complexities and provides a sufficient and structured low level representation for neural networks. Given this representation, convolutional neural

networks can then learn from data to extract high level representations of the biomolecular systems, while retaining the spatial relationships, and construct mappings from these representations to the target properties. For the prediction problems whose available datasets are small, multi-task learning by jointly learning the related prediction problems with larger available datasets helps to extract a proper high level representation for the target applications. The element-specific treatment is inspired by the RF-score method [115] for binding affinity prediction. Element-specific persistent homology is originated in our previous work using classic machine learning methods. [23, 24] In this work, we further develop topology based neural network (TopologyNet) models for the predictions of biomolecular structure-function relationships. Specifically, we integrate ESPH and convolutional neural networks (CNNs) to improve modern methods for protein-ligand binding affinity and protein mutation impact predictions from 3D biomolecular data. In this approach, topological invariants are used to reduce the dimensionality of 3D biomolecular data. Additionally, element-specific persistent barcodes offer image-like topological representations to facilitate convolutional deep neural networks. Moreover, biological information is retained by element-specific topological fingerprints and described in multichannels in our image like representation. Furthermore, convolutional neural networks uncover hidden relationships between biomolecular topological invariants and biological functions. Finally, a multi-task multichannel topological convolutional neural network (MM-TCNN) framework is introduced to exploit the relations among various structure-function predictions and enhance the prediction for problems with small and noisy training data. Our hypothesis is that many biomolecular predictions share a common set of topological fingerprints representations and are highly correlated to each other. As a result, multi-task deep learning by simultaneous training for globular proteins and membrane proteins improves upon existing predictions for the mutation induced stability

changes of membrane proteins whose training data is relatively small.

3.2 Methods

In this section, we give a brief explanation of persistent homology before introducing topological representations of protein-ligand binding and protein changes upon mutation. Multi-channel topological deep learning and multi-task topological deep learning architectures are constructed for binding affinity and mutation impact predictions. The source codes with examples of feature construction for the binding problem and the mutation problem can be found in supplementary material (S3_Code and S4_Code) of [26]. The network architectures, parameters, and training procedures are listed in S2_Text of [26]. Some auxiliary features such as sequence features are used to enhance the models for the mutation applications. The description of the auxiliary features together with pseudocode for the mutation application are listed in S1_Text of [26].

3.2.1 Persistent homology

Simplicial homology gives a computable way to distinguish one space from another in topology and is built on simplicial complexes which can be used to extract topological invariants in a given data set. A simplicial complex K is a topological space that is constructed from geometric components of a data set, including discrete vertices (nodes or atoms in a protein), edges (line segments or bonds in a biomolecule), triangles, tetrahedrons and their high dimensional counterparts, under certain rules. Specifically, a 0-simplex is a vertex, a 1-simplex an edge, a 2-simplex a triangle, and a 3-simplex represents a tetrahedron. The identification of connectivity of a given data set can follow different rules which leads to,

for example, Vietoris-Rips (VR) complex, Čech complex and alpha complex. The linear combination of k-simplexes is called k-chain, which is introduced to associate the topological space, i.e., simplicial complex, with algebra groups, which further facilitate the computation of the topological invariants (i.e., Betti numbers) in a given data set. Specifically, the set of all k-chains of a simplicial complex K are elements of a chain group, which is an abelian group with a modulo-2 addition operation rule. Loosely speaking, a boundary operator systematically eliminates one vertex from the k-simplex at a time, which leads to a family of abelian groups, including the kth cycle group and the kth boundary group. The quotient group of the kth cycle group and the kth boundary group is called the kth homology group. The kth Betti number is computed for the rank of the kth homology group.

Persistent homology is constructed via a filtration process, in which the connectivity of the given data set is systematically reset according to a scale parameter. More specifically, a nested sequence of subcomplexes is defined via a filtration parameter, such as the growing radius of protein atoms located at their initial coordinates. For each subcomplex, homology groups and the corresponding Betti numbers can be computed. Therefore, the evolution of topological invariants over the filtration process can be recorded as a barcode [79] or a persistence diagram. For a given data set, barcodes represent the persistence of its topological features over different spatial scales.

3.2.2 Topological representation of biomolecules

Topological fingerprints

A basic assumption of persistent homology as applied to biomolecular function prediction is that 1D biomolecular persistent barcodes are able to effectively characterize 3D biomolecular structures. We call such barcodes topological fingerprints (TFs) [198, 195]. Fig. 3.1

illustrates the TFs of a wild type protein (PDB:1hmk) and its mutant obtained from persistent homology calculations using the VR complex. The mutation (W60A) occurred at residue 60 from Trp to Ala is shown at Figs. 3.1a and b. A large residue (Trp) at the protein surface is replaced by a relatively small one (Ala). The corresponding barcodes are given in Figs. 3.1c and d, where three panels from top to bottom are for Betti-0, Betti-1, and Betti-2, respectively. The barcodes for the wild type are generated using heavy atoms within 6Å from the mutation site. The mutant barcodes are obtained with the same set of heavy atoms in the protein except for those in the mutated residue. In two Betti-0 panels, the difference in the number of bars is equal to the difference in the number of heavy atoms between the wild type and mutant. Broadly speaking, the lengths of short bars reflect the bond length of the corresponding heavy atom. Therefore, in both the wild type protein and the mutant, bond lengths for most heavy atoms are smaller than 1.8Å. Additionally, bars that end between 1.8Å and 3.8 Å might correlate with hydrogen bonds. Comparing Fig. 3.1c and d, one can easily note the increase in the number of bars that end in the range of 1.8 - 3.8 Å in the mutant, which indicates a less compact atom arrangement. In Betti-1 and Betti-2 panels, the mutant has fewer bars than the wild type does because a smaller surface residue at 60 creates fewer ring and cavity contacts with the rest of the protein.

Element-specific persistent homology

The all heavy atom topological representation of proteins does not provide enough biological information about protein structures, such as bond length distribution of a given type of atoms, hydrogen bonds, hydrophobic and hydrophilic effects, etc. Therefore, we use the element-specific topological fingerprint (ESTF) to offer a more detailed characterization of protein-ligand binding and protein mutation. For example, Betti-1 and Betti-2 ESTFs from carbon atoms are associated with hydrophobic interaction networks in biomolecules.

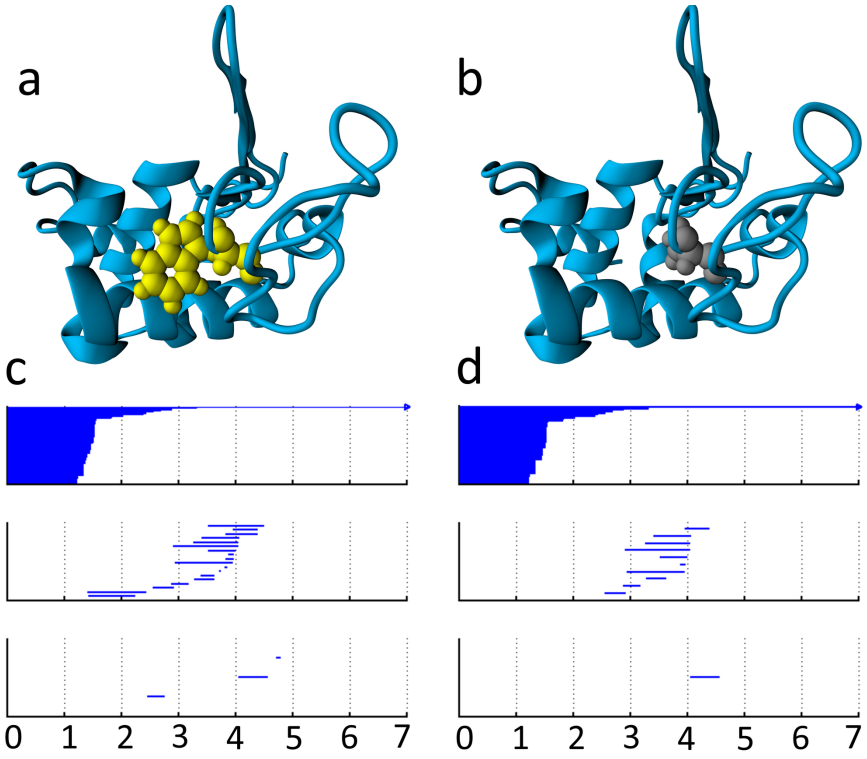

Figure 3.1: An illustration of barcode changes from wild type to mutant proteins.[26]
a: The wild type protein (PDB:1hmk) with residue 60 as Trp. b: The mutant with residue 60 as Ala. c: Wild type protein barcodes for heavy atoms within 6 Å of the mutation site. Three panels from top to bottom are Betti-0, Betti-1, and Betti-2 barcodes, respectively. The horizontal axis is the filtration radius (Å). d: Mutant protein barcodes obtained similarly to those of the wild type.

Similarly ESTFs between nitrogen and oxygen atoms correlate to hydrophilic interactions and/or hydrogen bonds in biomolcules. However, hydrogen atoms are typically absent from structures in the PDB and thus are not used in our data driven ESTF description. For proteins, commonly occurring heavy atom types include $\mathrm{C}, \mathrm{N}, \mathrm{O}$, and S. For ligands, we use 9 commonly occurring atom types, namely $\mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{S}, \mathrm{P}, \mathrm{F}, \mathrm{Cl}, \mathrm{Br}$, and I. To characterize the interactions between protein and ligand binding, we construct cross protein-ligand ESTFs such that one type of heavy atoms is chosen from the protein and the other from the ligand. Therefore, there are a total of 36 sets of ESTFs in each topological dimension. For mutation characterization, we describe the interactions between mutated residue and the rest of the protein and arrive at 9 sets of ESTFs in each topological dimension considering { C, N, O } for protein atoms. Similarly, we generate 9 sets of cross ESTFs in each topological dimension from the wild type protein to study the interactions between the residue to be mutated and the rest of the protein. However, high dimensional Betti-1 and Betti-2 invariants require the formation of high order complexes. As non-carbon atoms do not occur very often, Betti-1 and Betti-2 ESTFs are generated for all carbon atoms or all heavy atoms, except specified.

The TFs and ESTFs are originally stored as collections of barcodes denoted by $\mathbb{B}(\alpha, \mathcal{C}, \mathcal{D})$ with α labeling the selection of atoms depending on atom types and affiliations (i.e., protein, ligand or mutated residue). $\mathcal{C}$ denotes the type of simplicial complex (i.e., VR complex or alpha complex) and $\mathcal{D}$ indicates the dimension, such as Betti-0, Betti-1, or Betti-2. A collection of barcodes can have any number of barcodes and thus can not be directly fed to deep learning models. Additionally, as shown in Fig. 3.1, it is important to keep track of the birth, death, and persistence patterns of the barcodes, because this information is associated with the bond length, ring or cavity size, flexibility and steric effect. Moreover, Jeffrey suggested that there are strong, moderate and weak hydrogen bond interactions with donor-

acceptor distances of 2.2-2.5Å, 2.5-3.2Å, and 3.2-4.0Å, respectively [96]. To this end, we construct structured vectors $\mathbf{V}^{\mathrm{b}}$, $\mathbf{V}^{\mathrm{d}}$, and $\mathbf{V}^{\mathrm{p}}$ to respectively describe the birth, death, and persistent patterns of the barcodes in various spatial dimensions. Practically, the filtration interval $[0, L]$ is divided into n equal length subintervals and the patterns are characterized on each subinterval. The description vectors are defined as

$$\begin{aligned}
\mathbf{V}_i^{\mathrm{b}} &= \left\|\{(b_j, d_j) \in \mathbb{B}(\alpha, \mathcal{C}, \mathcal{D}) | (i-1)L/n \leq b_j \leq iL/n\}\right\|, \ 1 \leq i < n, \\
\mathbf{V}_i^{\mathrm{d}} &= \left\|\{(b_j, d_j) \in \mathbb{B}(\alpha, \mathcal{C}, \mathcal{D}) | (i-1)L/n \leq d_j \leq iL/n\}\right\|, \ 1 \leq i < n, \\
\mathbf{V}_i^{\mathrm{p}} &= \left\|\{(b_j, d_j) \in \mathbb{B}(\alpha, \mathcal{C}, \mathcal{D}) | (i-1)L/n \geq b_j, \, iL/n \leq d_j\}\right\|, \ 1 \leq i \leq n,
\end{aligned} \tag{3.1}$$

where $\|\cdot\|$ is cardinality of sets. Here b_j, d_j are birth and death of bar j. The three types of representation vectors are computed for sets of Betti-1 and Betti-2 bars. For Betti-0 bars, since their birth positions are uniformly 0, only $\mathbf{V}^{\mathrm{d}}$ needs to be addressed. To characterize pairwise interactions between atoms, it is convenient to simply use pairwise distance information between atoms. The corresponding image-like representation, denoted by $\mathbf{V}^{\mathrm{r}}$, can be constructed similarly to $\mathbf{V}^{\mathrm{d}}$ by substituting the set of barcodes by a collection of distances between the atom pairs of interest. It should be noted that $\mathbf{V}^{\mathrm{r}}$ is not equivalent to $\mathbf{V}^{\mathrm{d}}$ in most simplicial complex setups. Generally speaking, $\mathbf{V}^{\mathrm{r}}$ also reflects the 0th order topological connectivity information. It is used as the characterization of 0th order connectivity of the biomolecules in the applications shown in this work. Finally, we let X_s denote all the feature vectors for the sth sample and let Y_s denote the corresponding target value.

Image-like multichannel topological representation

To feed the outputs of TFs into the convolutional neural network, the barcodes are

transformed to a 1D-image-like representation with multiple channels. Topological feature vectors , $\mathbf{V}^{\mathrm{b}}$, $\mathbf{V}^{\mathrm{d}}$, and $\mathbf{V}^{\mathrm{p}}$, can be viewed as one-dimensional (1D) images. Each subinterval in the filtration axis represents a digit (or pixel) in the 1D-image-like representation. Such a treatment of topological features describes the topological information with appropriately chosen resolution of L/n. Meanwhile, the chemical information in the ESTFs of $\mathbb{B}(\alpha, \mathcal{C}, \mathcal{D})$ are described by multiple channels in the 1D-image-like representation, which is similar to the RGB color image representation. However, in our description, each pixel is associated with m channels to describe different element type, protein mutation status (i.e., wild type and mutant), topological dimension (i.e., Betti-0, Betti-1 and Betti-2), and topological event (i.e., birth, death, and persistence). Each element in the 1D-image-like representation is standardized to have zero mean and unit variance among the data sets. This 1D-image-like topological representation can be easily transferred among problems such as protein-ligand binding affinity modeling and prediction of protein stability change upon mutation. Traditional machine learning approaches require manual extraction of features for each domain of application. When the convolutional neural network is applied, the convolution layers identify local patterns of atomic interactions and the fully connected layers then extract higher level descriptions of the system by combining local patterns at various distance scales.

Multichannel topological invariants for protein-ligand binding prediction

In computation, the binding affinity, or alternatively the binding free energy, can be modeled via an energy cycle as shown in Fig. 3.2 where the main contributors to the process are intermolecular interactions and solvation effects. In this work, we consider the set of element types $\mathbb{L}^{\mathrm{e}} = \{\mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{S}, \mathrm{P}, \mathrm{F}, \mathrm{Cl}, \mathrm{Br}, \mathrm{I}\}$ contained in ligands and $\mathbb{P}^{\mathrm{e}} = \{\mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{S}\}$

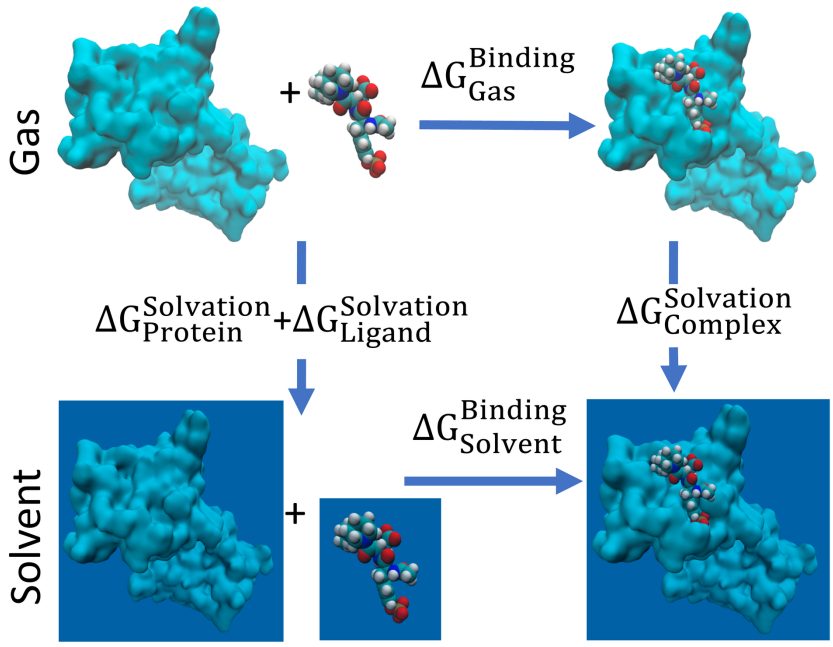

Figure 3.2: Energy cycle of protein-ligand binding free energy modeling.[26]

contained in proteins. We define an opposition distance between two atoms a_i and a_j as

$$d^{op}(a_i, a_j) = \begin{cases} d(a_i, a_j) & , A(a_i) \neq A(a_j) \\ \infty & , A(a_i) = A(a_j) \end{cases}, \tag{3.2}$$

where $d(\cdot, \cdot)$ is Euclidean distance between two atoms and $A(\cdot)$ denotes the affiliation of an atom which is either a protein or a ligand.

The ESTFs used in this application are summarized in Table 3.1. The structured description vectors of the ESTFs are generated according to the definition given in Eq (3.1). As shown in Table 3.1, five sets of ESTFs are constructed. The differences between the description vectors arising from Set 2 and Set 3, and between those arising from Set 4 and Set 5 are also employed as representation vectors to address the impact of ligand binding resulting in a total of 72 representation vectors (i.e., channels) forming the 1D-image-like

representation of the protein-ligand complex. Pairwise interactions are characterized for the 36 element pairs with {C, N, O, S} for the protein and {C, N, O, S, F, P, Cl, Br, I} for the ligand with $\mathbf{V}^d$ providing 36 channels. The birth ($\mathbf{V}^b$), death ($\mathbf{V}^d$), and persistence ($\mathbf{V}^p$) for Betti-1 and Betti-2 barcodes are computed for carbon atoms and all heavy atoms of the protein and the protein-ligand complex which results in 24 channels. The difference between the characterization of the protein and the protein-ligand complex accounts for another 12 channels. Thus, we have a total of 72 channels. Here, 0-dimensional TFs describe intramolecular interactions between the protein and ligand. All heavy atom TFs delineate the geometric effect of protein-ligand binding. The TFs of carbon atoms account for hydrophobic effects and also implicitly reflect the solvation effects. The distance scale interval, $[0, 50]$ Å is divided into bins of length 0.25 Å.

Set	Atoms used	Distance	Complex	Dim.
1	$\{a \in \mathbb{P} \vert T(a) = e_{\mathrm{P}}\} \cup \{a \in \mathbb{L} \vert T(a) = e_{\mathrm{L}}\}, e_{\mathrm{P}} \in \mathbb{P}^{\mathrm{e}}, e_{\mathrm{L}} \in \mathbb{L}^{\mathrm{e}}$	d^{op}	-	0
2	$\{a \in \mathbb{P} \vert T(a) \in \mathbb{P}^{\mathrm{e}}\}$	Euclidean	Alpha	1,2
3	$\{a \in \mathbb{P} \vert T(a) \in \mathbb{P}^{\mathrm{e}}\} \cup \{a \in \mathbb{L} \vert T(a) \in \mathbb{L}^{\mathrm{e}}\}$	Euclidean	Alpha	1,2
4	$\{a \in \mathbb{P} \vert T(a) = \mathrm{C}\}$	Euclidean	Alpha	1,2
5	$\{a \in \mathbb{P} \vert T(a) = \mathrm{C}\} \cup \{\mathrm{a} \in \mathbb{L} \vert \mathrm{T(a)} = \mathrm{C}\}$	Euclidean	Alpha	1,2

Table 3.1: Topological representations of protein-ligand complexes.
$\mathbb{P}$ and $\mathbb{L}$ are sets of atoms in protein and in ligand. $T(\cdot)$ denotes element type of an atom. e_{P} is an element type in protein and e_{L} is an element type in ligand. "Complex" refers to the type of simplicial complex used and "Dimension" refers to the dimensionality of a topological invariant.

Multichannel topological invariants for the prediction of protein folding free energy change upon mutation

Modeling protein folding free energy change upon mutation basically involves the unfolded states and folded structures of the mutant and the wild type as shown in Fig. 3.3. Since unfolded states of proteins are highly dynamic which significantly increases the modeling cost due to the need of sampling over large conformation space, we only analyze the folded states of the mutants and the wild type proteins in this application. Similar to the

protein-ligand binding affinity prediction, atomic interactions between specific element types, geometric effects, and hydrophobic effects are characterized. The persistent homology analysis performed in this application is summarized in Table 3.2. The differences between the description vectors arising from Sets 1 and 2, and between those arising from Sets 3 and 4 are also included to account for changes caused by mutation. The 1D-image-like representation in this application thus has a channel size of 45. The pairwise interaction pattern is characterized for 9 element pairs from the element set {C, N, O }. For example, the interactions between the carbon atoms of the mutation site and the nitrogen atoms from the rest of the protein. Such characterization for mutant protein, wild protein, and the difference between these characterizations account for 27 channels. The birth, death, and bar persistence are characterized for Betti-1 and Betti-2 barcodes for all heavy atoms of both the wild type protein and the mutant protein resulting in 12 channels. The difference between the mutant and the wild type, which accounts for 6 channels, is also included. Thus, we have a total of 45 channels. The distance scale interval, $[0, 12]$ Å is divided into bins of length 0.25 Å. An example of the persistent homology barcodes of a mutant and its wild type is given in Fig. 3.1.

Set	Atoms selected	Distance	Complex	Dim.
1	$\{a \in \mathbb{P}^{\mathrm{W}} \backslash \mathbb{M}^{\mathrm{W}} \vert T(a) = e_{\mathrm{P}}\} \cup \{a \in \mathbb{M}^{\mathrm{W}} \vert T(a) = e_{\mathrm{M}}\}, e_{\mathrm{P}}, e_{\mathrm{M}} \in \mathbb{P}^{\mathrm{e}}$	d^{op}	-	0
2	$\{a \in \mathbb{P}^{\mathrm{M}} \backslash \mathbb{M}^{\mathrm{M}} \vert T(a) = e_{\mathrm{P}}\} \cup \{a \in \mathbb{M}^{\mathrm{M}} \vert T(a) = e_{\mathrm{M}}\}, e_{\mathrm{P}}, e_{\mathrm{M}} \in \mathbb{P}^{\mathrm{e}}$	d^{op}	-	0
3	$\{a \in \mathbb{P}^{\mathrm{W}} \vert T(a) \in \mathbb{P}^{\mathrm{e}}\}$	Euclidean	Alpha	1,2
4	$\{a \in \mathbb{P}^{\mathrm{M}} \vert T(a) \in \mathbb{P}^{\mathrm{e}}\}$	Euclidean	Alpha	1,2

Table 3.2: Topological representations for protein mutation problem.
Here $\mathbb{P}^{\mathrm{W}}$, $\mathbb{P}^{\mathrm{M}}$, $\mathbb{M}^{\mathrm{W}}$, and $\mathbb{M}^{\mathrm{M}}$ are sets of atoms of wild type protein, mutant protein, mutation site in the wild type protein, and mutated site in the mutant protein. Here $\mathbb{P}^{\mathrm{e}} = \{\mathrm{C}, \mathrm{N}, \mathrm{O}\}$ and $T(\cdot)$ is the same as defined in Table 3.1. The distance function d^{op} is similar to the one defined in Eq (3.2), while the affiliation function $A(\cdot)$ returns either $\mathbb{M}$ or $\mathbb{P} \backslash \mathbb{M}$.

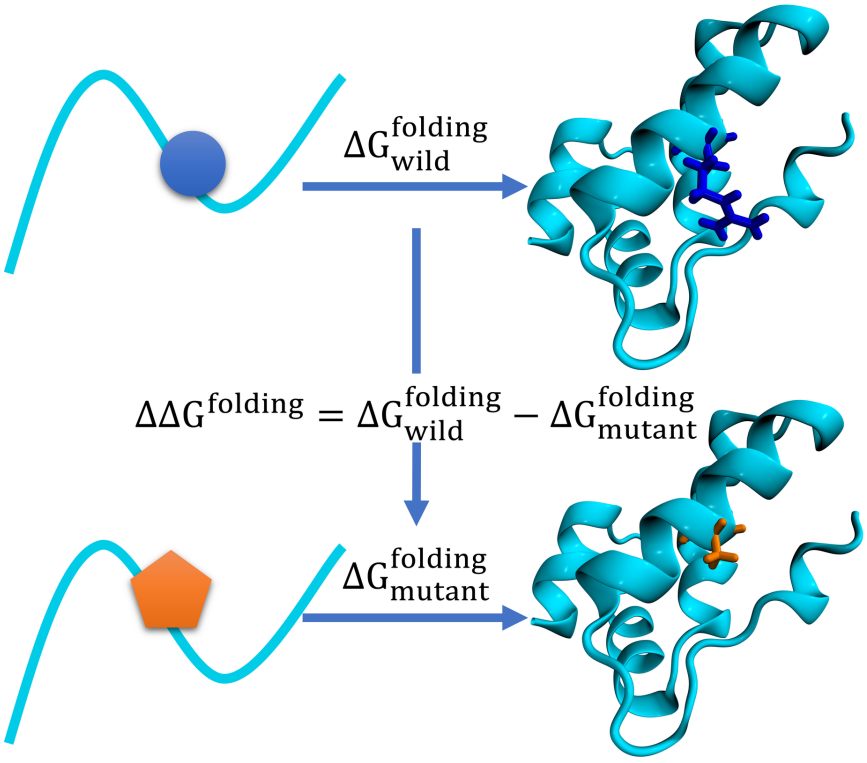

Figure 3.3: Mutation induced protein folding free energy changes.[26]

3.2.3 Neuron for persistence barcode

In addition to structured representations of barcodes, we introduce a neuron construction that directly take persistence barcode as inputs. The basic idea is to treat a barcode as a collection of points in a plane. Let $\mathbb{B} = \{[b_i, d_i)\}_{i \in I}$ be a barcode which can be regarded as a collection of pairs of numbers. With a chosen filtering function, one such neuron takes $\mathbb{B}$ as input and its output is defined to be

$$\mathcal{N}(\mathbb{B}; \Theta, c) = \psi \left(\sum_{i \in I} \phi(b_i, d_i; \Theta) + c \right), \tag{3.3}$$

where ψ is an activation function, ϕ is a filtering function, and Θ is the set of parameters in ϕ that are to be tuned by the training process of the neural network. One natural choice of ϕ is a radial basis function and a specific setup is to use the Gaussian,

$$\mathcal{N}_G(\mathbb{B}; \mu_b, \mu_d, \sigma_b, \sigma_d, c) = \psi \left(\sum_{i \in I} \exp\big(- \frac{(b_i - \mu_b)^2}{2\sigma_b^2} - \frac{(d_i - \mu_d)^2}{2\sigma_d^2} \big) + c \right). \tag{3.4}$$

The basic idea of such construction is that instead of scanning over a predetermined regular region of persistence diagram, we can let the network learn where to examine the pattern of the persistent homology computation result. Intuitively, when a radial basis function is used for ϕ, the output of a neuron characterizes the population of homology generators in that specific area. The location and coverage of each neuron is to be learned by the neural network with given training data. A collection of such neurons forms an input layer of the neural network which directly takes the persistence barcode as the input without formulating it into structured construction. Regular densely connected layers can be stacked upon this input layer to form a multi-layer neural network. The parameters in the filtering

function ϕ carried by the neurons are regarded as part of parameters to be tuned in the entire neural network via training process. This is feasible as long as a proper construction of derivative of ϕ can be derived. There are many possible choices for ϕ. For example, in addition to radial basis functions, a family of Hermite functions can also be used to examine higher order patterns.

3.2.4 Multichannel topological convolutional neural network

The preprocessed multichannel topological image is standardized with mean 0 and standard deviation 1 for use in the convolutional neural network. A convolutional neural network with a few 1D convolution layers, followed by several fully connected layers, is used to extract higher level features from multichannel topological images and to perform regression with the learned features. An illustration of the convolutional neural network structure is shown in Fig. 3.4. A brief review of multichannel topological convolutional neural network concepts is given in the case of 1D-image-like inputs. Convolution operation, optimization method for feedforward neural networks, and dropout out technique which prevents overfitting are discussed. One of the advantages of multichannel topological convolutional deep neural networks is their ability to extract features hierarchically from low level topological representations.

Convolution operation

Consider an $n \times m$ second order tensor $\mathbf{V}$, where n is the number of topological feature pixels and m is number of channels for each pixel. In this approach, n corresponds to the radius filtration dimension of the biomolecular topological analysis and m corresponds the number of representation vectors used which are defined in Eq (3.1). With a predefined window size w, a convolutional filter $\mathbf{F}$ can be represented by a $w \times m$ second order tensor.

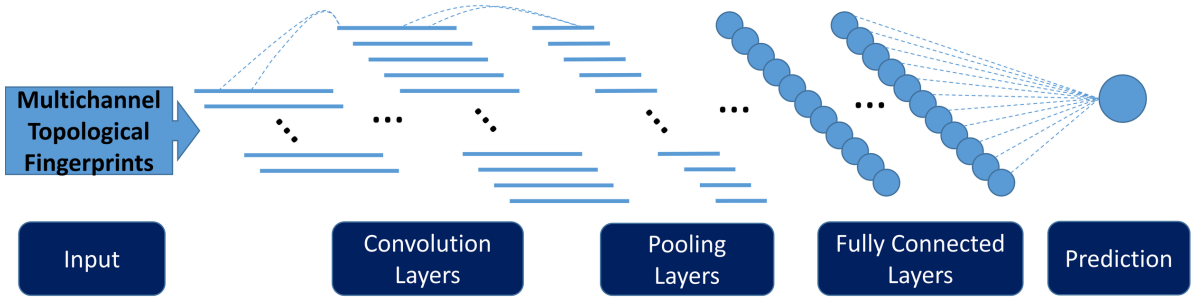

Figure 3.4: An illustration of the 1D convolutional neural network.[26]
The network consists of repeated convolution layers and pooling layers followed by several fully connected layers.

By moving the window of size w along the radius filtration direction of $\mathbf{V}$, a sequence of N_f second order tensors, which are subtensors of V, are obtained and can be concatenated to form an $N_f \times w \times m$ third order tensor $\mathbf{T}$. The filter $\mathbf{F}$ operated on $\mathbf{T}$ results in a first order tensor $\mathbf{T}_{ijk}\mathbf{F}_{jk}$ by tensor contraction. Concatenating the outputs of n_f filters gives an $N_f \times n_f$ second order tensor. Generally speaking, a 1D convolution layer takes an $n \times m$ tensor and outputs an $N_f \times n_f$ tensor.

Dropout

Neural networks with several convolution layers and fully connected layers possess a large number of degrees of freedom which can easily lead to overfitting. The dropout technique is an easy way of preventing network overfitting [173]. During the training process, hidden units are randomly chosen to feed zero values to their connected neighbors in the next layer. Suppose that a percentage of neurons at a certain layer are chosen to be dropped during training. Then, in the testing process, the output of this layer is computed by multiplying a coefficient such as $1 - \lambda$, where λ is the dropout rate, to approximate the average of the network after dropout in each training step.

Bagging (bootstrap aggregating)

In addition to dropout technique which regularizes each individual model, bagging is a technique to combine the output of several models trained separately by averaging to reduce generalization error. This is based on the assumption that models with randomness in the training process likely make different errors on testing data. Generally, bagging trains different models on different subsets of the training set. Specifically, as neural networks have relatively high underlying randomness caused by factors including the random weights initialization and the random mini-batch partition, it can benefit from bagging even if the individual models are trained on the same dataset. In this work, bagging of neural network models trained individually with the same architecture and training dataset is used.

Incorporating non-image-like features

Deep learning architecture also allows the use of non-image-like features together with image or image-like features. In this work, additional auxiliary features, which are important to mutation analysis, are incorporated after the convolution layers as shown in Fig. 3.5. This approach leads to a 9% improvement to mutation prediction of the "S2648" data set.

Multi-task learning

We construct a multi-task multichannel topological convolutional neural network (MM-TCNN) architecture to carry out simultaneous training and prediction. The common topological attributes and underlying physical interactions in features provide a basis for multi-task predictions. Because the deep neural networks are jointly trained from multiple prediction tasks, we expect the networks to generate robust high-level representations from low level TFs for prediction problems. We also expect that the refined representation would lead to prediction models with improved generalized performance. From the proposed deep learning models, we hope to gain insights into how the nonlinear and nonlocal interactions

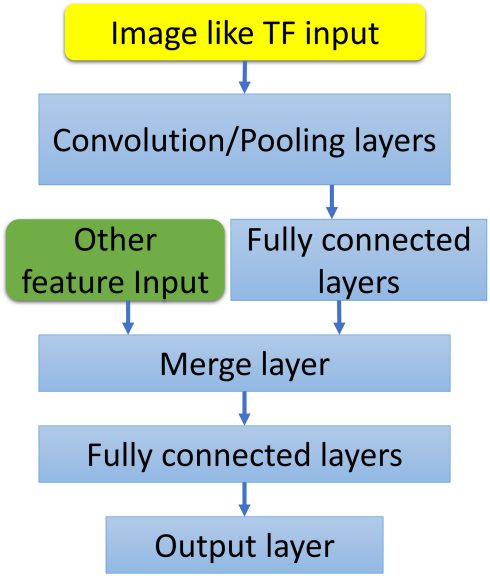

Figure 3.5: The deep learning architecture for the application to globular proteins.[26] The non-image-like features are incorporated in the multichannel topological convolutional deep neural network by merging the features into the network at one of the fully connected layers.

among topological features impact various prediction tasks, which could further lead to better understanding towards the interactions among biomolecular prediction tasks. Finally, tasks with insufficient training data sets will be more likely to benefit from the information collected from tasks with large training sets in a multi-task learning framework.

In the present mutation analysis, there are two data sets. The mutation data of the large data set for globular proteins are more reliable, while those of the small data set for membrane proteins are noisy and less reliable due to the fact that the current technologies for membrane protein mutagenesis experiments are immature. The prediction for membrane proteins benefits from joint learning with the prediction for globular proteins. The coupling of the two predictions through a neural network is shown in Fig. 3.6.

The general objective function to minimize for multi-task learning through neural networks can be decomposed into training loss, similarity penalty for shared layers, and regu-

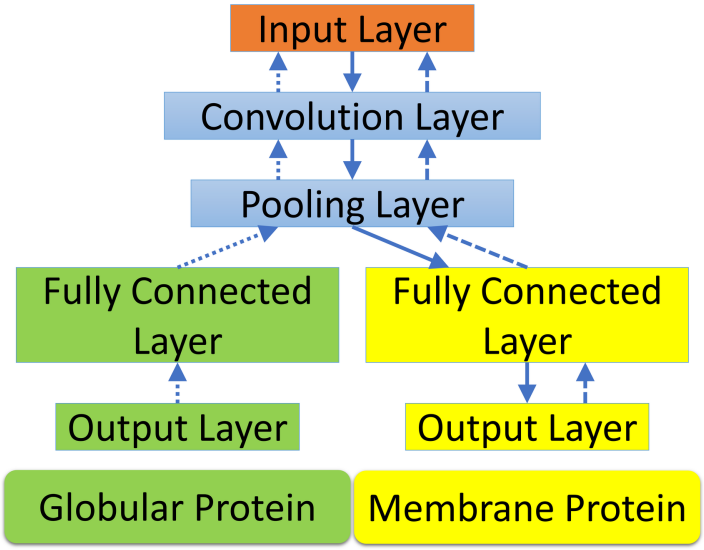

Figure 3.6: The multi-task deep learning architecture for membrane proteins.[26] The globular protein stability change upon mutation prediction is used as an auxiliary task to improve the task of predicting membrane protein stability changes upon mutation. The solid arrows show the path of information passing when the model is applied for predictions. The dotted and dashed arrows mark the paths of backpropagation when the network is trained with globular protein data set and membrane protein data set respectively.

larization term as

$$\begin{aligned}\mathcal{L}(\Theta; X, Y) = &\sum_{j=1}^{N} \mathcal{J}_j(\Theta_{Sj}, \Theta_{Bj}; X_j, Y_j) \\ &+ \mathcal{P}(\Theta_{S1}, \cdots, \Theta_{SN}) \\ &+ \mathcal{R}(\Theta),\end{aligned} \tag{3.5}$$

where Θ is the collection of all parameters to be updated, Θ_{Sj} is the set of parameters for the jth task of the shared layers, Θ_{Bj} is the set of parameters for the jth branch of neurons dedicated for the jth task, and (X_j, Y_j) are training data for the jth task. Here $\mathcal{P}$ is the penalty function which penalizes the difference among N sets of parameters. Finally $\mathcal{R}(\cdot)$ is the regularization term which prevents overfitting and $\mathcal{J}$ is the jth loss function. In this work, we force the shared layers of the two problems to be the same and the regularization of the network is realized using dropout.

Model training and prediction

Due to the complexity of the network for the mutation example with auxiliary features, a brief parameter search is performed using Hyperopt [15] with only 50 trials allowing flexibility in number of neurons, activation function, and weight initialization. In the protein-ligand binding example, only around 10 sets of parameters are selected manually and tested because of the large input size for the problem.

In the protein-ligand binding affinity predictions, we repeatedly train 100 single neural networks individually. To test the performance of bagging of the models, we randomly select 50 trained models from the 100 individually trained networks and output the average value of the outputs from the 50 selected models as the prediction. The performance is then computed for the bagging. This process is repeated 100 times and both median and best results are reported.

In the mutation induced protein stability predictions, we use the same procedure used in the protein-ligand binding prediction, for the "S350" task, where the training and testing split is predefined. In the case of cross validation, 10 sets of 5-fold splits are generated randomly and 20 single models are generated for each split. The average prediction is taken over the 20 models within each split and the median result of the 10 splits is reported. Bagging of only 20 models is performed here because it is not valid to do bagging of predictors on different cross validation splits. The bagging of 50 models will result in 50(individual models)x10(cross validation splits)x5(five folds)=2500 training processes which is too computationally expensive.

Software

Dionysus software [129] with CGAL library [50] is used for persistent homology computation on alpha complex. Javaplex [2] and Dipha [10] software packages are used for persistent

homology computation on Vietoris-Rips complex. The neural networks are realized using Keras [40] wrapper of Theano [177] backend. Various functions from Numpy and Scipy [185] packages are used to process data and evaluate the performance.

3.3 Results

3.3.1 Deep learning prediction of protein-ligand binding affinities

Protein-ligand binding is a fundamental biological process in cells and involves detailed molecular recognition, synergistic protein-ligand interaction, and may involve protein conformational changes. Agonist binding is crucial to receptor functions and typically triggers a physiological response, such as transmitter-mediated signal transduction, hormone and growth factor regulated metabolic pathways, stimulus-initiated gene expression, enzyme production, cell secretion, etc. Understanding protein-ligand interactions has been a fundamental issue in molecular biophysics, structural biology and medicine. A specific task in drug and protein design is to predict protein-ligand binding affinity from given structural information [82] Protein-ligand binding affinity is a measurement of rate of binding which indicates the degree of occupancy of a ligand at the corresponding protein binding site and is affected by several factors including intermolecular interaction strength and solvation effects. The ability to predict protein-ligand binding affinity to a desired accuracy is a prerequisite for the success of many applications in biochemistry such as protein-ligand docking and drug discovery. In general, there are three types of binding affinity predictors (commonly called scoring functions): physics based [147, 207], empirical [211, 183, 63, 190, 212, 131, 182, 94], and knowledge based [114, 105, 6]. In general, physics based scoring functions invoke QM and QM/MM approaches [121, 35] to provide unique insights into the molecular mecha-

nism of protein-ligand interactions. A prevalent view is that binding involves intermolecular forces, such as steric contacts, ionic bonds, hydrogen bonds, hydrophobic effects and van der Waals interactions. Empirical scoring functions work well but require carefully selected data sets and parametrization [183, 63, 190]. However, both physics based scoring functions and empirical scoring functions employ linear superposition principles that are not explicitly designed to deal with exponentially growing and increasingly diverse experimental data sets. Knowledge based scoring functions use modern machine learning techniques, which utilize nonlinear regression and exploit large data sets to uncover underlying patterns within the data sets. Given the current massive and complex data challenges, knowledge based scoring functions outperform other scoring functions. [211].

In this study, the proposed method is tested on the PDBBind 2007 data set [119]. The PDBBind 2007 core set of 195 protein-ligand complexes is used as the test set and the PDBBind 2007 refined set, excluding the PDBBind 2007 core set, is used as the training set with 1105 protein-ligand complexes. A comparison between our TNet-binding predictor (TNet-BP) and other binding affinity predictors is summarized in Table 3.3. TNet-BP outperforms other scoring functions reported by Li *et al* [115] on the task of binding affinity prediction from structures.

TNet-BP is also validated on a larger dataset, PDBBind v2016 refined set of 4057 complexes, where the training set contains 3767 samples which is the refined set minus the core set, and the testing set is the core set with 290 samples. All the model parameters and training procedures are the same as that used for v2007 dataset except that the epoch number is set to 500 instead of 2000 due to the larger data size. The median R_P and RMSE are 0.81 and 1.34 pKd/pKi units, respectively.

Method	R_P	RMSE
TNet-BP	0.826[a]	1.37
RF::VinaElem	0.803	1.42
RF:Vina	0.739	1.61
Cyscore	0.660	1.79
X-Score::HMScore	0.644	1.83
MLR::Vina	0.622	1.87
HYDE2.0::HbondsHydrophobic	0.620	1.89
DrugScore	0.569	1.96
SYBYL::ChemScore	0.555	1.98
AutoDock Vina	0.554	1.99
DS::PLP1	0.545	2.00
GOLD::ASP	0.534	2.02
SYBYL::G-Score	0.492	2.08
DS::LUDI3	0.487	2.09
DS:LigScore2	0.464	2.12
GlideScore-XP	0.457	2.14
DS::PMF	0.445	2.14
GOLD::ChemScore	0.441	2.15
PHOENIX	0.616	2.16
SYBYL::D-Score	0.392	2.19
DS::Jain	0.316	2.24
IMP::RankScore	0.322	2.25
GOLD::GoldScore	0.295	2.29
SYBYL::PMF-Score	0.268	2.29
SYBYL::F-Score	0.216	2.35

Table 3.3: Performance comparisons of TNet-BP and other methods

Comparison of optimal Pearson correlation coefficients R_P and RMSEs (pK_d/pK_i) of various scoring functions for the prediction of protein-ligand binding affinity of the PDBBind 2007 core set. Except for the result of our TNet-BP, all other results are adopted from Li *et al* [115]. [a] Median results (The best $R_P = 0.828$ and best RMSE=1.37 for this method).

3.3.2 Deep learning prediction of protein folding free energy changes upon mutation

Apart from some exceptions, proteins fold into specific three-dimensional structures to provide the structural basis for living organisms. Protein functions, i.e., acting as enzymes, cell signaling mediators, ligand receptors, and structural supports, are typical consequences of a delicate balance between protein structural stability and flexibility. Mutation that changes protein amino acid sequences through non-synonymous single nucleotide substitutions (nsSNPs) plays a fundamental role in selective evolution. Such substitutions may lead to the loss or the modification of certain functions. Mutations are often associated with various human diseases [210, 109]. For example, mutations in proteases and their natural inhibitors result in more than 60 human hereditary diseases [160]. Additionally, mutation can also lead to drug resistance [123]. Artificially designed mutations are used to understand mutation impacts to protein structural stability, flexibility and function, as well as mutagenic diseases, and evolution pathways of organisms [68]. However, mutagenesis experiments are typically costly and time-consuming. Computational prediction of mutation impacts is able to systematically explore protein structural instabilities, functions, disease connections, and organismal evolution pathways [85] and provide an economical, fast, and potentially accurate alternative to mutagenesis experiments. Many computational methods have been developed in the past decade, including support vector machine based approach [27], statistical potentials based approach[55], knowledge-modified MM/PBSA approach [78], Rosetta protocols [100], FoldX (3.0, beta 6.1) [85], SDM [193], DUET [159], PPSC (Prediction of Protein Stability, version 1.0) with the 8 (M8) and 47 (M47) feature sets [205], PROVEAN [39], ELASPIC [16], STRUM [161], and EASE-MM [69].

The proposed method is tested on a data set of 2648 mutation instances of 131 proteins named "S2648" data set [55] in a 5-fold cross validation task over the "S2648" set and a task of prediction of the "S350" set which is a subset of "S2648" set. The "S2648" set, excluding the "S350" subset, is used as the training set in the prediction of the "S350" set. All thermodynamic data are obtained from the ProTherm database [12]. A comparison of the performance of various methods is summarized in Table 3.4. Among them, STRUM [161] is based on structural, evolutionary and sequence information and results in excellent performance. We therefore have constructed two topology based neural network mutation predictors (TNet-MPs). TNet-MP-1 is solely based on topological information while TNet-MP-2 is aided by auxiliary features characterizing electrostatics, evolutionary, and sequence information, which is merged into the convolutional neural network at one of the fully connected layers. TNet-MP-2 is able to significantly improve our original topological prediction, indicating the importance of the aforementioned auxiliary information to mutation prediction.

Method	S350			S2648		
	n^d	R_P	RMSE	n^d	R_P^e	RMSE^f
TNet-MP-2	350	0.81	0.94	2648	0.77	0.94
STRUM^b	350	0.79	0.98	2647	0.77	0.94
TNet-MP-1	350	0.74	1.07	2648	0.72	1.02
mCSM^{b,c}	350	0.73	1.08	2643	0.69	1.07
INPS^{b,c}	350	0.68	1.25	2648	0.56	1.26
PoPMuSiC 2.0^b	350	0.67	1.16	2647	0.61	1.17
PoPMuSiC 1.0^a	350	0.62	1.23	-	-	-
I-Mutant 3.0^b	338	0.53	1.35	2636	0.60	1.19
Dmutant^a	350	0.48	1.38	-	-	-
Automute^a	315	0.46	1.42	-	-	-
CUPSAT^a	346	0.37	1.46	-	-	-
Eris^a	334	0.35	1.49	-	-	-
I-Mutant 2.0^a	346	0.29	1.50	-	-	-

Table 3.4: Performance comparisons of TNet-MP and other methods.
Comparison of Pearson correlation coefficients (R_P) and RMSEs (kcal/mol) of various methods on the prediction task of the "S350" set and 5-fold cross validation of the "S2648". TNet-MP-1 is our multichannel topological convolutional neural network model that solely utilizes topological information. TNet-MP-2 is our model that complements TNet-MP-1 with auxiliary features. ^{a} Data directly obtained from Worth *et al*[193]. ^{b} Data obtained from Quan *et al* [161]. ^{c} The results reported in the publications are listed in the table. According to Ref. [161], the data from the online server has R_p (RMSE) of 0.59 (1.28) and 0.70 (1.13) for INPS and mCSM respectively in the task of S350 set. ^{d} Number of samples successfully processed.

3.3.3 Multi-task deep learning prediction of membrane protein mutation impacts

Multi-task learning offers an efficient way to improve the predictions associated with small data sets by taking the advantage of other larger data sets [214]. Although a large amount of thermodynamic data is available for globular protein mutations, the mutation data set for membrane proteins is relatively small, between 200 and 300 proteins [108]. The small size of membrane protein mutation data limits the success of data driven approaches, such as ensemble of trees. While the popular multi-task learning framework built on linear regression with regularization techniques lacks the ability to extract the relationship between very low

level descriptors and the target quantity. A neural network with a hierarchical structure provides a promising option for such problems. We add the prediction of globular protein stability changes upon mutation as an auxiliary task for the prediction of membrane protein stability changes upon mutation. In the designed network architecture, two tasks share convolution layers and the network splits into two branches with fully connected layers for the two tasks. Intuitively, the task of globular protein mutation predictions help to extract higher level features from low level topological representations. Thus, the branch for membrane protein mutation predictions learns the feature-target relationship from the learned high level features.

The proposed method is tested on a set of 223 mutation instances of membrane proteins covering 7 protein families named "M223" data set [108] with 5-fold cross validation. A comparison with other methods is shown in Table 3.5. TNet-MMP-1 employes multichannel topological convolutional neural networks with topological features from the "M223" data set, while TNet-MMP-2 is a multi-task multichannel topological convolutional neural network (MM-TCNN) architecture. Unlike TNet-MP-2, both TNet-MMP-1 and TNet-MMP-2 do not use auxiliary features. Our goal is to test the performance of the multi-task architecture on the improvement of high level feature extraction from low level features. Pearson correlation coefficient of membrane protein mutation prediction is improved by 9.6%, i.e., from 0.52 to 0.57 by the multi-task algorithm that trains and predicts the present "M223" data set with the "S2648" date set. As noted by Kroncke *et al*, there is no reliable methods for the prediction of membrane protein mutation impacts at the present [108]. Our TNet results, though still not practically useful, are the best among the methods tested on this problem.

Method	R_P	RMSE
TNet-MMP-2[d]	0.57	1.09
TNet-MMP-1[c]	0.52	1.15
Rosetta-MP	0.31	-
Rosetta (High)[a]	0.28	-
FoldX	0.26	2.56
PROVEAN	0.26	4.23
Rosetta-MPddG	0.19	-
Rosetta (low)[b]	0.18	-
SDM	0.09	2.40

Table 3.5: Performance comparisons of TNet-MMP and other methods.
Comparison of Pearson correlation coefficients (R_P) and RMSEs (kcal/mol) on 5-fold cross validation for the "M223" data set for various methods. Except for the present results for TNet-MMP-1 and TNet-MMP-2, all other results are adopted from Kroncke *et al*[108]. The results of Rosetta methods are obtained from Fig. S1 of Ref. [108] where RMSE is not given. The results of other methods are obtained from Table S1 of Ref. [108]. Many less competitive results of the machine learning based methods reported in Ref. [108] are not listed since these servers were not trained on the membrane protein data set. Among the methods listed, only Rosetta methods have terms describing the membrane protein system. [a] High resolution. [b] Low resolution. [c] The multichannel topological convolutional neural network architecture with topological features from "S223" data set. [d] The multi-task multichannel topological convolutional neural network (MM-TCNN) architecture trained with an auxiliary task of globular protein prediction using the "S2648" data set.

3.4 Discussion and conclusion

The adoption of convolutional neural network concepts in this work is motivated by the underlying spatial relationship along the distance scale (filtration) dimension. Properties that reside in different distance scales are heterogeneous so unlike images or videos, there is no obvious transferable property of the convolution filters along the convolution dimension in the proposed method. To take this into consideration, the convolution layers are substituted with "locally connected layers", where the local connection properties are conserved whilst the filters applied to different distance scales are allowed to be different. The RMSE is in kcal/mol for the mutation problems and pKd/pKi units for the protein-ligand binding problem. The performance in R_P (RMSE) significantly decreases from 0.81 (0.94) to 0.77 (1.02) for the task of "S350" set prediction in the mutation impact example. This shows

that the construction of lower level features in the lower sparse layers benefits from sharing filters along the distance scale and indicates the existence of some common rules for feature extractions at different distance scales.

Intuitively, the dimension 0 inputs describe pairwise atomic interactions, which clearly contribute to the prediction of the target properties. In contrast, dimension 1 and dimension 2 topological features characterize the hydrophobic network and geometric rings and voids. To understand to what extent the higher topological dimensions help the characterization of biomolecules, we separate the dimension 0 inputs from higher dimensional inputs in the prediction of "S350" set in the mutation impact on protein stability example and in the protein-ligand binding affinity prediction for v2007 set example. To compare the performance of different sets of features, 50 single models are trained for each feature set. Twenty of the 50 trained models are randomly chosen and bagged, and this procedure is repeated 100 times with the median results reported. The individual performances measured by R_P (RMSE) for dimension 0 features are 0.73 (1.09) and 0.82 (1.40), respectively for the mutation and binding predictions. For dimensions 1 and 2 features, R_P (RMSE) are 0.66 (1.21) and 0.78 (1.54), respectively for the mutation and binding predictions. The combination of all dimension features results in better R_P (RMSE) of 0.74 (1.08) and 0.83 (1.37), respectively for the mutation and binding predictions, showing that two sets of features both contribute to predictions. The alpha complex is used for geometric characterization and therefore is in $\mathbb{R}^3$ with Betti number up to dimension 2. It is possible that the higher dimensional Betti numbers in a more abstract setup such as Vietoris-Rips complex for the characterization of an interaction network will enrich the representation and deliver improved results.

Another popular class of machine learning methods is the ensemble of trees methods. Many modern methods for biomolecular property prediction are based on random forest

(RF) and gradient boosting trees (GBTs). The ensemble of decision trees has the capability of learning complicated functions, but GBTs learn to partition the feature space based on the training data which means that they do not have the ability to appropriately extrapolate the learned function to broader situations than the provided training data. Additionally, it is generally the case that data samples are unevenly distributed. It has been observed that in many applications, where among the dataset, there are just a handful of samples with large absolute value for the target property, methods of ensembles of trees tend to overestimate (underestimate) the border cases with very negative (positive) target values. The neural network, due to its different ways of learning the underlying function, seems to be able to deliver better results for the border cases. Therefore, similar to the idea of bagging, methods of ensembles of trees and neural network based methods may result in different error characteristics for different samples and can potentially improve the predictive power by correcting each others' error when the results from different models are averaged. In the example of prediction of the "S350" set, we obtained performance of 0.82 (0.92) for R_P (RMSE) in our other work using handcrafted features with gradient boosting trees [23]. When the results are averaged for the two methods, the performance is improved to 0.83 (0.89) which is better than both individual methods. Similar improvement is observed for the protein-ligand binding example with v2007 set. Our method based on handcrafted features and gradient boosting trees with performance 0.82 (1.40) [24] and the method presented in this work with performance 0.83 (1.37) can achieve improved performance of 0.84 (1.35) when the two results are combined by averaging. An intuitive illustration is shown in Fig. 3.7. It can be seen from the plot that the neural network based method presented in this work performs better than the GBT based method for samples with high $\Delta\Delta G$ or with low $\Delta\Delta G$. The slope of linear fitting of the predicted values to the experimental data is

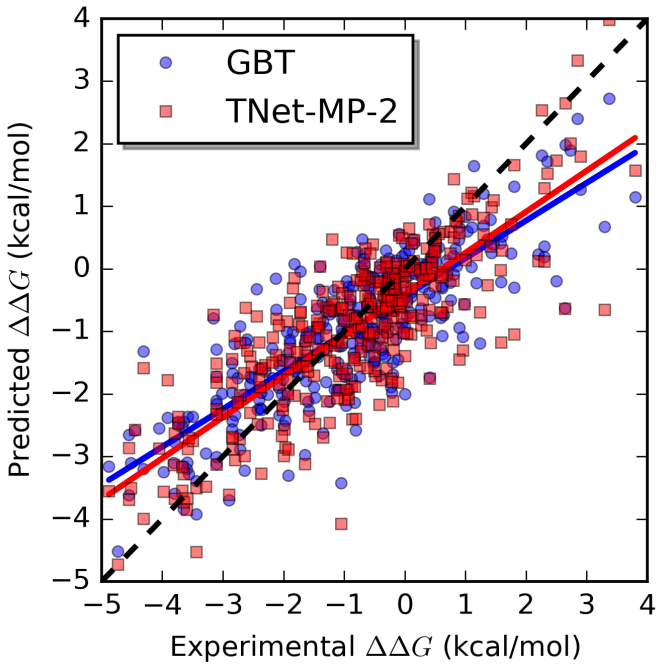

Figure 3.7: A comparison of behaviors of the GBT based method and the neural network based method.[26]
The plot is for the prediction task of the S350 dataset. The linear fit for GBT prediction [23]is $y = 0.603x - 0.435$ and for TNet-MP-2, $y = 0.657x - 0.422$.

0.66 for the neural network based method and 0.60 for the GBT based method which also illustrates that the neural network based method handles border cases better. The observed improvement is marginal since it is mainly on a small portion of the samples.

In conclusion, the approach introduced in this work utilizes element-specific persistent homology to efficiently characterize 3D biomolecular structures in terms of multichannel topological invariants. Convolutional neural network facilitates the automatic feature extraction from multichannel topological invariant inputs. The flexible and hierarchical structure of neural network allows seamless combination of automatically extracted features and handcrafted features. It also makes it easy to implement multi-task learning by combining related tasks to a desired level of model sharing by tuning the layer of model branching. The proposed topology based neural network (TopologyNet) methods have been shown to out-

perform other existing methods in protein-ligand binding affinity predictions and mutation induced protein stability change predictions. The proposed methods can be easily extended to other applications in the structural prediction of biomolecular properties. They have the potential to further benefit from the fast accumulating biomolecular data. The combination of the proposed methods and existing RF and GBT based methods is expected to deliver improved results.

Chapter 4

Persistent cohomology for data with heterogeneous dimensions

4.1 Introduction

With the advancements in sensor hardware, data collection software, data organization and storage frameworks, various datasets are expanding in an unprecedented speed where a large part of the newly accumulated data is high-dimensional, highly complex, diverse, and often noisy. The rapid growth of datasets demands robust and automatic data analysis tools. While many widely used data analysis methods make assumptions of data complexity and/or the underlying dimensionality and some other methods often require knowledge from domain experts, an emerging family of data analysis methods called topological data analysis [28] (TDA) makes minimal assumptions of data. TDA characterizes the shapes of data in various dimensions and scans over a wide range of scales and is often robust against noise.

Given a point cloud dataset, it can be regarded as embedded in the Euclidean space which allows the usage of radius filtration associated with alpha complex [60]. A more general distance filtration associated with a Vietoris-Rips complex [87] or a Čech complex can be used to allow a predefined distance function suitable for specific applications [198, 200]. It is also possible to use a more flexible construction by directly assigning filtration values to

simplices in a complex which is considered as the final structure at the end of the filtration. In many applications, persistent homology is used to analyze the topological structures of datasets with generalized but homogenous information. For example, once the genetic distance between genes is defined by the number of mutations, persistent homology can be used to analyze the topological property of a gene evolution dataset. When the information of a dataset is heterogeneous, i.e., it involves multicomponent information, for which special treatments are needed. For example, vineyards [45, 133] is used to study spatiotemporal data. Additionally, element specific persistent homology is introduced to deal with molecular datasets with chemical, physical and biological information [23, 24, 26, 20].

In Chapter 3, we developed a deep learning model for biomolecular predictive modeling based on topological representations of mainly the biomolecular geometries. When persistent homology is applied to complex molecular structures, in addition to the point cloud in the Euclidean space representing the coordinates of atoms, there are additional physical and chemical information such as element types, atomic partial charges, Coulomb and van der Waals interactions between atoms, and hydrophobic interactions among carbon atoms. Another general situation is that the data has multiple dimensions with heterogeneous meanings and persistent homology analysis is done on certain dimensions while the information of other dimensions are also to be reflected in the topological analysis. Therefore, there is a need to incorporate multicomponent heterogeneous information into topological representations. Although one can resort to the tight representative cycles of homology generators [144], we prefer the cohomology framework because it is flexible and it is natural to view cochains as assigning weights on the simplicial complex which provides more quantitative representations. We consider cohomology theory with a graph Laplacian or a Laplacian defined on simplicial complexes to localize and smooth the representatives of (co)homology

generators in the data and describe the additional information in the form of cochains (functions on chains) by computing inner product of these cochains and the smoothed (persistent) cohomology representatives.

Cohomology provides a richer algebraic structure for a topological space. The cohomology construction used in this work dualizes homology by building cochains as functions on chains in homology theory. Cohomology theory has been applied in both mathematics and the field of data analysis. One well known cohomology theory is the de Rham cohomology which studies the topological features of smooth manifolds using differential forms. The de Rham cohomology has led to further theoretical developments such as the Hodge theory. Recently, a discrete exterior calculus framework has been established [90] where manifolds are approximated by mesh-like simplicial complexes and the discrete counterparts of the continuous concepts such as differential forms are defined thereafter. This framework has wide applications, for example, the harmonic component of the discrete Hodge decomposition has been used in sensor network coverage problem to localize holes in a sensor network [13]. Cohomology theory has also been applied in the field of persistent homology. A 1-dimensional cohomology was used to assign circular values to the input data associated to a homology generator [54] which further led to applications in several fields including the analysis of neural data [167] and the study of periodic motion [181]. Persistent cohomology in higher dimensions has been used to produce coordinate representations which reduces dimensionality while retaining the topological property of data [155]. Weighted (co)homology and weighted Laplacian were introduced with biological applications [194]. Computationally, the duality between homology and cohomology [53] has set the basis for constructing more efficient algorithms that utilize cohomology to compute persistent homology. Several code implementations, such as Dionysus [129] and Ripser [8], speed up the persistent homology

computation by taking the advantage of this property.

In this work, we seek a formulation that can utilize a function fully or partially defined on a simplicial complex constructed from the input data at locations associated to homology generators. To this end, we need a representation that can locate homology generators. When manifold-like simplicial complexes are available, we can look for harmonic (in the sense of Laplace-de Rham operator) cohomologous cocycles under the framework of discrete exterior calculus [89]. A discrete version of the Hodge-de Rham theorem guarantees the uniqueness of the harmonic cocycle if certain conditions are satisfied [89]. However, this method requires the proper construction of the Hodge star operator which usually relies on a well-defined dual complex while in general applications, this is not always feasible. For example, when a user-defined distance matrix is used with the Rips complex, the distance may even not satisfy triangle inequality. Therefore, we relax our requirement on the accuracy of geometric localization and represent the set of simplices of a certain dimension in a simplicial complex as a graph where the simplices are represented by graph nodes and their adjacency is treated as graph edges. We can also define a Laplacian on simplicial complexes by first introducing an inner product of cochains and then constructing an adjoint of the coboundary operator. Then, the smoothness of a cocycle can be measured by a Laplacian. Specifically, given a representative cocycle of a homology generator, we look for a cohomologous cocycle that minimizes the norm of the output under the Laplacian. We can then consider such smoothed cocycles which distribute smoothly around the holes of certain dimensions as measures on simplicial complexes and describe the input functions defined on the simplicial complexes by integrating with respect to these measures. The present formulation also utilizes a filtration process to assign a function over the filtration interval associated to each bar in the barcode representation to result in an enriched barcode representation of persistent cohomology. A

modified Wasserstein distance is defined and implemented subsequently to facilitate the comparison of these enriched barcodes generated from data.

In the rest of this chapter, basic background of cohomology is given and the proposed method is described in detail in Section 4.2. In Section 4.3, we illustrate the proposed method by simple examples, example datasets, and the characterization of molecules. We also demonstrate the utility of the proposed persistent cohomology by the prediction of protein-ligand binding affinities from large datasets.

4.2 Methods

We refer readers to Section 2.1 for the basics and definition of persistent homology.

4.2.1 Cohomology

Like homology, cohomology is also a sequence of abelian groups associated to topological space X and is defined from a cochain complex, which is a function on the chain group in the homology theory. Specifically, a k-cochain is a function $\alpha : X^k \to R$ where R is a commutative ring. The set of all k-cochains following the addition in R is called the kth cochain group denoted $C^k(X, R)$. The coboundary operator $d_k : C^k(X, R) \to C^{k+1}(X, R)$ maps a cochain to a cochain one dimension higher and is the counterpart of boundary operators for chains, namely

$$d_k(\alpha)([v_0, \cdots, v_k]) = \sum_{i=0}^{k} (-1)^i \alpha([v_0, \cdots, \widehat{v_i}, \cdots, v_k]),$$

for a k-cochain α. It should be noted that in the matrix representation of the two operators, d_k and ∂_{k+1} are transpose to each other. When there is no ambiguity, we simply refer to d_k using d. A k-cochain is called a coboundary if it is in the image of d_{k-1}. A k-cochain is called a cocycle if its image under d_k is 0. The coboundary operators have the property that $d_k \circ d_{k-1} = 0$ following that $d_k \circ d_{k-1} = \partial_{k+1}^T \circ \partial_k^T = (\partial_k \circ \partial_{k+1})^T$. The kth cohomology group is defined to be the quotient group $H^k(X, R) = \mathrm{Ker}(d_k)/\mathrm{Im}(d_{k-1})$. Two cocycles are called cohomologous if they defer by a coboundary.

In practice, some field is used instead of a ring due to the computation of persistent (co)homology. In this work, we consider finite fields $\mathbb{Z}_p$ with some prime p when computing cohomology or persistent cohomology.

Given a filtration of a simplicial complex, similar to persistent homology, the persistent cohomology can be derived with the following relationship

$$H^k(X(x_0), \mathbb{Z}_p) \leftarrow H^k(X(x_1), \mathbb{Z}_p) \leftarrow \cdots \leftarrow H^k(X(x_l), \mathbb{Z}_p).$$

The universal coefficient theorem for cohomology (Theorem 3.2 in [86]) implies that there is a natural isomorphism $H^k(X, \mathbb{Z}_p) \equiv \mathrm{Hom}_{\mathbb{Z}_p}(H_k(X, \mathbb{Z}_p), \mathbb{Z}_p)$ so that the cohomology group can be considered as the dual space of the homology group. This property further implies that $\mathrm{rank}(H^k(X, \mathbb{Z}_p)) = \mathrm{rank}(H_k(X, \mathbb{Z}_p))$ and thus persistent homology and persistent cohomology have identical barcodes [53].

4.2.2 Smoothed cocycle

Some representative cocycles in persistent cohomology may not reflect the overall location and structure associated with their cohomology generators. To better embed the additional

information in the data into cohomology generators, we look for a smoothed representative cocycle in each cohomology class. The smoothness of functions can usually be measured by a Laplacian. We construct smoothed representative cocycles with a Laplacian in this section.

Laplacian on simplicial complex

A Laplacian for cochains can be defined by first defining an inner product and using the induced adjoint operator. Assuming the case of real number, for $\alpha_1, \alpha_2 \in C^k(X, \mathbb{R})$, the inner product can be defined as

$$< \alpha_1, \alpha_2 >_k = \sum_{\sigma \in X^k} \alpha_1(\sigma)\alpha_2(\sigma). \tag{4.1}$$

Then, the adjoint $d_k^* : C^{k+1}(X, \mathbb{R}) \to C^k(X, \mathbb{R})$ of the operator d_k with respect to this inner product can be defined by

$$< d_k\alpha, \beta >_{k+1} = < \alpha, d_k^*\beta >_k, \text{ for } \alpha \in C^k(X, \mathbb{R}), \beta \in C^{k+1}(X, \mathbb{R}). \tag{4.2}$$

Weights reflecting size of simplices can be used to reflect the geometry by defining a weighted inner product,

$$< \alpha_1, \alpha_2 >_k^w = \sum_{\sigma \in X^k} s_\sigma \alpha_1(\sigma)\alpha_2(\sigma), \tag{4.3}$$

where s_σ is the size of σ such as area or volume if σ is a 2- or 3-simplex. Then, a Laplacian on X^k can be defined by

$$\mathcal{L}_{sc} = d_k^* d_k + d_{k-1} d_{k-1}^*. \tag{4.4}$$

An inner product based on Wedge product can also be constructed if a manifold like simplicial complex is given.

Weighted graph Laplacian

We can also represent X^k as a graph where the nodes are simplices and the edges describe adjacency. Consider a graph associated to X^k where each k-simplex is represented by a node and there is an edge if two k-simplices have nonempty intersection. Note that this is a simple graph and we define a weight matrix $W = (w_{ij})$ to be

$$w_{ij} = \begin{cases} v(\sigma_i)v(\sigma_j),\ \sigma_i \cap \sigma_j \neq \emptyset, \\ 0,\ \text{otherwise}, \end{cases} \tag{4.5}$$

where $v(\sigma)$ is the size of σ, for example, the area of 2-simplices and the volume of 3-simplices. The size of a 0-simplex is defined to be 1. Then, the W-weighted graph Laplacian [42] $\mathcal{L}_W$ is defined as

$$\mathcal{L}_{W\,i,j} = \begin{cases} 1 - w_{ij}/w_i,\ \text{if}\, i = j,\ \text{and}\, w_i \neq 0, \\ -w_{ij}/\sqrt{w_i w_j},\ \text{if}\, \sigma_i \cap \sigma_j \neq \emptyset,\ \text{and}\, i \neq j, \\ 0,\ \text{otherwise}, \end{cases} \tag{4.6}$$

where $w_i = \sum_j w_{ji}$. A k-cochain α can be represented by a column vector given the basis in Eq. 4.7. The matrix $\mathcal{L}_W$ measures the difference between the value of the cochain on a simplex and the values on the neighbors of this simplex. A large penalty is given to prevent rapid changes through smaller simplices.

4.2.3 Enriched persistent barcode

We describe the work-flow in this section. Given a simplicial complex X of dimension n, and a function $f : X^k \to \mathbb{R}$ with $0 \leq k \leq n$, we seek a method to embed the information of f

on the persistence barcodes obtained with a chosen filtration of X. In other words, we seek a representation of f on cohomology generators. To this end, smoothed representations are first computed for cohomology generators. One of such smoothed representations induces a measure on the simplicial complex which allows us to integrate f on X. We describe the protocol of our approach below.

Dimension greater than 0

Consider a filtration of X, $\emptyset = X(x_0) \subseteq X(x_1) \subseteq \cdots \subseteq X(x_n) = X$ and an associated persistent cohomology with a prime p other than 2. Let ω be a representative cocycle for a persistence interval $[x_i, x_j)$ of dimension $k > 0$. The cocycle ω is first lifted to a cocycle $\hat{\omega}$ with integer coefficients satisfying that $\omega(\sigma) \equiv \hat{\omega}(\sigma) \,(\mathrm{mod}\, p)$ and $\hat{\omega}(\sigma) \in \{i \in \mathbb{Z} : -(p-1)/2 \leq i \leq (p-1)/2\}$ for all $\sigma \in X^k$. This is almost always possible [54]. Now that $\hat{\omega}$ is an interger cocycle and thus also a real cocycle. With a basis for k-cochains $\{\alpha_{\sigma_i}\}_i$ where

$$\alpha_{\sigma_i}(\sigma) = \begin{cases} 1, \, \sigma = \sigma_i \\ \\ 0, \, \text{otherwise} \end{cases} \tag{4.7}$$

Given a Laplacian on cochains $\mathcal{L}$ to measure smoothness, a smooth cocycle $\bar{\omega}$ can be obtained by solving a least square problem,

$$\bar{\alpha} = \underset{\alpha \in C^{k-1}(X, \mathbb{R})}{\arg\min} \|\mathcal{L}(\hat{\omega} + d\alpha)\|_2^2, \tag{4.8}$$

and letting $\bar{\omega} = \hat{\omega} + d\bar{\alpha}$. This smoothed cocycle $\bar{\omega}$ induces a measure μ on X^k by setting

$$\mu(\sigma) = |\bar{\omega}(\sigma)|. \tag{4.9}$$

To obtain a sequence of such smoothed real k-cocycles for the cohomology generator along a persistence interval, we restrict the representative integer cocycle $\hat{\omega}$ to subcomplexes of X and repeat the smoothing computation. Consider the integer k-cocycle $\hat{\omega}|_{X(x)}$ at filtration value x. The corresponding smoothed real k-cocycle $\bar{\omega}_x$ can be obtained by running the optimization problem for $\hat{\omega}|_{X(x)}$ as Eq. (4.8) on $C^{k-1}(X(x), \mathbb{R})$ and it induces a measure μ_x on $X^k(x)$ as described in Eq. (4.9). It suffices to compute for all different filtration values in $[x_i, x_j)$ because we have a finite filtration which gives $\{\mu_{x_\ell}\}_{\ell=i}^{j-1}$.

A function of filtration values f^* can be defined for each persistent interval $[x_i, x_j)$ as

$$f^*(x) = \int_{X^k(x)} f d\mu_x \Big/ \int_{X^k(x)} d\mu_x \tag{4.10}$$

for $x \in [x_i, x_j)$. We call each of the collection of persistent intervals being associated with one such function f^* an enriched persistent barcode.

Dimension 0

In the case of dimension 0, persistent homology tracks the appearance and merging of connected components. It is convenient to assign a smooth 0-cocycle to a persistent interval by assigning 1 to the nodes in the connected component associated with the interval right before the generator is killed due to merging with another connected component. This is implemented with a union-find algorithm.

4.2.4 Preprocessing of the input function

When given the original input function associated with the input data, we first need to generate a cochain of the dimension of interest out of this input function. The procedures in several situations are discussed in the rest of this section.

Case 1

When given a function $f_0 : X^{k0} \to \mathbb{R}$, and we are interested in its behavior associated with a k-dimensional homology where $k_0 \neq k$. We need to interpolate or extrapolate f_0 to a function $f : X^k \to \mathbb{R}$. We assume that f_0 is unoriented, i.e. $f_0(\sigma) = f_0(-\sigma)$. A simple way is to take unweighted averages,

$$f_a(\sigma) = \frac{1}{n_\sigma} \sum_{i=1}^{n_\sigma} f_0(\sigma_i'), \tag{4.11}$$

where each σ_i' is a k_0-simplex satisfying that $\sigma_i' < \sigma$ if $k > k_0$ and $\sigma_i' > \sigma$ if $k < k_0$ and n_σ is the total number of such k_0-simplices. A weighted version based on geometry can be defined as

$$f_w(\sigma) = \sum_{i=1}^{n_\sigma} w_i f_0(\sigma_i') \Big/ \sum_{i=1}^{n_\sigma} w_i, \tag{4.12}$$

where w_i is the reciprocal of the distance between the barycenters of σ and σ_i'.

An example of this situation is the pairwise interaction strengths between atoms of a molecule which are naturally defined on edges connecting the vertices representing the atoms. Another example is the atomic partial charges defined on the vertices representing the atoms in a molecule or a molecular complex.

Case 2

When given a function $f_0 : \mathbb{R}^n \to \mathbb{R}$ with $n >= k$ and a geometric simplicial complex, we can integrate it on every k-simplex in X to obtain a function $f_i : X^k \to \mathbb{R}$. For simplicity, we require f_0 to be bounded. Then, f_i is defined as

$$f_i(\sigma) = \int_\sigma f_0 d\sigma \Big/ \int_\sigma d\sigma, \tag{4.13}$$

for a k-simplex σ and $\int_\sigma d\sigma$ computes the k-dimensional volume of σ. In many cases, f_0 is given as results of numerical simulations which is often defined on grid points. Then, the integrals can be computed by some chosen quadrature formula and interpolating f_0 to the collocation points.

4.2.5 Modified Wasserstein distance

An enriched bar can be represented by three elements, birth value b, death value d, and function f^* constructed by Eq. (4.10). Given two enriched barcodes of the same dimension represented by $B = \{\{b_i, d_i, f_i^*\}\}_{i\in I}$ and $B' = \{\{b'_j, d'_j, f'^*_j\}\}_{j\in J}$, we would like to compute their distance analogous to Wasserstein distance. We first define two pairwise distances, i.e., Δ_b that measures the distance between two persistence bars

$$\Delta_b\left([b,d),[b',d')\right) = \max\{|b-b'|,|d-d'|\} \tag{4.14}$$

and Δ_f that measures the distance between f^* and f'^*

$$\Delta_f(f^*,f'^*) = \left|\frac{1}{d-b}\int_b^d f^*(x)dx - \frac{1}{d'-b'}\int_{b'}^{d'} f'^*(x)dx\right|. \tag{4.15}$$

In practice, it sometimes is too costly to compute the output values of f^* for all possible filtration values, and only a subset of possible filtration values is selected, such as only the middle value of a bar. In this case, we use middle Riemann sum to approximate the integration in Eq. 4.15. For a bijection $\theta : \bar{I} \to \bar{J}$ where $\bar{I}$ and $\bar{J}$ are subsets of I and J, the

associated penalties are defined as

$$\begin{aligned} P_b(\theta; q, B, B') = &\sum_{i\in\bar{I}} \left(\Delta_b \left([b_i, d_i), [b'_{\theta(i)}, d'_{\theta(i)})\right)\right)^q \\ &+ \sum_{i\in I\setminus\bar{I}} \left(\Delta_b \left([b_i, d_i), [(b_i + d_i)/2, (b_i + d_i)/2)\right)\right)^q \\ &+ \sum_{i\in J\setminus\bar{J}} \left(\Delta_b \left([b'_i, d'_i), [(b'_i + d'_i)/2, (b'_i + d'_i)/2)\right)\right)^q \end{aligned} \tag{4.16}$$

and

$$\begin{aligned} P_f(\theta; q, B, B') = &\sum_{i\in\bar{I}} \left(\Delta'_f \left(f_i^*, f'^*_{\theta(i)}\right)\right)^q \\ &+ \sum_{i\in I\setminus\bar{I}} \left(\Delta'_f(f_i^*, 0)\right)^q \\ &+ \sum_{i\in J\setminus\bar{J}} \left(\Delta'_f(f'^*_i, 0)\right)^q. \end{aligned} \tag{4.17}$$

The qth modified Wasserstein distance is defined as

$$W^{q,\gamma}(B, B') = \inf_{\theta\in\Theta} \left(\gamma P_b(\theta; q, B, B') + (1-\gamma) P_f(\theta; q, B, B')\right)^{\frac{1}{q}}, \tag{4.18}$$

where γ is a weight parameter and we denote the minimizer by $\theta^{q,\gamma}$. Similar to receiver operating characteristic curve, instead of fixing γ we let it change from 0 to 1 which results in a function $\mathcal{W}^q : [0, 1] \to \mathbb{R}^2$ defined as

$$\mathcal{W}^q(\gamma) = [P_b(\theta^{q,\gamma}; q, B, B')^{\frac{1}{q}}, P_f(\theta^{q,\gamma}; q, B, B')^{\frac{1}{q}}], \tag{4.19}$$

and we call it a Wasserstein characteristic curve.

The optimization problem can be considered as an assignment problem and solved by

Hungarian algorithm. Given two enriched barcodes $B = \{\{b_i, d_i, f_i^*\}\}_{i=1}^m$ and $B' = \{\{b'_j, d'_j, f'^*_j\}\}_{j=1}^n$, we first construct pseudo barcode for each of them to account for the situation where a bar is not paired with another. The pseudo barcodes are $B_{B'} = \{\{(b'_j + d'_j)/2, (b'_j + d'_j)/2, 0\}\}_{j=1}^n$ and $B'_B = \{\{(b_i + d_i)/2, (b_i + d_i)/2, 0\}\}_{i=1}^m$. Then the assignment problem between $B \cup B_{B'}$ and $B' \cup B'_B$ is solved with the cost $(\gamma P_b + (1-\gamma)P_f)^{\frac{1}{q}}$. The linear_sum_assignment tool under optimize module of SciPy package [97] is used.

4.3 Examples and results

4.3.1 A minimalist example

Consider a simplicial complex X with four vertices and four edges with unit length that forms a square as shown in Figure 4.1. The 1-cochain $\hat{\omega} = [1, 0, 0, 0]^T$ is a real cocycle. The notation means that $\hat{\omega}(e0) = 1$ and $\hat{\omega}(e1) = \hat{\omega}(e2) = \hat{\omega}(e3) = 0$. The weighted laplacian matrix $\mathcal{L}_W$ defined in Eq. 4.6 for X^1 is

$$\frac{1}{3}\begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$$

when a uniform weight of 1 is assigned to all edges. Then, we obtain a smoothed cocycle $\bar{\omega} = \omega + d\bar{\alpha} = [0.5, 0.5, 0.5, 0.5]^T$ with a 0-cochain $\bar{\alpha} = [1, 0.5, 1, 1.5]^T$ which minimizes $\|\mathcal{L}_W(\hat{\omega} + d\bar{\alpha})\|_2^2$ to 0.

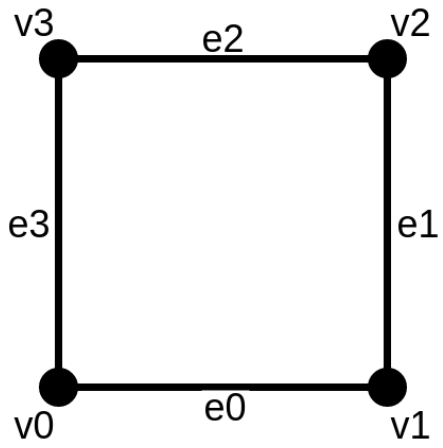

Figure 4.1: A simple example loop.[25]

4.3.2 Example datasets

In this section, we show the smoothed representative 1- and 2-cocycles and the enriched barcodes using artificial datasets. We create some example input functions defined on the nodes and aim to reflect the information about these functions on the enriched barcodes.

Two adjacent annuluses

We first consider a point cloud sampled from two adjacent annulus with radii 1 and centered at $(0, 0)$ and $(2, 2)$ as shown in Figure 4.2. There are two significant H_1 bars associated to the two major circles. An example of the representative cocycles for the two long H_1 bars are shown in Figure 4.3a and b. The associated smoothed cocycles obtained by using the method described in Section 4.2.3 are shown in Figure 4.3c and d.

Given two datasets with similar geometry but different values on the nodes, we can use enriched barcodes to distinguish between them. See Figure 4.4 for example.

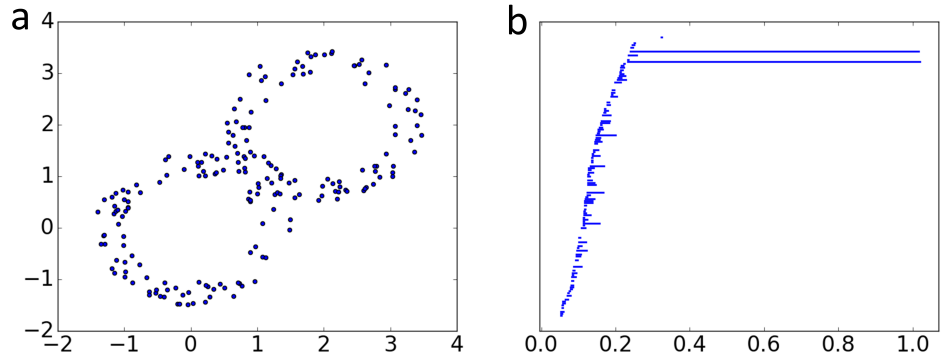

Figure 4.2: a: A point cloud sampled from two adjacent annulus. b: The corresponding H_1 barcode using alpha complex filtration.[25]

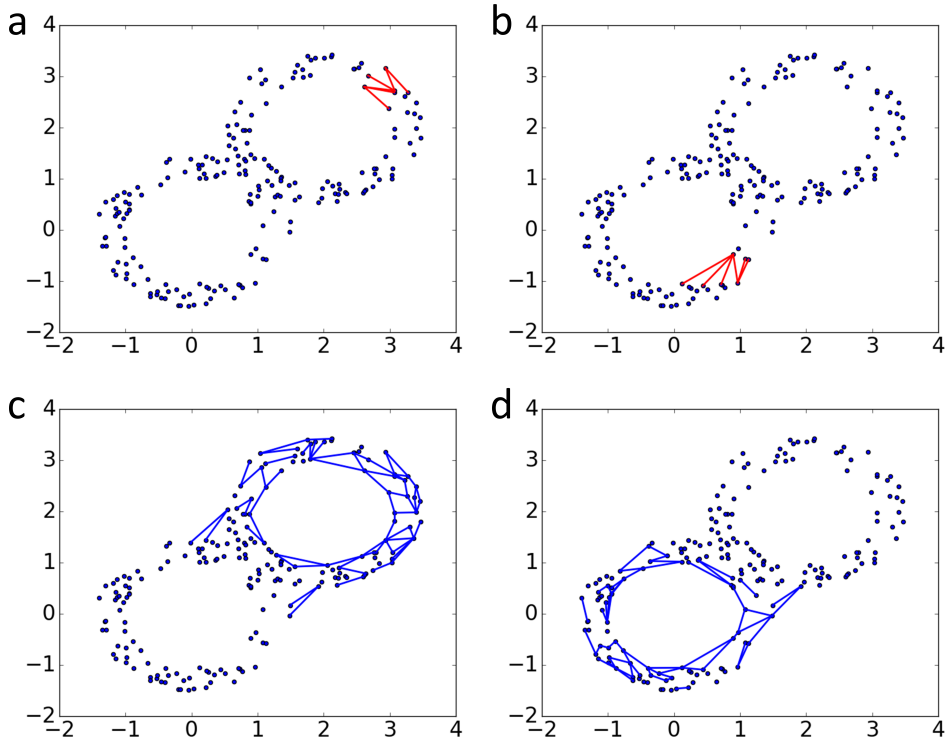

Figure 4.3: Example of smoothed H^1 cocycle.[25]
a and b: Two representative 1-cocycles corresponding to the two long H_1 bars. The edges where the cocycles take nonzero values are drawn in red. c and d: The smoothed 1-cocycles associated to the representative cocycles. The edges where the cocycles take values with magnitudes greater than or equal to 0.035 are drawn in blue. The smoothing is done on the subcomplexes associated to the filtration values at the middle of the corresponding bars.

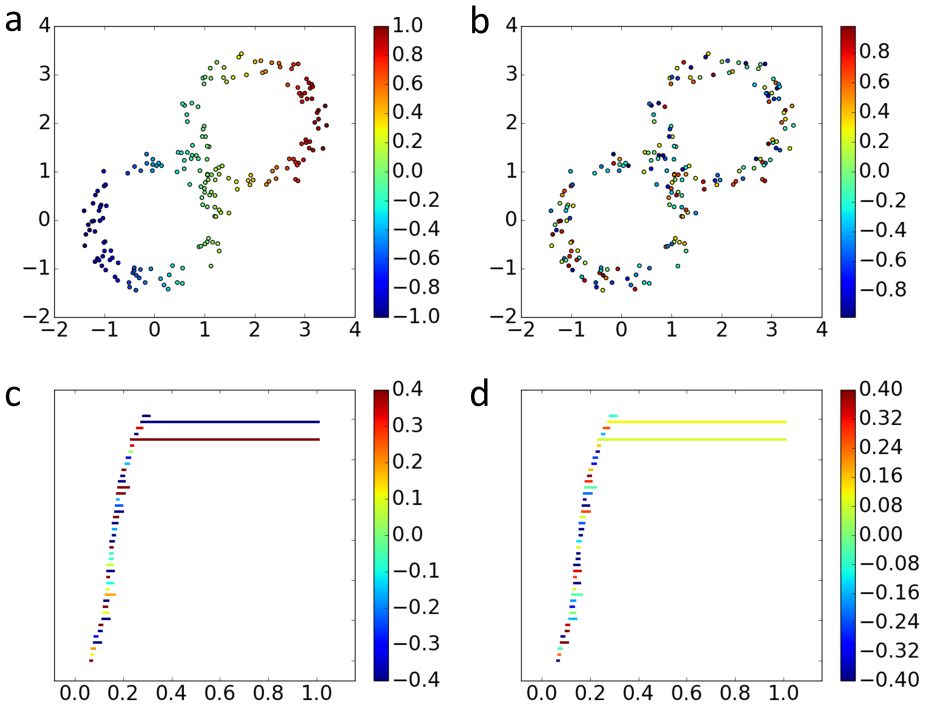

Figure 4.4: a and b: Two datasets with similar geometry but different information given on the nodes. c and d: The differences are revealed in the enriched H_1 barcodes.[25]

Cuboid minus two balls

In this example, the object considered is a rectangular cuboid ($[0,4] \times [0,2] \times [0,2]$) subtracted by two balls with radius of 0.5 centered at $(1,1,1)$ and $(3,1,1)$. Two thousand points are first sampled from a uniform distribution over the cuboid and the ones that are inside the balls are deleted. The dataset with values on the points, the two smoothed cocycles corresponding to the two voids, and the enriched barcode are shown in Figure 4.5.

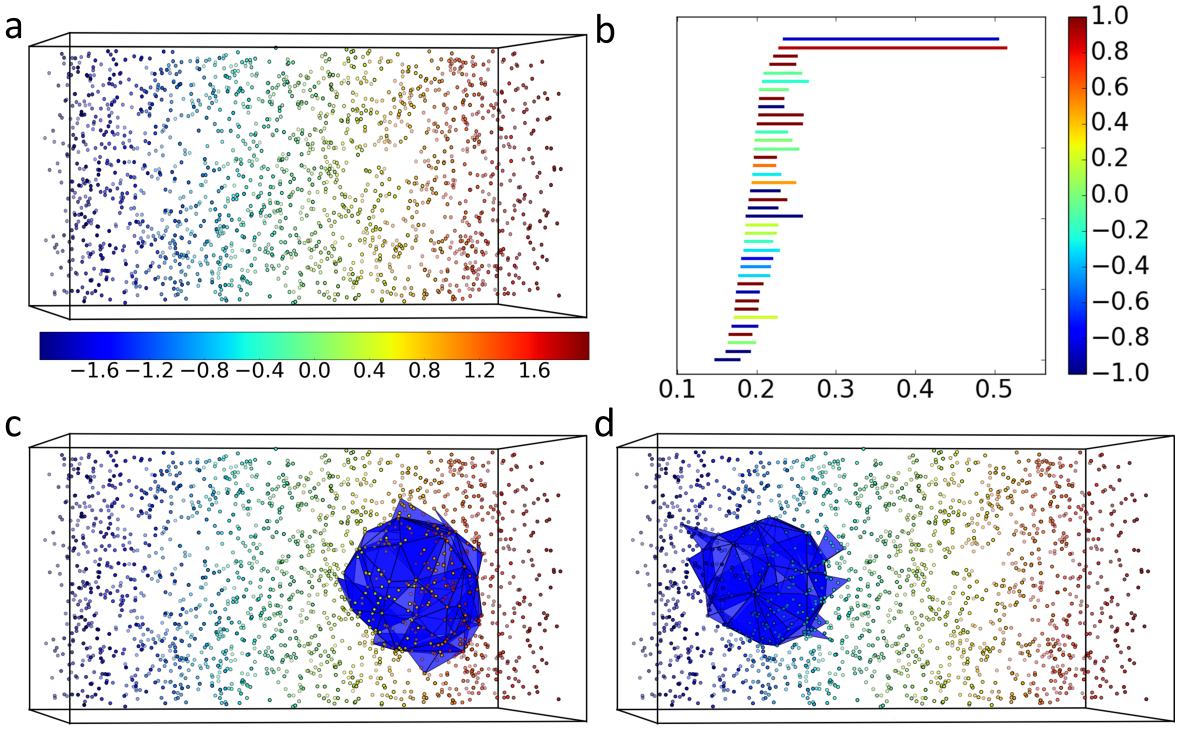

Figure 4.5: Persistent cohomology enriched barcode example of data points sampled from porous cuboid.[25]
a: The points sampled from an object which is a box subtracted by two balls. b: The H_2 enriched barcode showing the two voids in the blue and red regions of the original dataset. c and d: The two smoothed 2-cocycles. The faces where the cocycles take absolute values greater than or equal to 0.01 are drawn in blue. The smoothing is done on the subcomplexes associated to the filtration values at the middle of the corresponding bars.

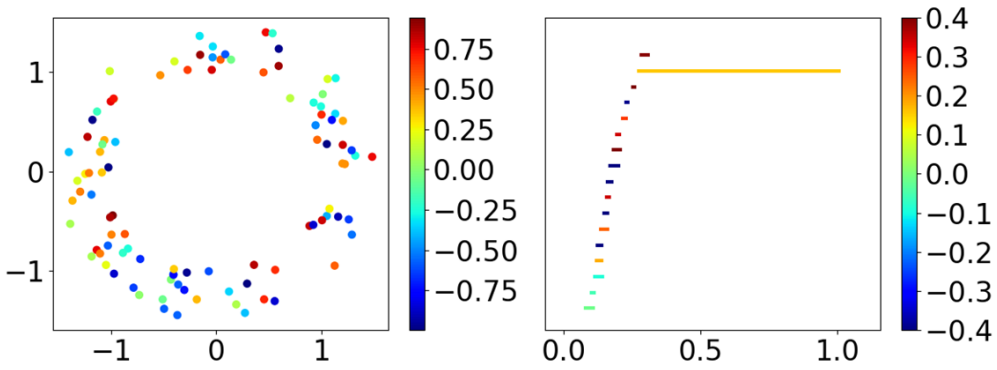

Figure 4.6: D3 dataset sampled from an annulus with randomly assigned values on the points and corresponding H_1 enriched barcode.

4.3.3 Wasserstein distance based similarity

We illustrate in this section the measurement of similarities among the persistent cohomology enriched barcodes. We use the enriched barcodes from three datasets, D1, D2, and D3. Here, D1 and D2 are the two datasets shown in Fig. 4.4a and b, respectively, while D3 is shown in the left chart of Fig. 4.6. The Wasserstein characteristics curve defined in Eq. (4.19) for three datasets, i.e., D1, D2 and D3, are shown in Fig. 4.7. Here, D1 and D2 have the same geometry and thus their curve is more on the left side which means a smaller distance between their persistent homology barcodes. On the other hand, D3 has a similar value assignment on the points as that of D2, so their curve is on the bottom which means a smaller distance in the non-geometric information.

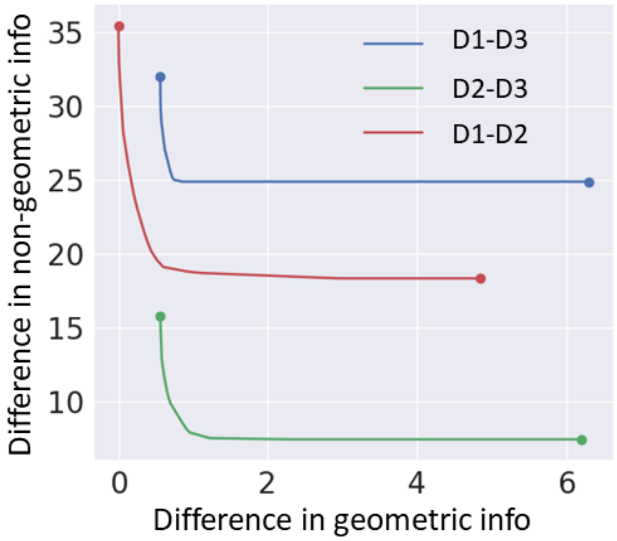

Figure 4.7: Wasserstein characteristics curve.

4.3.4 Analysis of molecules

Cyclic and cage-like structures often exist in complicated macromolecules in various scales. They can be as small as a benzene (a ring) containing 6 heavy atoms or an adamantane (a cage) containing 10 heavy atoms. And some macromolecules have a global configuration of cyclic or cage-like structures such as buckminsterfullerene and carbon nanotubes which are consisted of tens or hundreds of atoms. Persistent homology is good at detecting these structures in multiple scales and when we label the atoms by their element types, we can also reveal the element composition of the detected structures. Specifically, if oxygen is of interest, we construct an input function f_0 (see Section 4.2.4) that is defined on the nodes representing the atoms, and outputs 1 on oxygen atoms and 0 elsewhere. We illustrate this application using a cyclic structure cucurbit[8]uril and a cage-like structure $B_{24}N_{24}$ cage in this section.

Cucurbituril

In this example, we consider a macrocyclic molecule cucurbit[8]uril from the cucurbi-

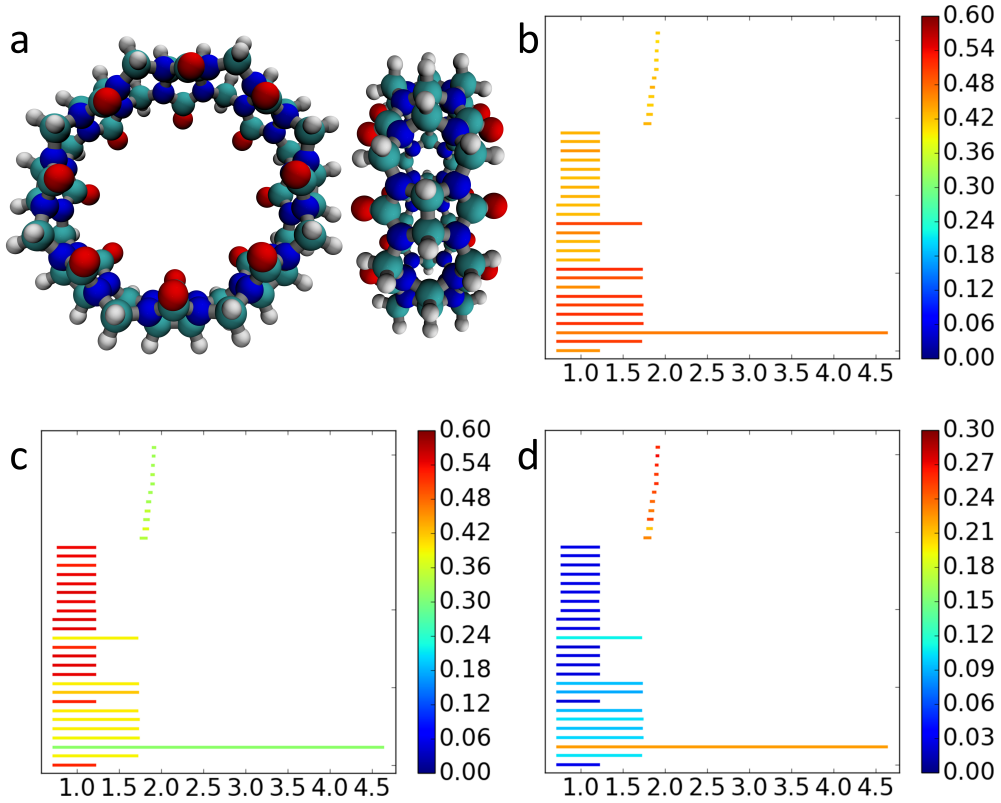

Figure 4.8: a: The cucurbit[8]uril molecule viewed from two different angles. The hydrogen, carbon, nitrogen, and oxygen atoms are colored in white, grey, blue, and red. b, c, and d: The H_1 enriched barcodes obtained by assigning 1 to nodes of the selected atom types (carbon, nitrogen, and oxygen) and 0 elsewhere.[25]

turil family. The molecule contains eight 6-membered rings and sixteen 5-membered rings consisted of carbon and nitrogen atoms. The rings form a big cyclic structure with a relatively tighter opening surrounded by oxygen atoms. The structure is taken from the provided structure in the SAMPL6 challenge [1] and the resulting H_1 barcodes are shown in Figure 4.8.

Boron nitride cage

The fullerene-like boron nitride cages exhibit spherical structures similar to fullerenes but are consisted of boron and nitrogen atoms. The global spherical structure is composed of a collection of local rings containing several atoms. A possible structure of $B_{24}N_{24}$ cage given in the supporting information of [209] is used in this example. The molecule and the

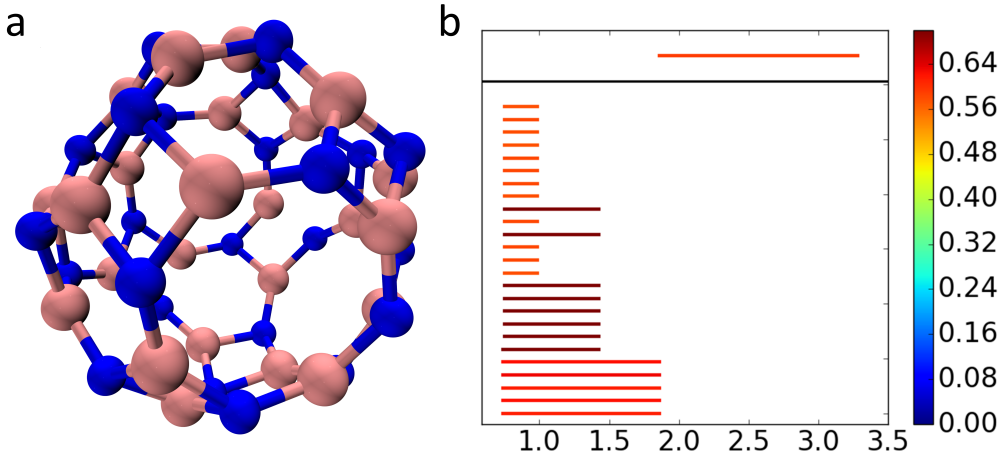

Figure 4.9: **a**: A structure of the $B_{24}N_{24}$ cage. The nitrogen and boron atoms are colored in blue and grey. **b**: The enriched barcodes obtained by assigning 1 to Boron atoms and 0 elsewhere. H_1 and H_2 barcodes are plotted in bottom and top panels.[25]

enriched barcode is shown in Figure 4.9.

In this application, the element type could be substituted by other information that the user is interested in, such as partial charge, van der Waals potential, and electrostatic solvation free energy.

4.3.5 An application to protein-ligand binding

An important component of computer-aided drug design is the prediction of protein-ligand binding affinity based on given protein-ligand complex structures. Persistent homology is good at identifying rings, tunnels, and cavities in various scales which are crucial to the protein-ligand complex stability and instability. In addition to geometry and topology, chemical and biological complexity also need to be addressed toward a practically useful method for this application. To this end, for example, the behavior of atoms of different element types can be described by computing persistent homology for subsets of atoms of the molecule of certain element types. The interaction between protein and ligand can be

emphasized by prohibiting an edge to form between two atoms both in the protein or the ligand. And the electrostatic interactions can be revealed by tweaking the distance matrix used for filtration to be the interaction strength computed with a chosen physical model such as Coulomb's law. However, the approaches described above disturb the original geometry and topology of the protein-ligand complexes. With the method proposed in this work, we are able to naturally embed the information such as atom type, atomic partial charges, and electrostatic interactions to the barcodes without disturbing the original geometric and topological setup of the molecular systems.

We compute the enriched barcodes for protein-ligand complexes, turn them into structured features, and combine with machine learning methods for the prediction of binding affinity. The procedure is validated on datasets from the PDBBind database [119] which includes experimentally derived protein-ligand complex structures and the associated binding affinities. An example of enriched barcode for atomic partial charges is shown in Fig. 4.10.

Enriched barcodes generation

In addition to the traditional barcode obtained from persistent homology computation, we also like to add descriptions of the electrostatic properties of the system. An efficient characterization of this property is the Coulomb potential where the interaction between two point charges is relatively described by $q_i q_j / r_{ij}$ where q_i and q_j are the point charges with a distance of r_{ij}. The atomic partial charges of proteins are assigned by using PDB2PQR software [58] with CHARMM22 force field. Two types of construction of the physical information are used to characterize the systems.

For dimension 0, a collection of subsets of atoms are first identified according to atom type. Specifically, 10 element types (C, N, O, S, P, F, Cl, Br, I, H) are considered for ligands and 5 element types are considered for proteins (C, N, O, S, H) and a total of 50 subsets of

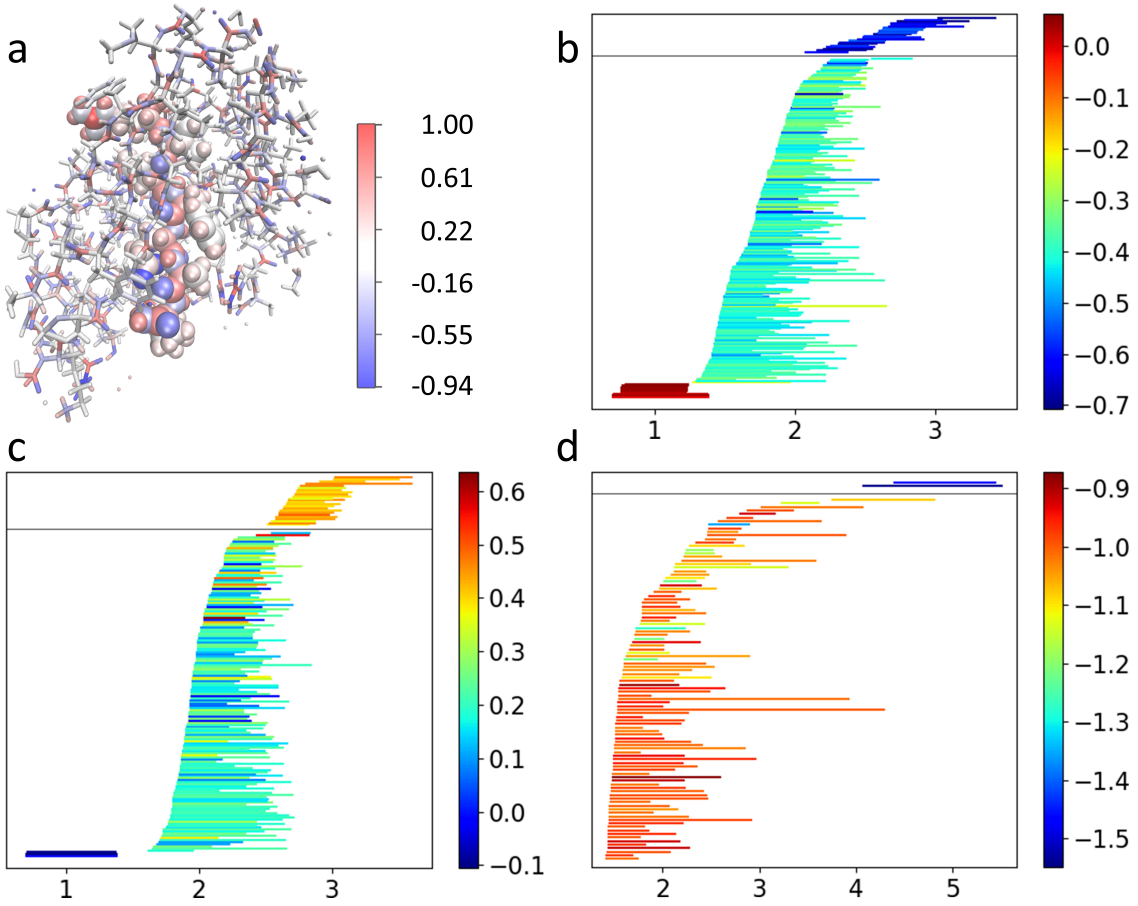

Figure 4.10: Enriched barcodes focusing on atomic partial charges. [25]

a: Ligand (as van der Waals spheres) and surrounding protein atoms (within 12 Å of ligand as thick sticks) of PDB entry 1a94. The color reflects the atomic partial charges. b, c, and d: Enriched barcodes for partial charges generated by computing persistent cohomology with alpha complex filtration on all heavy atoms, all carbon atoms, and nitrogen and oxygen atoms respectively. The top panel shows H_2 barcode and the bottom one shows H_1 barcode.

atoms are selected by choosing one element type from each component (protein or ligand). The pairwise distance matrix based on Euclidean distance is tweaked by setting distances between atoms both from protein or ligand to infinity which emphasizes the interactions between protein and ligand. Based on the tweaked distance matrix, persistent (co)homology computation with Rips complex is performed. The electric potential is computed for each atom with its nearest neighbor in the different part of the protein-ligand complex and is put on this atom as the additional information. We define the input function $f_0^0 : X^0 \to \mathbb{R}$ to take 0 on protein atoms and to take the value discussed above on ligand atoms. The average potential over ligand atoms in each 0-cocycle representative is used to generate features. In this way, the favorability of the protein ligand electrostatic interactions is explicitly described.

For dimension 1 and 2, the input function $f_0^1 : X^1 \to \mathbb{R}$ is defined to output the absolute value of electric potential on edges connecting two atoms to characterize the interaction strengths. The Coulomb potential is modeled as

$$E_{ij} = k_e \frac{q_i q_j}{r_{ij}},$$

where k_e is Coulomb's constant, q_i and q_j are the partial charges of atoms i and j, and r_{ij} is the distance between the two atoms. Persistent (co)homology with alpha complex is computed on three subsets of the protein-ligand complexes, all heavy atoms, all carbon atoms, and all oxygen/nitrogen atoms. For simplicity, all enriched barcodes are computed only at the middle points of the bars.

Featurization of barcodes

Given an enriched barcode, $B = \{\{b_i, d_i, f_i^*\}\}_{i \in I}$ obtained by applying the proposed method to a dataset with an input function f_0 (see Section 4.2.4), we turn it into fixed

shape array required by the machine learning algorithms we choose. Here, the input function is f_0^0 or f_0^1 described in the previous section when computing 0th dimensional persistent (co)homology or in higher dimensions.

For dimension 0, we first identify a range of scales to focus on and in this application, we are interested in the interval $[0, 12)$Å. The interval is then divided into 6 subintervals $\{[l_j^0, r_j^0)\}_j = \{[0, 2.5), [2.5, 3), [3, 3.5), [3.5, 4.5), [4.5, 6), [6, 12)\}$ to address different types of interactions. For dimension 0, we are interested in the death values of the bars. Therefore, a collection of index sets marking the deaths values of the bars that fall into each subinterval is calculated as

$$I_j^0 = \{i \in I | l_j \leq d_i < r_j\}.$$

For dimension 1 and 2, we are interested in the interval $[0, 6)$Å with Alpha complex filtration. The interval is then divided into 6 equal length subintervals $\{[l_j^{1,2}, r_j^{1,2})\}_j$. We then define a collection of index sets marking the bars that overlap with each subinterval,

$$I_j^{1,2} = \{i \in I | b_i < r_j^{1,2}, d_i \geq l_j^{1,2}\}.$$

Given a collection of index sets $\{I_j\}_j$, a feature vector $\mathbf{v}^h(B)$ is defined as

$$\Big(\mathbf{v}^h(B)\Big)_j = |I_j|.$$

When $\{I_j^0\}_j$ is used, it characterizes the number of component merging events in each filtration parameter interval. When $\{I_j^{1,2}\}$ is used, it reflects the ranks of homology groups at certain stage along the course of filtration.

And a feature vector $\mathbf{v}^f(B, f_0)$ can be generated subsequently to address the information

of the predefined function on the homology generators,

$$\Big(\mathbf{v}^f(B, f_0)\Big)_j = \frac{\sum\limits_{i \in I_j} \bar{f}_i^*}{|I_j|},$$

where $\bar{f}_i^* = (\int_{b_i}^{d_i} f_i^*(x)dx)/(d_i - b_i)$.

Machine learning model

The application of predicting protein-ligand binding affinity based on structures can be regarded as a supervised learning problem. Generally speaking, we are given a collection of pairs of input and output $\{(x_i, y_i)\}$ and there is a chosen model which is a function $M(x; \theta)$ with tunable parameters θ. The training process is to find a specific setting for the function M that globally or locally minimizes a penalty function which depends on the given data $\{(x_i, y_i)\}$ and the parameter set θ. Once trained, the model can be used to predict the output for a newly given input. We choose gradient booting trees (GBT) method for its accuracy, robustness, and efficiency. GBT is an ensemble of trees method with single decision trees as building blocks. The training of a GBT model is done by adding one tree at a time according to reduce loss of current model. In practice, different randomly selected subsets of the training data and features are used for each update of the model to reduce overfitting. For every result reported in Table 4.2, a parameter search is done by cross-validation within the training set where model performance is judged by Pearson's correlation coefficient. The candidate values for hyper-parameters tried are summarized in Table 4.1. Another hyper-parameter max_feature is set to sqrt because of the relatively large number of features. The GradientBoostingRegressor module in scikit-learn (version 0.17.1) [152] software is used.

Application

We test the improvement of the enriched barcodes with electrostatic information in the

Hyper Prm.	Candidates
n_estimators	5000, 10000, 20000
max_depth	4, 8, 16
min_samples_split	5, 10, 20
learning_rate	0.0025, 0.005, 0.01
subsample	0.25, 0.5, 0.75
min_samples_leaf	1, 3

Table 4.1: Candidate values for hyper-parameters of the gradient boosting trees model.

	2007	2013	2015	2016
Dim 0 w/o elec.	0.802 (1.47))	0.754 (1.56)	0.745 (1.56)	0.824 (1.32)
Dim 0 w. elec.	0.796 (1.50)	0.768 (1.53)	0.763 (1.53)	0.833 (1.31)
Dim 1&2 w/o elec.	0.726 (1.65)	0.706 (1.67)	0.718 (1.62)	0.767 (1.46)
Dim 1&2 w. elec.	0.738 (1.64)	0.734 (1.60)	0.737 (1.59)	0.778 (1.44)

Table 4.2: The predictor performance is evaluated by training on PDBBind refined set excluding the core set and testing on the core set of a certain year's version. The median Pearson's correlation coefficient (root mean squared error) among 10 repeated experiments is reported.

cases of 0th dimension and higher dimensions using the PDBBind database. The predictor performance is improved by using the enriched barcode embedding the electrostatics information. The results are listed in Table 4.2.

4.4 Discussion and conclusion

Utilizing the richer information carried by cohomology, we introduce a method to reflect in the barcodes the additional information from the dimensions that are not used for persistent homology computation. This is achieved by finding a smoothed representative cocycle with respect to a Laplacian directly defined on the simplicial complexes or a weighted graph Laplacian. The smoothed cocycles then serve as measures on the simplicial complexes and

allow us to do integration of the additional information. As a result, in addition to the original persistence barcodes, functions of filtration values associated to each persistence pair are constructed which enriches the information carried by the barcodes. A similarity score based on Wasserstein distance is introduced to analyze these enriched barcodes. Going back to the problem that physical properties should be embedded in the persistence barcodes to better describe the biomolecules which motivated the development of this method at the beginning, the method shows to improve the performance in practical tasks of protein-ligand binding affinity prediction by adding electrostatics information to the barcodes.

This method is potentially useful for a wider range of applications where data come with multiple heterogeneous dimensions. For example, when analyzing time series dataset in 3-dimensional space using persistent homology, some specific treatment such as Vineyards [45] is used instead of directly doing the computation in $\mathbb{R}^4$. Computing persistence over multiple dimensions at the same time [32] also helps to address this general situation. For one specific dimension of a multidimensional dataset, there are also cases where we would like to embed the information carried in this dimension to the persistence barcodes computed for other dimensions rather than looking at the persistence for this dimension. For example, persistent homology can find us loops and voids in biomolecular structures and we are interested in question that what kind of charges do these homology generators carry. In this case, the duality between homology and cohomology enables us to better localize the homology generators and to examine the charge distributions associated to each generator.

Chapter 5

Evolutionary homology for coupled dynamical systems

5.1 Introduction

In addition to analyzing static structures of biomolecules, we are also interested in analyzing the dynamics of the biomolecular systems which is often related to important biomolecular properties such as stability and instability. The time evolution of complex phenomena is often described by dynamical systems, i.e., mathematical models built on differential equations for continuous dynamical systems or on difference equations for discrete dynamical systems [197, 148, 92, 192]. Most dynamical systems have their origins in Newtonian mechanics. However, these mathematical models typically only admit highly reduced descriptions of the original complex physical systems, and thus their continuous forms do not have to satisfy the Euler-Lagrange equation of the least action principle. Although a low-dimensional dynamical system is not expected to describe the full dynamics of a complex physical system, its long-term behavior, such as the existence of steady states (i.e., fixed points) and/or chaotic states, offers a qualitative understanding of the underlying system. Focused on ergodic systems, dynamic mappings, bifurcation theory, and chaos theory, the study of dynamical systems is a mathematical subject in its own right, drawing on analysis, geometry, and topology. Dynam-

ical systems are motivated by real-world applications, having a wide range of applications to physics, chemistry, biology, medicine, engineering, economics, and finance. Nevertheless, essentially all of the analyses in these applications are qualitative and phenomenological in nature.

In order to pass from qualitative to quantitative evaluation of these systems, we look to the newly emerging field of topological data analysis (TDA) [28, 61, 79, 98, 81, 134]. Specifically, we use persistent homology which provides multiscale topological characterization of datasets. The use of homology for the analysis of dynamical systems and time series analysis predates and intertwines with the beginnings of persistent homology [98, 126, 76, 3, 164, 163, 162]. More recently, there has been increased interest in the combination of persistent homology with time series analysis [165]. Some common methods include computing the persistent homology of the Takens embedding [157, 156, 154, 103, 102, 104], studying the sublevelset homology of movies [106, 178], and working with the additional structure afforded by persistent cohomology [54, 18, 181]. Wang and Wei have defined temporal persistent homology over the solution of a partial differential equation derived from differential geometry [186]. This method encodes spatial connectivity into temporal persistence in the Laplace-Beltrami flow, and offers accurate quantitative prediction of fullerene isomer stability in terms of total curvature energy for over 500 fullerene molecules. Closely related to our work, Stolz *et al.* have recently constructed persistent homology from time-dependent functional networks associated with coupled time series [174]. This work uses weight rank clique filtration over a defined parameter reflecting similarities between trajectories to characterize coupled dynamical systems.

The objective of the present work is to (1) define a new simplicial complex filtration using a coupled dynamical system as input, which encodes the time evolution and synchronization

of the system, and (2) use the persistent homology of this filtration to study the system itself. The resulting persistence barcode is what we call the evolutionary homology (EH) barcode. We are particularly interested in the encoding and decoding of the topological connectivity of a real physical system into a dynamical system. To this end, we regulate the dynamical system by a generalized graph Laplacian matrix defined on a physical system with distinct topology. As such, the regulation encodes the topological information into the time evolution of the dynamical system. We use a well-studied dynamical system, the Lorenz system, to illustrate our EH formulation. The Lorenz attractor is utilized to facilitate the control and synchronization of chaotic oscillators by weighted graph Laplacian matrices generated from protein C_α networks. We create features from the EH barcodes originating from protein networks by using the Wasserstein and bottleneck metrics. The resulting outputs in various topological dimensions are directly correlated with physical properties of protein residue networks. Finally, to demonstrate the quantitative analysis power of the proposed EH, we apply the present method to the prediction of protein thermal fluctuations characterized by experimental B-factors. We show that the present EH provides some of the most accurate B-factor predictions for a set of 364 proteins. Our approach not only provides a new tool for quantitatively analyzing the behavior of dynamical systems but also extends the utility of dynamical systems to the quantitative modeling and prediction of important physical/biological problems.

5.2 Methods

This section is devoted to the methods and algorithms. In Sec. 5.2.1, we give a brief discussion of coupled dynamical systems and their stability control via a correlation (coupling) matrix

which embeds topological connectivity of a physical system into the dynamical system. For background of persistent homology theory, we refer readers to Section 2.1. A concept of topological learning is given in Section 5.2.1.3. We then define evolutionary homology on coupled dynamical systems in Section 5.2.2. Finally, the full pipeline as applied to protein flexibility analysis is outlined in Section 5.2.3.

5.2.1 Coupled dynamical systems

The general control of coupled dynamical systems has been well-studied in the literature [148, 92, 192, 197]. A brief review is given in this section.

5.2.1.1 Oscillators and coupling

We consider the coupling of N n-dimensional dynamical systems

$$\frac{d\mathbf{u}_i}{dt} = g(\mathbf{u}_i),\ i = 1, 2, \cdots, N,$$

where $\mathbf{u}_i = \{u_{i,1}, u_{i,2}, \cdots, u_{i,n}\}^T$ is a column vector of size n. In our setup, each $\mathbf{u}_i$ is associated to a point $\mathbf{r}_i \in \mathbb{R}^d$ which will be used to determine influence in the coupling.

The coupling of the systems can be very general, but a specific selection can be an $N \times N$ graph Laplacian matrix A defined for pairwise interactions

$$A_{ij} = \begin{cases} I(i,j),\ i \neq j, \\ -\sum\limits_{l \neq i} A_{il},\ i = j, \end{cases}$$

where $I(i,j)$ is a value describing the degree of influence on the ith system induced by the jth

system. We assume undirected graph edges $I(i,j) = I(j,i)$. Let $\mathbf{u} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_N\}^T$ be a column vector with $\mathbf{u}_i = \{u_{i,1}, u_{i,2}, \cdots, u_{i,n}\}^T$. The coupled system is an $N \times n$-dimensional dynamical system modeled as

$$\frac{d\mathbf{u}}{dt} = \mathbf{G}(\mathbf{u}) + \epsilon(A \otimes \Gamma)\mathbf{u}, \tag{5.1}$$

where $\mathbf{G}(\mathbf{u}) = \{g(\mathbf{u}_1), g(\mathbf{u}_2), \cdots, g(\mathbf{u}_N)\}^T$, ϵ is a parameter, and Γ is an $n \times n$ predefined linking matrix.

One choice of g is the Lorenz oscillator defined as

$$g(\mathbf{u}_i) = \begin{bmatrix} \delta(u_{i,2} - u_{i,1}) \\ u_{i,1}(\gamma - u_{i,3}) - u_{i,2} \\ u_{i,1}u_{i,2} - \beta u_{i,3} \end{bmatrix} \tag{5.2}$$

where δ, γ, and β are parameters determining the state of the Lorenz oscillator. This system is used in this work because of its relative simplicity, rich dynamics and well-understood behavior.

5.2.1.2 Stability and controllability

Let $\mathbf{s}(t)$ satisfy $d\mathbf{s}/dt = g(\mathbf{s})$. We say the coupled systems are in synchronous state if

$$\mathbf{u}_1(t) = \mathbf{u}_2(t) = \cdots = \mathbf{u}_N(t) = \mathbf{s}(t).$$

The stability can be analyzed using $\mathbf{v} = \{\mathbf{u}_1 - \mathbf{s}, \mathbf{u}_2 - \mathbf{s}, \cdots, \mathbf{u}_N - \mathbf{s}\}^T$ with the following equation obtained by linearizing Eq. (5.1)

$$\frac{d\mathbf{v}}{dt} = [I_N \otimes Dg(\mathbf{s}) + \epsilon(A \otimes \Gamma)]\mathbf{v}, \tag{5.3}$$

where I_N is the $N \times N$ unit matrix and $Dg(\mathbf{s})$ is the Jacobian of g on $\mathbf{s}$.

The stability of the synchronous state in Eq. (5.3) can be studied by eigenvalue analysis of graph Laplacian A. Since the graph Laplacian A for undirected graph is symmetric, it only admits real eigenvalues. After diagonalizing A as

$$A\phi_j = \lambda_j \phi_j,\ j = 1, 2, \cdots, N,$$

where λ_j is the jth eigenvalue and ϕ_j is the jth eigenvector, $\mathbf{v}$ can be represented by

$$\mathbf{v} = \sum_{j=1}^{N} \mathbf{w}_j(t)\phi_j.$$

Then, the original problem on the coupled systems of dimension $N \times n$ can be studied independently on the n-dimensional systems

$$\frac{d\mathbf{w}_j}{dt} = (Dg(\mathbf{s}) + \epsilon\lambda_j\Gamma)\mathbf{w}_j,\ j = 1, 2, \cdots, N. \tag{5.4}$$

Let L_{max} be the largest Lyapunov characteristic exponent of the jth system governed by Eq. (5.4). It can be decomposed as $L_{max} = L_g + L_c$, where L_g is the largest Lyapunov exponent of the system $d\mathbf{s}/dt = g(\mathbf{s})$ and L_c depends on λ_j and Γ. In many numerical experiments in this work, we set $\Gamma = I_n$, an $n \times n$ identity matrix. Then the stability of

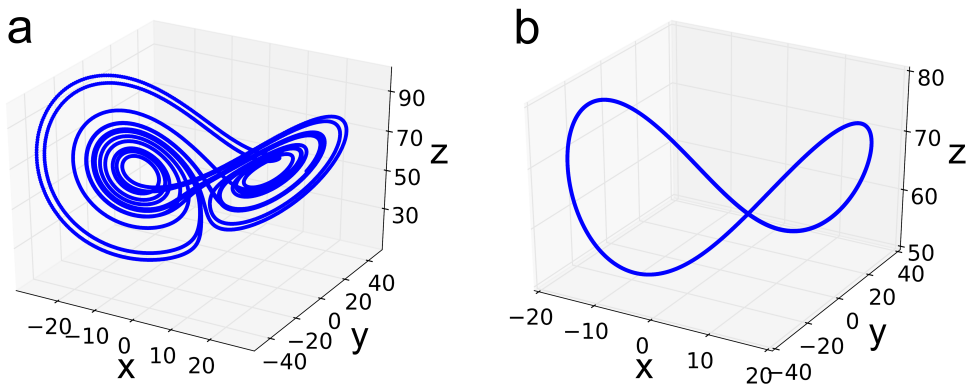

Figure 5.1: a: Chaotic trajectory of one oscillator without coupling. b: The 70 synchronized oscillators associated with the carbon C_α atoms of protein PDB:1E68 are plotted together.[22]

the coupled systems is determined by the second largest eigenvalue λ_2. The critical coupling strength ϵ_0 can, therefore, be derived as $\epsilon_0 = L_g/(-\lambda_2)$. A requirement for the coupled systems to synchronize is that $\epsilon > \epsilon_0$, while $\epsilon \leq \epsilon_0$ causes instability.

An example of chaos controlled by coupling is shown in Fig. 5.1. In this example, each alpha carbon atom (C_α) of protein PDB:1E68 is associated with a Lorenz oscillator and the underlying locations of the oscillators are used to construct the coupling matrix. The specific coupling matrix $A = A^{\text{geo}} + A^{\text{seq}}$ used in this example is a sum of a graph Laplacian matrix defined using the geometric coupling,

$$A_{ij}^{\text{geo}} = \begin{cases} -1, \text{ if } i \neq j \text{ and } d_{ij}^{\text{org}} < \epsilon_d, \\ -\sum_{l \neq i} A_{il}^{\text{geo}}, \ i = j, \end{cases}$$

and another which takes the amino acid sequence into account,

$$A_{ij}^{\text{seq}} = \begin{cases} \epsilon_{\text{seq}}, \text{ if } (i+1+N) \bmod N = j, \\ -\epsilon_{\text{seq}}, \text{ if } (i-1+N) \bmod N = j, \\ 0, \text{ otherwise.} \end{cases}$$

Here, d^{org} is the distance function in the original space; that is, the Euclidean distance between atoms in this example. The mod operator is used because the protein in this example is circular. The parameters used for the example of Fig. 5.1 are $\epsilon_{\text{seq}} = 0.7$ for sequence coupling, $\epsilon_d = 4\text{Å}$ for spatial cutoff, and $\delta = 10$, $\gamma = 60$, and $\beta = 8/3$ for the Lorenz system. The parameters in Eq. (5.1) are $\epsilon = 10$ and

$$\Gamma = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Initial values for all oscillators are randomly chosen.

5.2.1.3 Topological learning

The proposed method provides a vast but relatively abstract characterization of the objects of interest. It is potentially powerful in quantitative analysis, but is difficult to use out of the box machine learning or data analysis techniques. In regression analysis or the training part of supervised learning, with $\mathbf{B}_i$ being the collection of sets of barcodes corresponding to the ith entry in the training data, the problem can be cast into the following minimization

problem,

$$\min_{\theta_b \in \mathbf{\Theta}_b, \theta_m \in \mathbf{\Theta}_m} \sum_{i \in I} L(\mathbf{y}_i, \mathbf{F}(\mathbf{B}_i; \theta_b); \theta_m),$$

where L is a scalar loss function, $\mathbf{y}_i$ is the collection of target values in the training set, $\mathbf{F}$ is a function that maps barcodes to suitable input for the learning models, and θ_b and θ_m are the parameters to be optimized within the search domains $\mathbf{\Theta}_b$ and $\mathbf{\Theta}_m$ respectively. The form of the loss function also depends on the choice of metric and machine learning/regression model.

A function $\mathbf{F}$ which translates barcodes to structured representation (tensors with fixed dimension) can be used with popular machine learning models including random forest, gradient boosting trees and deep neural networks. Another popular class of models are the kernel based models that depend on an abstract measurement of the similarity or distance between the entries.

Our choices for $\mathbf{F}$, defined in Eq. (5.9) of Sec. 5.2.3, will arise from looking at the distance from the specified barcode to the empty barcode and there is no tuning of θ_b. In Sec. 5.3.2 where we quantitatively analyze protein residue flexibility, we evaluate our method by checking the correlation between each topological feature defined in Eq. (5.9) and the experimental value (blind prediction) as well as the correlation between the output of a linear regression with multiple topological features and the experimental value (regression). In the former case, there is no parameter to be optimized, while in the latter case, the specific minimization problem can be written as

$$\min_{\theta_m \in \mathbb{R}^{n+1}} \sum_{i \in I} \left(y_i - \left[\mathrm{EH}_i^{p_1,1}, \cdots, \mathrm{EH}_i^{p_n,n}, 1 \right] \cdot \theta_m \right)^2,$$

where $\mathrm{EH}_i^{p_k,k}$ is the topological parameter by computing the p_k-Wasserstein distance of the empty set to the kth barcode associated with the EH computation of the ith protein residue (node). I is the set of indexes of all residues in the protein and y_i is the experimental B-factor for the ith protein residue which quantitatively reflects flexibility.

5.2.2 Evolutionary homology (EH) and the EH barcodes

Consider a system of N not yet synchronized oscillators $\{\mathbf{u}_1, \cdots, \mathbf{u}_N\}$ associated to a collection of N embedded points, $\{\mathbf{r}_1, \cdots, \mathbf{r}_N\} \subset \mathbb{R}^d$. We assume the global synchronized state is a periodic orbit denoted $\mathbf{s}(t)$ for $t \in [t_0, t_1]$ where $\mathbf{s}(t_0) = \mathbf{s}(t_1)$. For flexibility and generality, we work on post-processed trajectories obtained by applying a transformation function on the original trajectories, $\widehat{\mathbf{u}}_i(t) := T(\mathbf{u}_i(t))$. The choice of function T is flexible and should fit the applications; in this work, we choose

$$T(\mathbf{u}_i(t)) = \min_{t' \in [t_0, t_1]} \|\mathbf{u}_i(t) - \mathbf{s}(t')\|_2, \tag{5.5}$$

which gives 1-dimensional trajectories for simplicity. Further, in our specific example, $\widehat{\mathbf{s}}(t) := T(\mathbf{s}(t)) = 0$, but, again, this is not necessary in general.

We wish to study the effects on the synchronized system of N oscillators (an $(N \times 3)$-dimensional system) after perturbing one oscillator of interest. To this end, we set the initial values of all the oscillators except that of the ith oscillator to $\mathbf{s}(\bar{t})$ for a fixed $\bar{t} \in [t_0, t_1]$. The initial value of the ith oscillator is set to $\rho(\mathbf{s}(\bar{t}))$ where ρ is a predefined function playing the role of introducing perturbance to the system. After the system starts running, some oscillators will be dragged away from and then go back to the periodic orbit as the perturbance is propagated and relaxed through the system. Let $\widehat{\mathbf{u}}_j^i(t)$ denote the modified

trajectory of the jth oscillator after perturbing the ith oscillator at $t = 0$. We focus on the subset of nodes that are affected by the perturbation,

$$V^i = \left\{ n_j \mid \max_{t>0} \{ \min_{t' \in [t_0, t_1]} \|\widehat{\mathbf{u}}_j^i(t) - \widehat{\mathbf{s}}(t')\|_2 \} \geq \epsilon_p \right\}$$

for some fixed ϵ_p determining how much deviation from synchronization constitutes "being affected".

5.2.2.1 Filtration function defined for coupled dynamical systems

Assuming we have perturbed the oscillator for node n_i, let $M = |V_i|$. We will now construct a function f on the complete simplicial complex, denoted by K or K_M, with M vertices. Here, we abuse notation and write $V_i = \{n_1, \cdots, n_M\}$. The filtration function $f : K_M \to \mathbb{R}$ is built to take into account the temporal pattern of the propagation of the perturbance through the coupled systems and the relaxation (going back to synchronization) of the coupled systems. It requires the advance choice of three parameters:

- $\epsilon_p \geq 0$, mentioned above, which determines when a trajectory is far enough from the global synchronized state, $\mathbf{s}(t)$ to be considered unsynchronized,
- $\epsilon_{\text{sync}} \geq 0$ which controls when two trajectories are close enough to be considered synchronized with each other, and
- $\epsilon_d \geq 0$ which is a distance parameter in the space where the points $\mathbf{r}_i$ are embedded, giving control on when the objects represented by the oscillators are far enough apart to be considered insignificant to each other.

We will define the function f by giving its value on simplices in the order of increasing dimension. Define

$$t^i_{\text{sync}} = \min\left\{ t \mid \int_t^\infty \|\widehat{\mathbf{u}}^i_j(t') - \widehat{\mathbf{u}}^i_k(t')\|_2 \, dt' \leq \frac{\epsilon_{\text{sync}}}{2}, \, \forall j,k \right\}.$$

That is, t^i_{sync} is the first time at which all oscillators have returned to the global synchronized state after perturbing the ith oscillator. The value of the filtration function for the vertex n_j is defined as

$$f(n_j) = \min\left\{ \{t \mid \min_{t' \in [t_0, t_1]} \|\widehat{\mathbf{u}}^i_j(t) - \widehat{\mathbf{s}}(t')\|_2 \geq \epsilon_p\} \cup \{t^i_{\text{sync}}\} \right\}. \tag{5.6}$$

Next, we give the function value f for the edges of K. To avoid the involvement of any insignificant interaction between oscillators, an edge between n_j and n_k denoted by e_{jk} is allowed in the earlier stage of the filtration only if $d^{\text{org}}_{jk} \leq \epsilon_d$ where d^{org}_{jk} is the distance between $\mathbf{r}_i$ and $\mathbf{r}_j$ in $\mathbb{R}^d$. Specifically, the value of the filtration function for the edge e_{jk} is defined as

$$f(e_{jk}) = \begin{cases} \max\left\{ \min\{t | \int_t^\infty \|\widehat{\mathbf{u}}^i_j(t') - \widehat{\mathbf{u}}^i_k(t')\|_2 \, dt' \leq \epsilon_{\text{sync}}\}, f(n_j), f(n_k) \right\}, & \text{if } d^{\text{org}}_{jk} \leq \epsilon_d \\ t^i_{\text{sync}}, & \text{if } d^{\text{org}}_{jk} > \epsilon_d. \end{cases} \tag{5.7}$$

It should be noted that to this point, f defines a filtration function because when $d^{\text{org}}_{jk} \leq \epsilon_d$, $f(n_j) \leq f(e_{jk})$ according to the definition given in Eq. (5.7). The property also holds when $d^{\text{org}}_{jk} > \epsilon_d$ because $f(n_j) \leq t_{\text{sync}}$ according to the definition in Eq. (5.6) and $f(e_{jk})$ equals t_{sync} in this case.

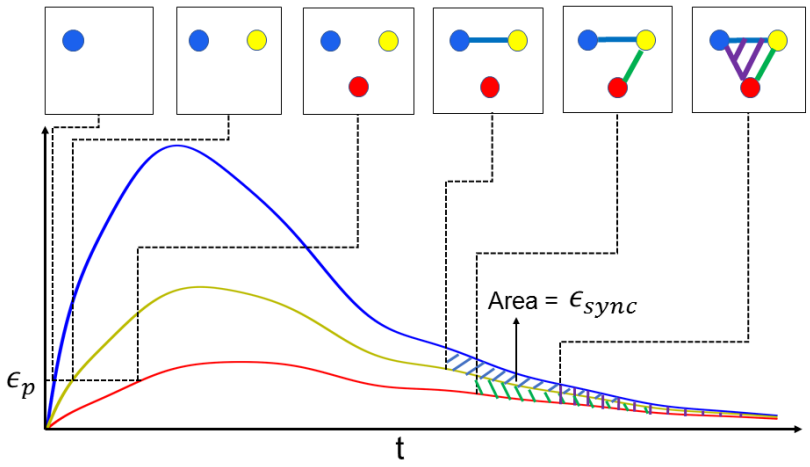

Figure 5.2: The filtration of the simplicial complex associated to three 1-dimensional trajectories.[22]
The trajectories are generate as defined in Sec. 5.2.2.1. Here, each vertex corresponds to the trajectory with the same color. A vertex is added when its trajectory value exceeds the parameter ϵ_p; an edge is added when its two associated trajectories become close enough together that the area between the curves after that time is below the parameter ϵ_{sync}. Triangles and higher dimensional simplices are added when all necessary edges have been included in the filtration.

We extend the function to the higher dimensional simplices using the definition on the 1-skeleton. A simplex σ of dimension higher than one is included in $K(x)$ if all of its 1-dimensional faces are already included; that is, its filtration value is defined iteratively by dimension as

$$f(\sigma) = \max_{\tau \leq \sigma} f(\tau),$$

where the max is taken over all codim-1 faces of σ. Taking the filtration of K using this function (c.f. Eq. (2.1)) means that topological changes only occur at the collection of function values $\{f(n_i)\}_i \cup \{f(e_{jk})\}_{j \neq k}$. Fig. 5.2 shows the filtration constructed for an example consisting of three trajectories.

5.2.2.2 Definition of evolutionary homology

The previous section gives a function $f_i : K_{|V^i|} \to \mathbb{R}$ defined on the complete simplicial complex with $|V^i|$ vertices for each $i = 1, \cdots, N$. From the filtration defined by f_i, we then compute the persistence barcode for homology dimension k, which we call the *kth EH barcode*, denoted B_i^k. The persistent homology computation for dimension ≥ 1 on the filtered simplicial complex is done using the software package Ripser [8] using the fact that k-dimensional homology only requires knowledge of k and $k+1$-dimensional simplices. The 0-dimensional homology is computed with a modification of the union-find algorithm.

Fig. 5.3 gives an example of the geometric configurations of two sets of points associated to Lorenz oscillators and their resulting EH barcodes. The EH barcodes effectively examine the local properties of significant cycles in the original space which is important when the data is intrinsically discrete instead of a discrete sampling of a continuous space. As a result, the point clouds with different geometry but similar barcodes using traditional persistence methods[1] may be distinguished by EH barcodes.

5.2.3 Protein residue flexibility analysis

In this section, we combine all the methods to formulate realistic protein residue flexibility analysis using the EH barcodes. Consider a protein with N residues and let $\mathbf{r}_i$ denote the position of the alpha carbon (C_α) atom of the ith residue. The coupled systems defined in Eq. (5.1) are used to study protein flexibility with each protein residue represented by a 3-dimensional Lorenz system. Define the distance for the atoms in the original space as the Euclidean distance between the Cα atoms, $d^{\text{org}}(\mathbf{r}_i, \mathbf{r}_j) = \|\mathbf{r}_i - \mathbf{r}_j\|_2$. A weighted graph

[1] Here, traditional means the Vietoris-Rips filtration on the point cloud induced by the embedding

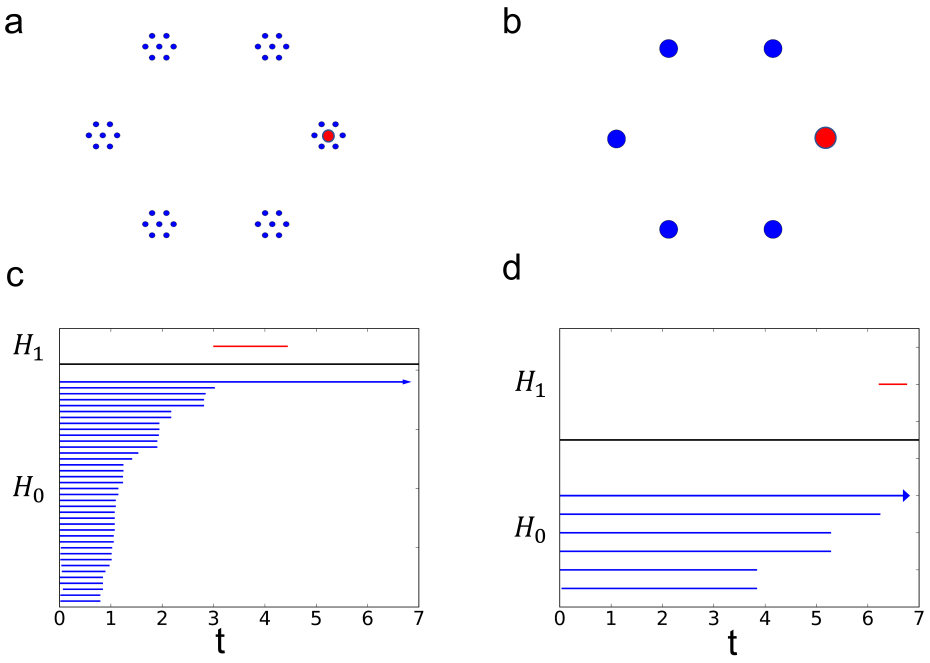

Figure 5.3: An example of the construction of the evolutionary homology barcode.[22] The geometry of two embedded systems is shown in Figures (a) and (b). Specifically, (b) consists of six vertices of a regular hexagon with side length of e_1; and (a) consists of the vertices in (b) with the addition of the vertices of hexagons with a side length of $e_2 \ll e_1$ centered at each of the previous vertices; here, $e_1 = 8$ and $e_2 = 1$. Figures (c) and (d) are the EH barcodes corresponding to Figures (a) and (b) respectively. A collection of coupled Lorenz systems is used with parameters $\delta = 1$, $\gamma = 12$, $\beta = 8/3$, $\mu = 8$, $k = 2$, $\Gamma = I_3$, and $\epsilon = 0.12$; see Eqs. (5.2), (5.8) and (5.1). In the model for the ith residue, marked in red, the system is perturbed from the synchronized state by setting $u_{i,3} = 2s_3$ with s_3 being the value of the third variable of the dynamical system at the synchronized state and is simulated with step size $h = 0.01$ from $t = 0$ using the fourth-order Runge-Kutta method. The calculation of persistent homology using the Vietoris-Rips filtration with Euclidean distance on the point clouds delivers similar bars corresponding to the 1-dimensional holes in (a) and (b) which are $[e_1 - e_2, 2(e_1 - e_2))$ and $[e_1, 2e_1)$.

Laplacian matrix is constructed based on the distance function d^{org} to prescribe the coupling strength between the oscillators and is defined as

$$A_{ij} = \begin{cases} e^{-(d^{\text{org}}(\mathbf{r}_i,\mathbf{r}_j)/\mu)^\kappa}, \ i \neq j, \\ \\ -\sum\limits_{l \neq i} A_{il}, \ i = j, \end{cases} \tag{5.8}$$

where μ and κ are tunable parameters.

To quantitatively study the flexibility of a protein, one needs to extract topological in-

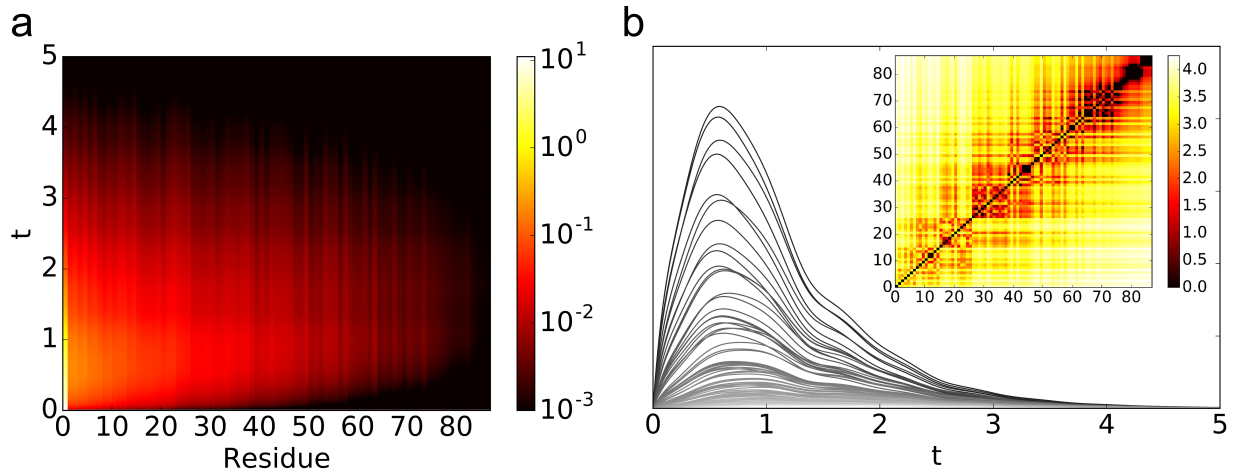

Figure 5.4: The result of perturbing residue 31 in protein (PDB:1ABA).[22]
(a) The modified trajectories as defined in Eq. (5.5) is plotted for each residue after the perturbation at $t = 0$ as a heatmap. The residues are ordered by the (geometric) distance to the perturbed site from the closest to the farthest. (b) The modified trajectories as defined in Eq. (5.5) is plotted for each residue after the perturbation at $t = 0$ as line plots. The darker lines are closer to the perturbed site. The heatmap shows filtration value for the edges as defined in Eq. (5.7) and the order of residues is the same as in (a). The parameters for the coupled Lorenz system and the perturbation method are the same as that of Fig. 5.3.

formation for each residue. To this end, we go through the process given in the previous sections once for each residue. When addressing the ith residue, we perturb the ith oscillator at a time point in a synchronized system and take this state as the initial condition for the coupled systems. See Fig. 5.4 for an example of this procedure when perturbing the oscillator attached to a residue for a given embedding of one particular protein.

A collection of modified trajectories $\{\widehat{\mathbf{u}}_i(t)\}_{i=1}^N$ is obtained with the transformation function defined in Eq. (5.5). The persistence over time for $\{\widehat{\mathbf{u}}_i(t)\}_{i=1}^N$ is computed following the filtration procedure defined in Sec. 5.2.2.1. Let B_i^k be the kth EH barcode obtained from the experiment of perturbing the oscillator corresponding to residue i. We introduce the following topological features to relate to protein flexibility:

$$\mathrm{EH}_i^{p,k} = d_{W,p}(B_i^k, \emptyset), \tag{5.9}$$

where $d_{W,p}$ for $1 \leq p < \infty$ is the p-Wasserstein distance and $p = \infty$ is the bottleneck distance. We will show that these features characterize the behavior of this particular collection of barcodes, which in turn, captures the topological pattern of the coupled dynamical systems arising from the underlying protein structure.

The flexibility of any given residue is reflected by how the perturbation induced stress is propagated and relaxed through the interaction with the neighbors. Such a relaxation process will induce the change in the states of the nearby oscillators. Therefore, the records of the time evolution of this subset of coupled oscillators in terms of topological invariants can be used to analyze and predict protein flexibility.

The difference in results of the procedure can be seen in the example of Fig. 5.5 where the control of chaotic oscillators attached to a partially disordered protein (PDB:2RVQ) and a well-folded protein (PDB:1UBQ) is demonstrated. Clearly, the folded part of protein 2RVQ has strong correlations or interactions among residues from residue 25 to residue 110, which leads to the synchronization of the associated chaotic oscillators. In contrast, the random coil part of protein 2RVQ does not have much coupling or interaction among residues. Consequently, the associated chaotic oscillators remain in chaotic dynamics during the time evolution. For folded protein 1UBQ, the associated chaotic oscillators become synchronized within a few steps of simulation, except for a small flexible tail. This behavior underpins the use of coupled dynamical systems for protein flexibility analysis.

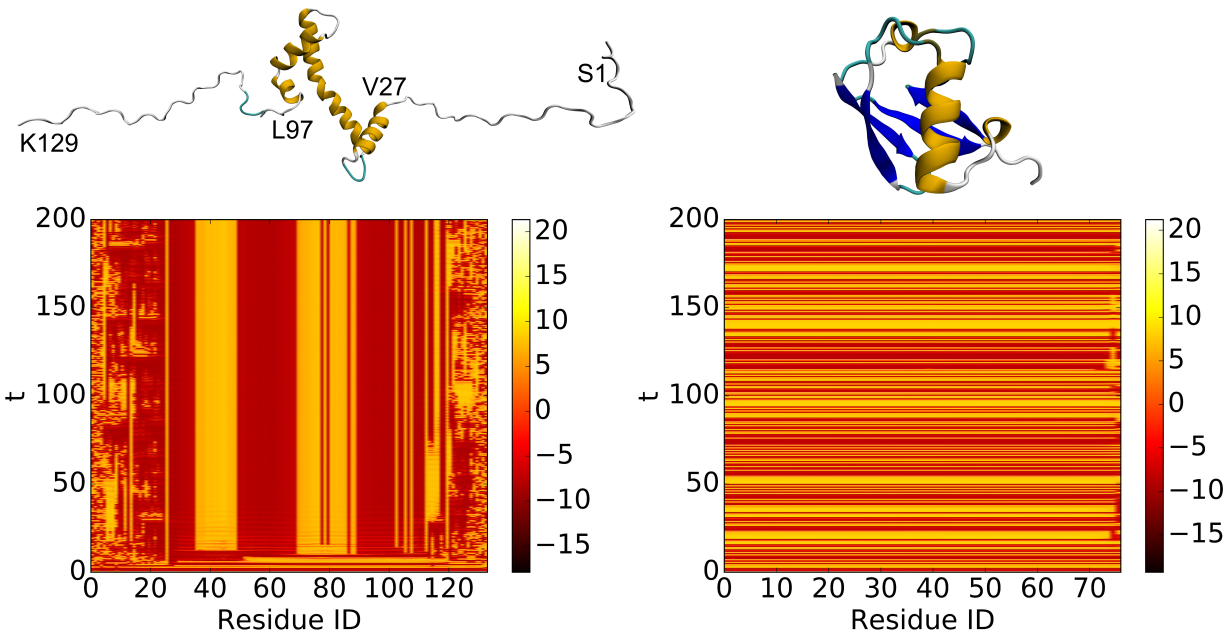

Figure 5.5: Left: partially disordered protein, model 1 of PDB:2RVQ. Right: well folded protien, PDB:1UBQ.[22]
The $u_{i,1}$ value of each dynamical system is plotted as heatmap. The Lorenz system defined in Eq. (5.2) is used with the parameters $\delta = 10, \gamma = 28, \beta = 8/3$. The coupling matrix A defined in Eq. (5.8) has parameters $\mu = 14, \kappa = 2$. The coupled system defined in Eq. (5.1) has parameters $\Gamma = I_3$ and $\epsilon = 0.12$. The system is initialized with a random value between 0 and 1 and is simulated from $t = 0$ to $t = 200$ with step size $h = 0.01$. The system is numerically solved using the 4-th order Runge-Kutta method. It can be seen from the heatmaps that the oscillators corresponding to the disordered regions behave asynchronously.

5.3 Results

5.3.1 Disordered and flexible protein regions

To illustrate the correlation between protein residue flexibility and the topological features defined in Eq. (5.9), we study several proteins with intrinsically disordered regions. Intrinsically disordered proteins lack stable 3-dimensional molecular structures. Partially disordered proteins refer to the intrinsically disordered proteins that contain both stable structure and flexible regions. In nature, the disordered regions may play important roles in biological processes which requires flexibility.

In what follows, we always work with the coupled Lorenz system parameters, pertur-

bation method for the ith residue, and simulation described in Fig. 5.3. The simulation is stopped when all oscillators go back to synchronized state. This process is repeated for each residue. Two NMR structures of partially disordered proteins PDB:2ME9 and PDB:2MT6 are studied. The topological features are computed for each model of the structures and are averaged over the models. The results are plotted in Fig. 5.6. The disordered regions clearly correlate to the peaks of $\mathrm{EH}^{\infty,0}$ and the valleys of $\mathrm{EH}^{\infty,1}$, $\mathrm{EH}^{1,0}$, and $\mathrm{EH}^{1,1}$. The topological features are also able to distinguish between relatively stable coils (the coils that are consistent among the NMR models) and the disordered parts (the parts that differ among the NMR models).

5.3.2 Protein B-factor prediction

Protein B-factors quantitatively measure the relative thermal motion of each atom and reflects atomic flexibility. The x-ray crystal structures deposited to the Protein Data Bank contain experimentally derived B-factors which can be used to validate the proposed method [151, 145]. To analyze protein flexible regions, B-factor prediction is needed for protein structures built from computational models and some experimentally solved structures using NMR or cryo-EM techniques. Normal mode analysis (NMA) is one of the first methods proposed for B-factor predictions [83]. The Gaussian network model (GNM) [7] was known for its better accuracy and efficiency compared to a variety of earlier methods [204]. The multiscale flexibility-rigidity index (FRI), which is about 20% more accurate than GNM, has been established as the state-of-the-art in the B-factor predictions [146].

In this section, we compute the correlation between the topological features and the experimentally derived protein B-factors. We further test the proposed topological features by building a simple linear regression model with a least square penalty against the exper-

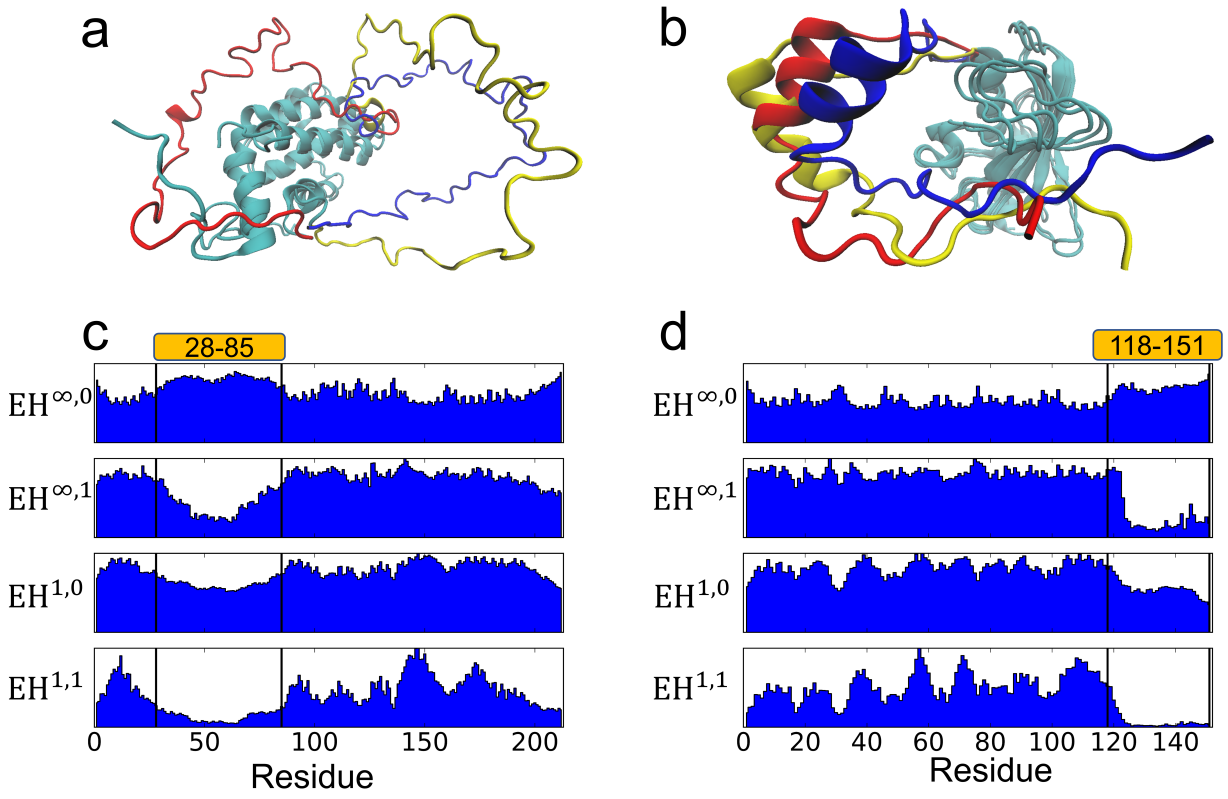

Figure 5.6: (a) Models 1-3 of PDB:2ME9 with the disordered region colored in blue, red, and yellow for the three models. (b) Similar plot as (a) for PDB:2MT6. (c) Topological features for PDB:2ME9 whose large disordered region is from residue 28 to residue 85. (d) Topological features for PDB:2MT6 whose large disordered region is from residue 118 to residue 151. [22]

imental B-factors. A collection of 364 diverse proteins reported in the literature is chosen as the validation data (The set of 365 proteins [145] excepts PDB:1AGN due to issue in reported B-factors [146]). The size of the proteins ranges from tens to thousands of amino acid residues. The topological features in the model are the same as the setup given in Sec. 5.3.1. An example of the resulting persistence barcodes for relatively rigid and relatively flexible residues are shown in Fig. 5.7. It is seen that the residue with a relatively small B-factor has many H_0, H_1 and H_2 bars. Compared to the residue having a large B-factor, it has a much richer dynamical response and barcodes with more bars. Additionally, its H_0 bars are much shorter, indicating a stronger interaction with neighbor residues.

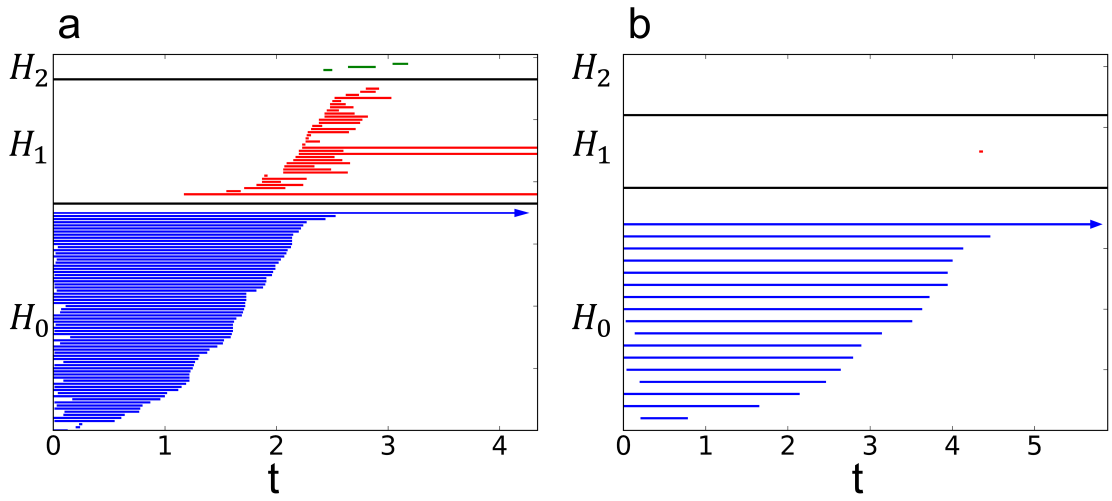

Figure 5.7: Barcode plots for two residues. (a) Residue 6 of PDB:2NUH with a B-factor of 12.13 Å^2. (b) Residue 49 of PDB:2NUH with a B-factor of 33.4 Å^2.[22]

The computed topological features are plotted against a relatively small protein and a relatively large protein in Fig. 5.8. Clearly, 0-dimensional topological features, specifically EH$^{\infty,0}$, provide a reasonable approximation to experimental B-factors. The regression using all topological information, EH, offers very good approximation to experimental B-factors. A summary of the results and a comparison to other methods is shown in Table 5.1 for the set of 364 proteins. It is seen that the present evolutionary topology based prediction outperforms other methods in computational biophysics. A possible reason for this excellent performance is that the proposed method gives a more detailed description of residue interactions in terms of three different topological dimensions and two distance metrics. This example indicates that the proposed EH has a great potential for other important biophysical applications, including the predictions of protein-ligand binding affinities, mutation induced protein stability changes and protein-protein interactions.

Method	R_P	Description
$EH^{\infty,0}$	0.586	Topological feature
$EH^{\infty,1}$	-0.039	Topological feature
$EH^{\infty,2}$	-0.097	Topological feature
$EH^{1,0}$	-0.477	Topological feature
$EH^{1,1}$	-0.381	Topological feature
$EH^{1,2}$	-0.104	Topological feature
$EH^{2,0}$	0.188	Topological feature
$EH^{2,1}$	-0.258	Topological feature
$EH^{2,2}$	-0.100	Topological feature
EH	0.691	Topological features

Method	R_P	Description
EH	0.691	Topological metrics
mFRI	0.670	Multiscale FRI [146]
pfFRI	0.626	Parameter free FRI [145]
GNM	0.565	Gaussian network model [145]

Table 5.1: The averaged Pearson correlation coefficients (R_P) between the computed values (blind prediction for the topological features and regression for the rest of the models) and the experimental B-factors for a set of 364 proteins [146] (Left: Prediction R_Ps based on EH barcodes. Right: A comparison of the R_Ps of predictions from different methods.). Here, EH is the linear regression using $EH^{\infty,0}$, $EH^{\infty,1}$, $EH^{1,0}$, $EH^{1,1}$, $EH^{2,0}$, and $EH^{2,1}$ within each protein. For a few large and multi-chain proteins (i.e., 1F8R, 1H6V, 1KMM, 2D5W, 3HHP, 1QKI, and 2Q52), to reduce the computational time and as a good approximation, we compute their EH barcodes on separated (protein) chains. We see from the table at right that the proposed EH barcode method outperforms other methods in this application.

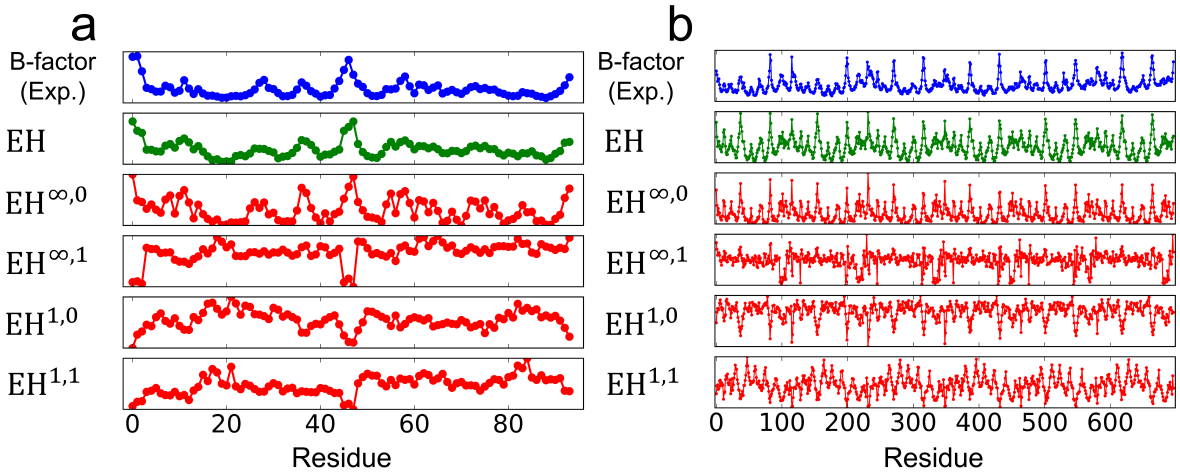

Figure 5.8: B-factors and the computed topological features. EH shows the linear regression with $EH^{\infty,0}$, $EH^{\infty,1}$, $EH^{1,0}$, $EH^{1,1}$, $EH^{2,0}$, and $EH^{2,1}$ within each protein. (a) PDB:3PSM with 94 residues. (b) PDB:3SZH with 697 residues.[22]

5.4 Conclusion

Many dynamical systems are designed to understand the time-dependent phenomena in the real world. The topological analysis of dynamical systems is scarce in general, partially due to the fact that the topological structure of most dynamical systems is typically simple. In this work, we have introduced evolutionary homology (EH) to analyze the topology and its time evolution of dynamical systems. We present a method to embed external topology of a physical system into dynamical systems. EH examines the embedded topology and converts it into topological invariants over time. The resulting barcode representation of the topological persistence is able to unveil the quantitative topology-function relationship of the embedded physical system.

We have chosen the well-known Lorenz system as an example to illustrate our EH formulation. An important biophysical problem, protein flexibility analysis, is employed to demonstrate the proposed topological embedding of realistic physical systems into dynamical systems. Specifically, we construct weighted graph Laplacian matrices from protein

networks to regulate the Lorenz system, which leads to the synchronization of the chaotic oscillators associated with protein residue network nodes. Simplices, simplicial complexes, and homology groups are subsequently defined using the adjacent Lorenz oscillator trajectories. Topological invariants and their persistence are computed over the time evolution (filtration) of these oscillators, unveiling protein thermal fluctuations at each residue. The Wasserstein and bottleneck metrics are used to quantitatively discriminate EH barcodes from different protein residues. The resulting model using the EH barcodes is found to outperform both geometric graph and spectral graph theory based methods in the protein B-factor predictions of a commonly used benchmark set of 364 proteins.

The proposed EH method can be used to study the topological structure of a general physical system. Moreover, the present method extends the utility of dynamical systems, which are usually designed for qualitative analysis, to the quantitative modeling and prediction of realistic physical systems. Finally, the proposed approach can be readily applied to the study of a wide variety of topology-function relationships, both within computational biology such as the role of topology in protein-ligand, protein-protein, protein-metal and protein-nucleic acid interactions; but also to other interactive graphs and networks in science and engineering.

Chapter 6

Topological characterization of static macrobiomolecules and small molecules

6.1 Introduction

Despite the competitive out-of-box performance of persistent homology as a featurization tool in supervised learning tasks, careful designs tailored for the field of applications can further improve the performance. To this end, we present a comprehensive assessment of representability of persistent homology for both macromolecules and small molecules. We introduce several ways of characterizing macromolecules and small molecules whose quality are judged by the problem of protein-ligand binding affinity prediction. We propose multi-level persistent homology specifically designed for the characterization of small molecules and we show that such representation is able to capture subtle changes in small molecules. The best protocol benchmarked on the protein-ligand binding affinity prediction problem is then applied to virtual screening where more than a hundred thousand target-candidate pairs are involved and our model achieved top performance in a benchmark using the DUD (directory of useful decoys) database.

The rest of this chapter is organized as follows. In Section 6.2, we discuss in detail the biology or chemistry that needs to be addressed in a representation tool. We introduce several persistent homology based methods focusing on geometry, chemistry, electrostatics properties of macromolecules and small molecules in Section 6.3. The results are shown in Section 6.4 followed by a detailed discussion about several model aspects in persistent homology based machine learning models for biomolecular systems in Section 6.5.

6.2 Biological considerations

The development of persistent homology was motivated by its potential in the dimensionality reduction, abstraction and simplification of biomolcular complexity [62]. In the early applications of persistent homology to biomolecules, emphasis was given on major or global features (long-persistent features) to derive descriptive tools. For example, persistent homology was used to identify the tunnel in a Gramicidin A channel [62] and to study membrane fusion [99]. For the predictive modeling of biomolecules, features of a wide range of scales might all be important to the target quantity [198]. At the global scale, the biomolecular conformation should be captured. At the intermediate scale, the smaller intra-domain cavities need to be identified. At the most local scale, the important substructures should be addressed, such as the pyrrolidine in the side chain of proline. Biomolecules are both structurally and biologically complex. Their geometric and biological complexities include covalent bonds, non-covalent interactions, effects of chirality, cis and trans distinctions, multi-leveled protein structures, and protein-ligand and protein-nucleic acid complexes. Covering a large range of spatial scales is not enough for a powerful model. The biological details should also be explored. We address the underlying biology and physics by modifying the distance func-

tion and selecting various sets of atoms according to element types, to describe different interactions. Some biological considerations are discussed in this section.

Covalent bonds.

Covalent bonds are formed via shared electron pairs or bonding pairs. The lengths and the number of covalent bonds can be easily detected from 0th dimensional barcodes. For macromolecules, the same type of covalent bonds have very similar bond lengths and thus 0th dimensional barcode patterns.

Non-covalent interactions.

Non-covalent interactions play a critical role in maintaining the 3D structure of biomolecules and mediating chemical and biological processes, such as solvation, binding, protein-DNA specification, molecular self-assembly, etc. Physically, non-covalent interactions are due to electrostatic, van der Waals forces, hydrogen bonds, π-effects, hydrophobic effects, etc. The ability to characterize non-covalent interactions is an essential task in any methodological development. The 1st and 2nd dimensional barcodes are suitable for the characterization of the arrangement of such interactions in a larger scale. Additionally, we propose multi-level persistence and electrostatic persistence to reveal local and pairwise non-covalent interactions via 0th dimensional barcodes as well.

Chirality, cis effect and trans effect.

Chirality, cis and trans effects are geometric properties of many molecules. Among them, chirality is a symmetry property such that a chiral molecule cannot be superposed on its mirror image. Cis and trans effects are due to molecular steric and electronic effects. Chirality, cis and trans effects often play a role in molecular kinetics, activity and catalysis, and thus their characterization is an important issue in developing topological methods. These effects should be reflected from barcodes of various dimensions.

Multi-leveled protein structures.

Protein structures are typically described in terms of primary, secondary, tertiary and quaternary levels. The protein primary structure is the linear sequence of amino acids in the polypeptide chain. Protein secondary structure refers to the local 3D structure of protein segments containing mainly α-helix and β-sheets, which are highly regular and can be easily detected by distinct Frenet-Serret frames. A tertiary structure refers to the 3D structure of a single polypeptide chain. Its formation involves various non-covalent and covalent interactions including salt bridges, hydrophobic effects, and often disulfide bonds. A quaternary structure refers to the aggregation of two or more individual folded protein subunits into a 3D multi-subunit complex. Protein structures are further complicated by its functional domains, motifs, and particular folds. The protein structural diversity and complexity result in the challenge and opportunity for methodological developments.

Protein-ligand, protein-protein, and protein-nucleic acid complexes.

Topological characterization of proteins is further complicated by protein interactions or binding with ligands (drugs), other proteins, DNA and/or RNA molecules. Although a normal protein involves only carbon (C), hydrogen (H), nitrogen (N), oxygen (O) and sulfur (S) atoms, its protein-ligand complexes bring a variety of other elements into the play, including, phosphorus (P), fluorine (F), chlorine (Cl), Bromine (Br), iodine (I), and many important biometals, such as calcium (Ca), potassium (K) sodium (Na), iron (Fe), copper (Cu), cobalt (Co), zinc (Zn), manganese (Mn), chromium (Cr), vanadium (V), tin (Sn), and molybdenum (Mo). Each biological element has important biological functions and its presence in biomolecules should be treated uniformly as a set of points in the point cloud data. The interaction of protein and nucleic acids can be very intricate.

6.3 Methods

6.3.1 Element specific persistent homology

One important issue is how to protect chemical and biological information during the topological simplification. As mentioned earlier, one should not treat different types of atoms as homogeneous points in a point cloud data. To this end, we propose element specific persistent homology or multi-component persistent homology to retain biological information in topological analysis [24]. The element selection is similar to a predefined vertex color configuration for graphs.

When all atoms are passed to persistent homology algorithms, the information extracted mainly reflects the overall geometric arrangement of a biomoelcule at different scales. By passing only atoms of certain element types or of certain roles to the persistent homology analysis, different types of interactions or geometric arrangements can be revealed. In protein-ligand binding modeling, the selection of all carbon atoms characterizes the hydrophobic interaction network whilst the selection of all nitrogen and/or oxygen atoms characterizes hydrophilic network and the network of potential hydrogen bonds. In the protein structural analysis, computation on all atoms can identify geometric voids inside the protein which may suggest structural instability and computation on only C_α atoms reveals the overall structure of amino acid backbones. In addition, combination of various selections of atoms based on element types provides very detailed description of the biomolecular system and the hidden relationships from the structure to function can then be learned by machine learning algorithms. This may lead to the discovery of important interactions not realized as *a prior*. This can be realized by passing the set of atoms of the selected element types to the persistent homology computation. This concept can be used with various constructions

of distance matrix for persistent homology computation based on Vietoris-Rips complex filtration.

6.3.2 Construction of distance matrix

Biomolecular systems are not only complex in geometry, but also in chemistry and biology. To effectively describe complex biomolecular systems, it is necessary to modify the filtration process. There are three commonly used filtrations, namely, radius filtration, distance matrix filtration, and density filtration, for biomolecules [198, 201]. The distance matrices can be used with a more abstract construction of simplicial complexes, such as Vietoris-Rips complex.

6.3.2.1 Multi-level persistent homology.

Small molecules such as ligands in protein-ligand complexes usually contain fewer atoms than large biomolecules such as proteins. Bonded atoms stay closer than non-bonded ones in most cases. As a result, the collection of 0th dimensional bars will mostly provide the information about the length of covalent bonds and the higher dimension barcodes will most likely be very sparse. It is difficult to capture non covalent bond interactions among atoms especially hydrogen bonds and van der Waals pairwise interactions in 0th dimensional barcodes. In order to describe non covalent interactions , we propose multi-level persistent homology, by simply modifying the distance matrix. Given the original distance matrix $\mathbf{M} = (d_{ij})$ with $1 \leq i, j \leq N$, the modified distance matrix is defined as

$$\widetilde{\mathbf{M}}_{ij} = \begin{cases} d_\infty, \text{ if atoms i and j are bonded,} \\ \mathbf{M}_{ij}, \text{ otherwise,} \end{cases} \tag{6.1}$$

where d_∞ is a large number which is set to be greater than the upper limit of the filtration value chosen by a persistent homology algorithm. Note that this matrix may fail to satisfy triangle inequality whilst still satisfies the construction principle of Rips complex.

The present multi-level persistent homology is able to describe any selected interactions of interest and delivers two benefits in characterizing biomolecules. Firstly, the pairwise non-covalent interactions can be reflected by the 0th dimensional barcodes. Secondly, such treatment generates more higher dimensional barcodes and the small structural fluctuation among different conformations of the same molecule can be captured. The persistent barcode representation of the molecule can be significantly enriched to better distinguish between different molecular structures and isomers. As an illustration, we take the ligand from the protein-ligand complex with PDB code "1BCD" which only has 10 atoms. A different conformation of the ligand is generated by using the Frog2 web server [128]. The persistent barcodes generated using Rips complex with the distance matrices $\mathbf{M}$ are identical and only have 0th dimensional bars due to the simple structure. In this case, the 0th dimensional bars only reflect the length of each bond and therefore fail to distinguish the two slightly different conformations of the same molecule. However, when the modified distance matrices $\widetilde{\mathbf{M}}$ are employed, the barcode representation is significantly enriched and is able to capture the tiny structural perturbation between the conformations. An illustration of the outcome from the modified distance matrix $\widetilde{\mathbf{M}}$ is shown in Fig. 6.1. A general nth level persistence characterization of molecules can be obtained with the distance matrix $\widetilde{\mathbf{M}}^n$ as,

$$\widetilde{\mathbf{M}}^n_{ij} = \begin{cases} d_\infty, D(i,j) \leq n \\ \\ \mathbf{M}_{ij}, \text{ otherwise,} \end{cases} \tag{6.2}$$

where $D(i, j)$ is the smallest number of bonds to travel from atom i to atom j and d_∞ is some number greater than the upper limit of the filtration value.

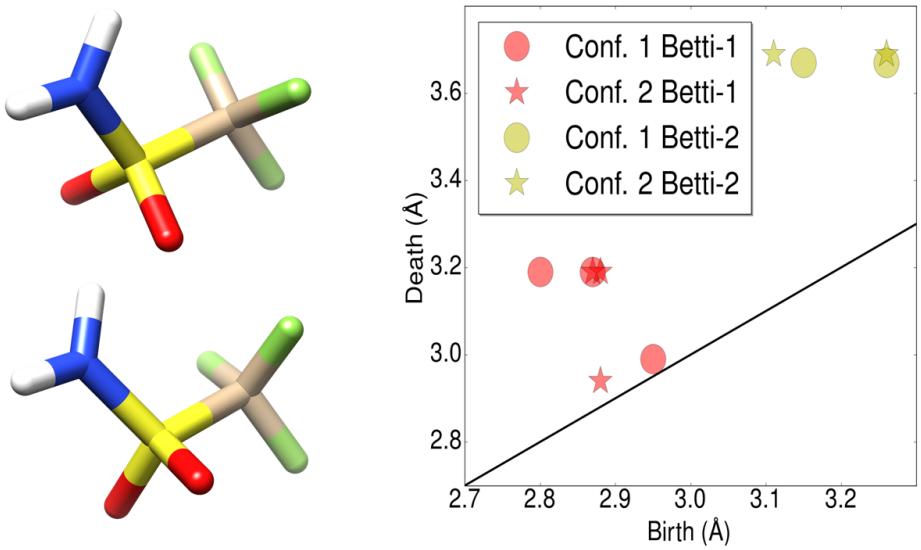

Figure 6.1: Multi-level persistent homology on simple small molecules.[20]
Illustration of representation ability of $\widetilde{\mathbf{M}}$ in reflecting structural perturbations among conformations of the same molecule. Left: The structural alignment of two conformations of the ligand in protein-ligand complex (PDB:1BCD). Right: The persistence diagram showing the 1st and 2nd dimensional results generated using Rips complex with $\widetilde{\mathbf{M}}$ for two conformations. It is worth noticing that the barcodes generated using Rips complex with $\mathbf{M}$ are identical for the two conformations.

6.3.2.2 Interactive persistent homology.

In protein-ligand binding analysis and analysis involving interactions, we are interested in the change of topological invariants induced by interactions that are caused by binding or other processes. Similar to the idea of multi-level persistent homology, we can design a distance matrix to focus on the interactions of interest. For a set of atoms, $A = A_1 \cup A_2$ with $A_1 \cap A_2 = \emptyset$ where only interactions between atoms from A_1 and atoms from A_2 are

of interest [24]. The interactive distance matrix $\widehat{\mathbf{M}}$ is defined as

$$\widehat{\mathbf{M}}_{ij} = \begin{cases} \mathbf{M}_{ij}, \text{ if } a_i \in A_1,\, a_j \in A_2 \text{ or } a_i \in A_2,\, a_j \in A_1, \\ d_\infty, \text{ otherwise,} \end{cases} \tag{6.3}$$

where $\mathbf{M}$ is the original distance matrix induced from Euclidean metrics or other correlation function based distances, a_i and a_j are atoms i and j , and d_∞ is a number greater than the upper limit of the filtration value. In applications, A_1 and A_2 can be respectively a set of atoms of the protein and a set of atoms of the ligand in a protein-ligand complex. In this case, the characterization of interactions between ligand and protein is an important task. In the modeling of point mutation induced protein stability changes, A_1 could be the set of atoms at the mutation site and A_2 could be the set of atoms of surrounding residues close to the mutation site. Similar treatment can be used for protein-protein and protein-nucleic acid interactions.

6.3.2.3 Correlation function based persistent homology.

For biomolecules, the interaction strength between pair of atoms usually does not align linearly to their Euclidean distances. For example, van der Waals interaction is often described by the Lennard-Jones potential. Therefore, kernel function filtration can be used to emphasize certain geometric scales [198].

$$\bar{\mathbf{M}}_{ij} = 1 - \Phi(d_{ij}, \eta_{ij}), \tag{6.4}$$

where $\Phi(d_{ij}, \eta_{ij})$ is a radial basis function and η_{ij} is a scale parameter. This filtration can be incorporated in the element specific persistent homology

$$\grave{\mathbf{M}}_{ij} = \begin{cases} d_{\infty}, \text{ if atom i or atom j} \in \mathcal{U}, \\ 1 - \Phi(d_{ij}, \eta_{ij}), \text{ otherwise.} \end{cases} \tag{6.5}$$

Additionally, one can simultaneously use two or more correlation functions characterized by different scales to generate a multiscale representation of biomolecules [140].

One form of the correlation function based filtration matrix is constructed by flexibility and rigidity index. In this case, the Lorentz function is used in Eq. (6.5)

$$\Phi(d_{ij}; \eta_{ij}, \nu) = \frac{1}{1 + \left(\frac{d_{ij}}{\eta_{ij}}\right)^{\nu}}, \tag{6.6}$$

where d_{ij} is the Euclidean distance between point i and point j and η_{ij} is a parameter controlling the scale and is related to radius of two atoms. When distance matrices based on such correlation functions are used, patterns at different spatial scales can be addressed separately by altering the scale parameter η_{ij}. Note that the rigidity index is given by [196]

$$\mu_i = \sum_j \Phi(d_{ij}; \eta_{ij}, \nu). \tag{6.7}$$

This expression is closely related to the rigidity density based volumetric filtration [201].

6.3.2.4 Electrostatic persistence

Electrostatic effects are some of the most important effects in biomolecular structure, function, and dynamics. The embedding of electrostatics in topological invariants is of particular

interest and can be very useful in describing highly charged biomolecules such as nucleic acids and their complexes. We introduce electrostatics interaction induced distance functions in Eq. (6.8) to address the electrostatic interactions among charged atoms. The abstract distance between two charged particles are rescaled according to their charges and their geometric distance, and is modeled as

$$\Phi(d_{ij}, q_i, q_j; c) = \frac{1}{1 + \exp(-cq_iq_j/d_{ij})}, \tag{6.8}$$

where d_{ij} is the distance between the two atoms, q_i and q_j are the partial charges of the two atoms, and c is a nonzero tunable parameter. c is set to a positive number if opposite charge interactions are to be addressed and is set to a negative number if like charge interactions are of interest. The form of the function is adopted from sigmoid function which is widely used as an activation function in artificial neural networks. Such function regularizes the input signal to the $[0, 1]$ interval. Other functions can be similarly used. This formulation can be extended to systems with dipole or higher order multipole approximations to electron density. The weak interactions due to long distances or neutral charges result in correlation values close to 0.5. When $c > 0$, the repulsive interaction and attractive interaction deliver the correlation values in $(0.5, 1)$ and $(0, 0.5)$ respectively. The distances induced by $\Phi(d_{ij}, q_i, q_j; c)$ are used to characterize electrostatic effects. The parameter c is rather physical but chosen to effectively spread the computed values over the $(0, 1)$ interval so that the results can be used by machine learning methods. Another simple choice of charge correlation functions is

$$\Phi(d_{ij}, \eta_{ij}, q_i, q_j) = q_iq_j \exp(-d_{ij}/\eta_{ij}).$$

However, this choice will lead to a different filtration domain. Additionally, a charge density can be constructed

$$\mu^c(\mathbf{r}) = \sum_j q_j \exp(-\|\mathbf{r} - \mathbf{r}_j\|/\eta_j), \tag{6.9}$$

where $\mathbf{r}$ is a position vector, $\|\mathbf{r} - \mathbf{r}_j\|$ is the Euclidean distance between $\mathbf{r}$ and jth atom position $\mathbf{r}_j$ and η_j is a scale parameter. Equation (6.9) can be used for electrostatic filtration as well. In this case, the filtration parameter can be the charge density value and cubical complex based filtration can be used.

6.3.3 Feature generation from topological invariants

Barcode representation of topological invariants offers a visualization of persistent homology analysis. In machine learning analysis, we convert the barcode representation of topological invariants into structured feature arrays for machine learning. To this end, we introduce several procedures to generate feature vectors from sets of barcodes. These methods are discussed below.

Counts in bins.

For a given set of atoms A, we denote its barcodes as $\mathbf{B} = \{I_\alpha\}_{\alpha \in A}$ and represent each bar by an interval $I_\alpha = [b_\alpha, d_\alpha]$, where b_α and d_α are respectively the birth and death positions on the filtration axis. The length of each bar, or the persistence of topological invariant is given by $p_\alpha = d_\alpha - b_\alpha$. To locate the position of all bars and persistences, we further split the set of barcodes on the filtration axis into a predefined collection of N bins $\mathbf{Bin} = \{\mathrm{Bin}_i\}_{i=1}^N$ with $\mathrm{Bin}_i = [l_i, r_i]$, where l_i and r_i are the left and the right positions of the ith bin. We generate features by counting the numbers of births, deaths, and persistences in each bin, which leads to three counting feature vectors, namely, counts of birth F_b^C, death

F_d^C, and persistence F_p^C,

$$
\begin{aligned}
F_{b,i}^C(\mathbf{B}) &= \|\{[b_\alpha, d_\alpha] \in \mathbf{B} | l_i \leq b_\alpha \leq r_i\}\|, 1 \leq i \leq N, \\
F_{d,i}^C(\mathbf{B}) &= \|\{[b_\alpha, d_\alpha] \in \mathbf{B} | l_i \leq d_\alpha \leq r_i\}\|, 1 \leq i \leq N, \\
F_{p,i}^C(\mathbf{B}) &= \|\{[b_\alpha, d_\alpha] \in \mathbf{B} | b_\alpha \leq r_i \text{ or } l_i \leq d_\alpha\}\|, 1 \leq i \leq N,
\end{aligned}
\tag{6.10}
$$

where $\|\cdot\|$ is number of elements in a set. Note that the above discussion should be applied to three topological dimensions, i.e., barcodes of the 0th dimension ($\mathbf{B}^0$), 1st dimension ($\mathbf{B}^1$) and 2nd dimension ($\mathbf{B}^2$). In general, this approach enables the description of bond lengths, including the length of non-covalent interactions, in biomolecules.

Barcode statistics.

Another method of feature vector generation from a set of barcodes is to extract important statistics of barcode collections such as maximum values and standard deviations. Given a set of bars $\mathbf{B} = \{[b_\alpha, d_\alpha]\}_{\alpha \in A}$, we define sets of **Birth** $= \{b_\alpha\}_{\alpha \in A}$, **Death** $= \{d_\alpha\}_{\alpha \in A}$, and **Persistence** $= \{d_\alpha - b_\alpha\}_{\alpha \in A}$. Three statistic feature vectors F_b^S, F_d^S, and F_p^S can then be generated in the sense of the statistics of the collection of barcodes. For example, F_b^S consists of avg(**Birth**), std(**Birth**), max(**Birth**), min(**Birth**), sum(**Birth**), and cnt(**Birth**), where avg$(\cdot)$ is the average value of a set of numbers, std$(\cdot)$ is the standard deviation of a set of numbers, max$(\cdot)$ and min$(\cdot)$ are maximum and minimum values in a set of numbers, sum$(\cdot)$ is the summation of elements in a set of numbers, and cnt$(\cdot)$ is the count of elements in a set. The generation of F_d^S is the same by examining the set **Death**. F_p^S contains the same information with two extra terms, the birth and death values of the longest bar. Statistics feature vectors are collected from barcodes of three topological dimensions, i.e., the 0th, 1st, and 2nd dimensions.

2D representation.

The construction of multi-dimensional persistence is an interesting topic in persistent homology. In general, it is believed that multi-dimensional persistence has better representational power for complex systems described by multiple parameters [32]. Although multidimensional persistence is hard to compute, one can compute persistence for one parameter while fixing the rest of the parameters to a sequence of fixed values. In the case where there are two parameters, a bifiltration can be done by taking turns to fix one parameter to a sequence of fixed values while computing persistence for the other parameter. For example, one can take a sequence of resolutions and compute persistence for distance with each fixed resolution. The sequence of outputs can be stacked to form a multidimensional representation [199].

Computing persistence multiple times and stacking the results is especially useful when the parameters that are not chosen to be the filtration parameter are naturally discrete with underlying orders. For example, the multi-component or element specific persistent homology will result in many persistent homology computations over different selections of atoms. These results can be ordered by the percentage of atoms used of the whole molecule or by their importance scores in classical machine learning methods. Also, multiple underlying dimensions exist in the element specific persistent homology characterization of molecules. This property enables 2D or 3D topological representation of molecules. Based on the observation that the performance of the predictor degenerates when too many element combinations are used, we order the element combinations according to their individual performance on the task using methods of ensemble of trees. Combining the dimension of spatial scale and dimension of element combinations, a 2D topological representation is obtained. Such representation is expected to work better in the case of complex geometry

such as protein-ligand complexes. With $\mathbf{E} = \{E_j\}_{j=1}^{N_E}$ denoting the collection of element combinations ordered by their individual importance scores on the task and $\mathbf{B}^k(E_i)$ being the kth dimensional barcodes obtained with atoms of element combination E_j, eight 2D representations are defined as

$$\begin{aligned}&\{F_{d,i}^C(\mathbf{B}^0(E_j)), F_{p,i}^C(\mathbf{B}^0(E_j)), F_{b,i}^C(\mathbf{B}^1(E_j)), F_{d,i}^C(\mathbf{B}^1(E_j)),\\ &F_{p,i}^C(\mathbf{B}^1(E_j)), F_{b,i}^C(\mathbf{B}^2(E_j)), F_{d,i}^C(\mathbf{B}^2(E_j)), F_{p,i}^C(\mathbf{B}^2(E_j))\}_{i=1,\cdots,N}^{j=1,\cdots,N_E},\end{aligned} \tag{6.11}$$

where $F_{\gamma,i}^C$ with $\gamma = b, d, p$ is the barcode counting rule defined in Eq. (6.10). For 0th dimensional, since all bars start from zero, there is no need for $F_{b,i}^C(\mathbf{B}^0(E_j))$. These eight 2D representations are regarded as eight channels of a 2D topological image. In protein-ligand binding analysis, 2D topological features are generated for the barcodes of a protein-ligand complex and for the differences between barcodes of the protein-ligand complex and those of the protein. Therefore, we have a total of 16 channels in a 2D image for the protein-ligand complex. This 16-channel image can be fed into the training or the prediction of convolutional neural networks.

As an example, in the characterization of protein-ligand complexes using alpha complexes, 2D features are generated from the alpha complex based on persistent homology computations of protein and protein-ligand complex. A total of 128 element combinations are considered. The $[0, 12]$Å interval is divided into 120 equal length bins, which defines the resolution of topological images. Therefore, the input feature for each sample is a $120\times128\times16$ tensor.

When there are fewer element combinations considered which can hardly form another axis, the axis of element combinations can be added into the original channels to form 1D representations that can be used in 1D CNN.

6.3.4 Machine learning algorithms

Three machine learning algorithms, including k-nearest neighbors (KNN) regression, gradient boosting trees and deep convolutional neural networks, are integrated with our topological representations to construct topological learning algorithms.

K-nearest neighbors algorithm via barcode space metrics.

One of the simplest machine learning algorithms is k-nearest neighbors (KNN) for classification or for regression. In KNN regression, for a given object, its property values is obtained by the average or the weighted average of the values of its k nearest neighbors induced by a given metric of similarity. Then, the problem becomes how to construct a metric on the dataset.

In the present work, instead of computing similarities from constructed feature vectors, the similarity between biomolecules can simply be derived from distances between sets of barcodes generated from different biomolecules. Popular barcode space metrics include the bottleneck distance [43] and more generally, the Wasserstein metrics [44, 29]. The definition of the two metrics can be found in Section 2.1.3.

The barcode space metrics can be directly used to assess the representation power of various persistent homology methods on biomolecules without being affected by the choice of machine learning models and hyperparameters. We show in the section of results that the barcode space metrics induced similarity measurement is significantly correlated to molecule functions.

Wasserstein metric measures from biomolecules can also be directly implemented in a kernel based method such as nonlinear support vector machine algorithm for classification and regression tasks. However, this aspect is not explored in the present work.

Gradient boosting trees.

Gradient boosting trees is an ensemble method which ensembles individual decision trees to achieve the capability of learning complex feature target maps and can effectively prevent overfitting by using shrinkage technique. The gradient boosting trees method is realized using the GradientBoostingRegressor module in scikit-learn software package [152] (version 0.17.1). A set of parameters found to be efficient in our previous study on the protein-ligand binding affinity prediction [24] is used uniformly unless specified. The parameters used are *n_estimators=20000, max_depth=8, learning_rate=0.005, loss='ls', subsample=0.7, max_features='sqrt'.*

Deep convolutional neural networks.

The deep convolutional neural networks in this work are implemented using Keras [40] (version 1.1.2) with Theano backend [177] (version 0.8.2).

For TopBP-DL(Complex), a widely used convolutional neural network architecture is employed beginning with convolution layers followed by dense layers. Due to the limited computation resources, parameter optimization is not performed, while most parameters are adopted from our earlier work [26]. Reasonable parameters are assigned manually. The detailed architecture is shown in Fig. 6.2. The Adam optimizer with learning rate 0.0001 is used. The loss function is the mean squared error function. The network is trained with a batch size of 16 and 150 epochs. The training data is shuffled for each epoch.

The network architecture of TopVS-DL is shown in Fig. 6.3. The Adam optimizer with learning rate set to 0.0001 is used. The loss function is binary cross-entropy. The network is trained with a batch size of 1024 and 10 epochs. The training data is shuffled for each epoch. The batch size is larger than that used in TopBP-DL due to the much larger training set in this problem. Because of the same reason, the training process converges to a small

loss very fast with only a few training steps.

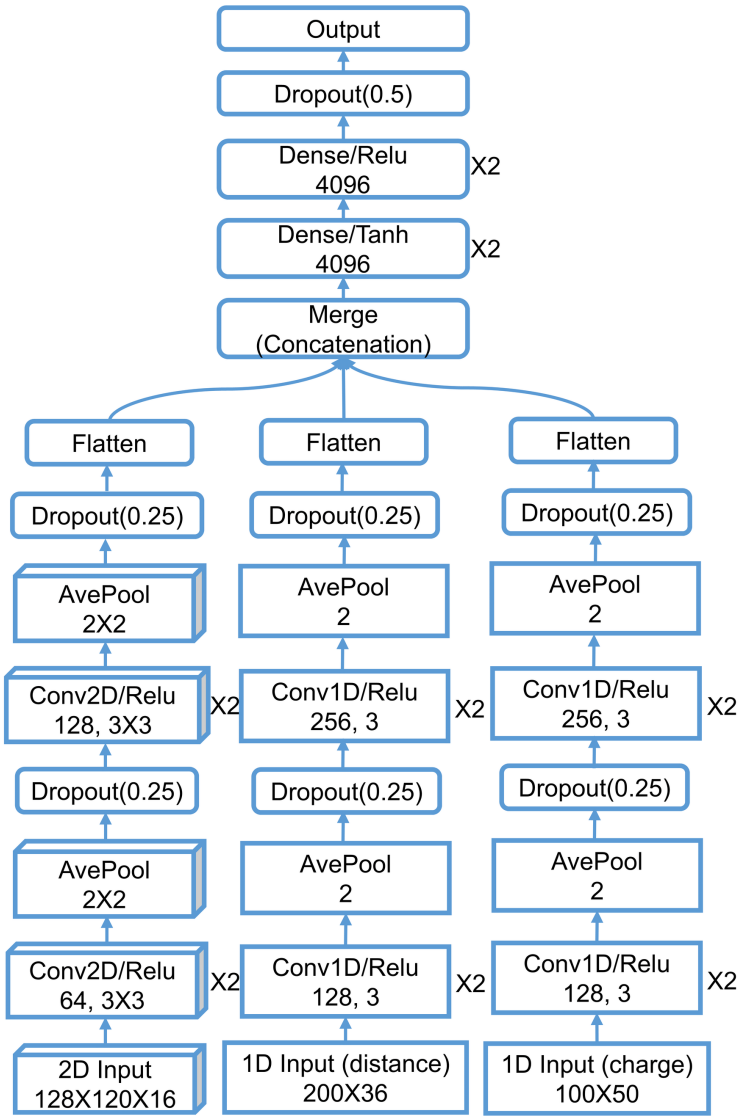

Figure 6.2: The network architecture of TopBP-DL.[20]

The structured layers are shown in boxes/rectangles with sharp corners for 2D/1D image-like content and the unstructured layers are shown in rectangles. The numbers in convolution layers mean the number of filters and filter size from left to right. The dense layers are drawn with number of neurons and activation function. The pooling size of the pooling layers and dropout rate of the dropout layers are listed. The layers that are repeated n times are marked with "$\times n$" sign on the right side of the layer.

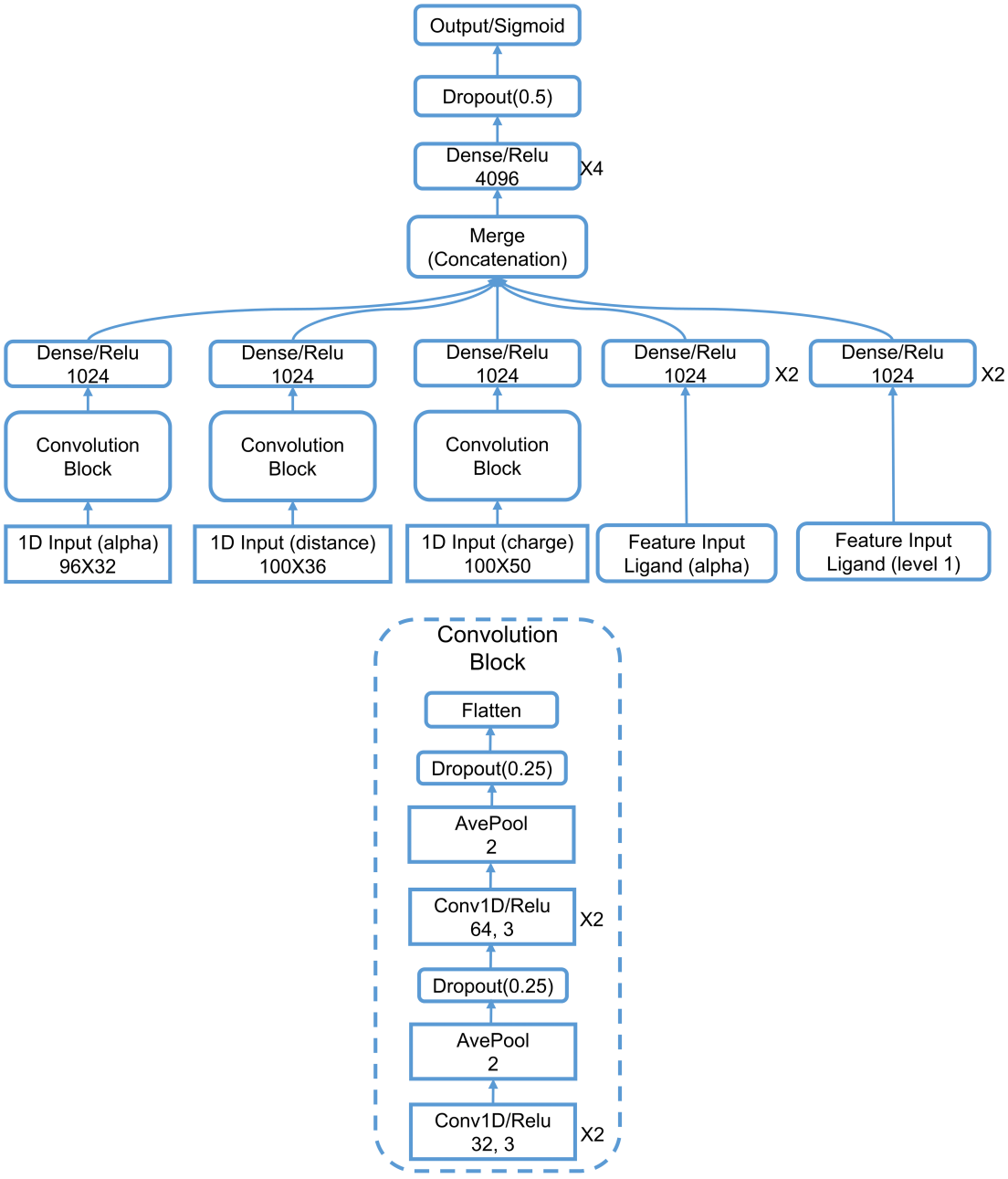

Figure 6.3: The network architecture of TopVS-DL.[20]
The 1D image-like layers are shown in sharp-corner rectangles. The numbers in convolution layers mean the number of filters and filter size from left to right. The pooling size of the pooling layers and dropout rate of the dropout layers are listed. The layers that are repeated n times are marked with "$\times n$" sign on the right side of the layer.

6.4 Results

Rational drug design and discovery have rapidly evolved into some of the most important and exciting research fields in medicine and biology. These approaches potentially have a profound impact on human health. The ultimate goal is to determine and predict whether a given drug candidate will bind to a target so as to activate or inhibit its function, which results in a therapeutic benefit to the patient. Virtual screening is an important process in rational drug design and discovery which aims to identify actives of a given target from a library of small molecules. There are mainly two types of screening techniques, ligand-based and structure-based. Ligand-based approaches depend on the similarity among small molecule candidates. Structure-based approaches try to dock a candidate molecule to the target protein and judge the candidate with the modeled binding affinity based on docking poses. Various molecular docking software packages have been developed for these purposes. Molecular docking involves both pose generation and binding affinity scoring. Currently, pose generation is quite robust while scoring power is still limited. Therefore, knowledge based rescoring methods using machine learning or deep learning approaches can improve scoring accuracy [158, 59, 4]. We also apply our topological learning method as a rescoring tool to rerank the candidates based on docking poses generated by docking software.

This section explores the representational power of the proposed persistent homology methods for the prediction of protein-ligand binding affinities and the discrimination of actives and non-actives for protein targets. To this end, we use the present method to investigate three types of problems. First, we develop topological learning models for ligand based protein-ligand binding affinity predictions. This problem is designed to examine the representability of the proposed topological methods for small molecules. Then, we develop

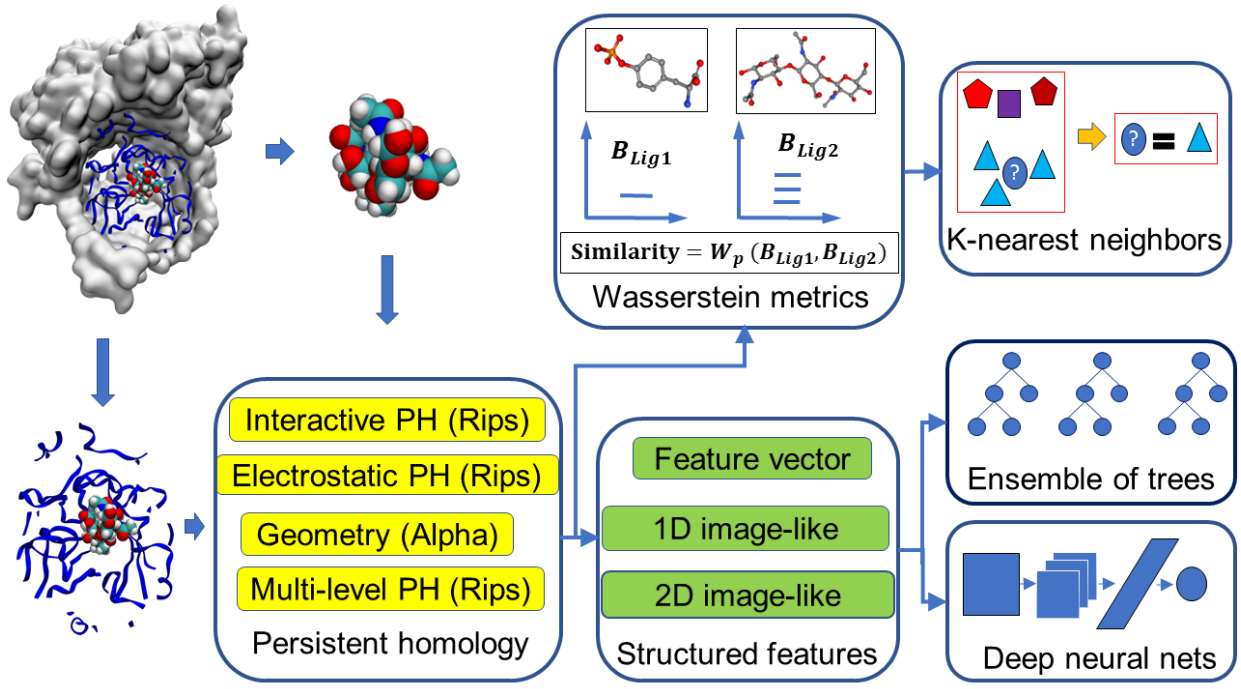

Figure 6.4: An illustration of the topology based machine learning algorithms used in scoring and virtual screening.[20]

topological learning models for protein-ligand complex based binding affinity prediction. This problem enables us to understand the capability of the proposed topological learning methods for dealing with protein-ligand complexes. Finally, we examine the structure-based classification of active ligands and decoys which are highly possible to be non-actives, i.e., structure-based virtual screening (VS). The optimal selection of features and methods are determined by studying the first two applications and this finding leads to the main application studied in this work, the topological structure-based virtual screening. Computational algorithms used in this study are illustrated in Fig. 6.4.

6.4.1 Ligand based protein-ligand binding affinity prediction

In this section, we address the representation of small molecules by element specific persistent homology, especially the proposed multi-level persistent homology designed for small molecules.

Data set

To assess the representational ability of the present persistent homology algorithms on small molecules, we use a high quality data set of 1322 protein-ligand complexes with binding affinity data involving 7 protein clusters introduced earlier (denoted as S1322) [187]. It is a subset of the PDBBind v2015 refined set and its detail is given in the Supplementary material 1 of Ref. [187]. We consider a ligand based approach to predict the binding affinities of protein-ligand complexes in various protein clusters. As such, only the ligand information is used in our topological analysis. The ligand structures are taken from PDBBind database without modification. Numbers of ligands in protein clusters range from 94 to 333.

Models and performance

Two models, i.e., TopBP-KNN(Ligand) and TopBP-ML(Ligand), are constructed. TopBP-KNN(Ligand) is used to directly assess the representation power of persistent homology for small molecules and TopBP-ML(Ligand) is the final practical model. The results are shown in Table 6.1. All the gradient boosting trees models take the setup described in *Section Methods/Machine learning algorithms/Gradient boosting trees*.

In TopBP-ML(Ligand), we process the geometry, the shape, and the covalent bond information of the small molecules using alpha complex, and the non-covalent intramolecular interactions using multi-level persistent homology with Rips complex. The features used are A-B012-E-S-GBT and R-B012-M1-S-GBT as described in *Section Discussion/Ligand based*

protein-ligand binding affinity prediction. Gradient boosting trees method is used.

In TopBP-KNN(Ligand), we represent the small molecules with a collection of sets of barcodes from element specific persistent homology calculations. Wasserstein distance with $p = 2$ is applied to measure similarities between two sets of barcodes. The similarity between each pair of small molecules is then measured by taking the average of the Wasserstein distances between all sets of considered barcodes. K-nearest-neighbor (KNN) regression is then applied to the measured similarity. In detail, the 6 sets of barcodes considered are, R-B0-E-KNN, R-B1-E-KNN, R-B2-E-KNN, R-B0-M1-KNN, R-B1-M1-KNN, and R-B2-M1-KNN as described in *Section Discussion/Ligand based protein-ligand binding affinity prediction*. Leave-one-out validation within each protein cluster with $k = 3$ is used for this model.

Methods	TopBP-KNN(Ligand)	TopBP-ML(Ligand) (5-fold)	FFT-BP (5-fold) [187]
CL 1 (333)	0.698(1.66)	0.713(1.60)	(1.93)
CL 2 (264)	0.817(1.28)	0.843(1.15)	(1.32)
CL 3 (219)	0.620(1.68)	0.693(1.51)	(2.01)
CL 4 (156)	0.645(1.41)	0.670(1.35)	(1.61)
CL 5 (134)	0.756(1.68)	0.831(1.34)	(2.02)
CL 6 (122)	0.658(1.68)	0.698(1.56)	(2.06)
CL 7 (94)	0.739(1.31)	0.737(1.26)	(1.71)
Average	0.705(1.49)	0.741(1.40)	(1.81)

Table 6.1: Pearson correlation coefficients (RMSE in kcal/mol) of ligand based topological model on the S1322 dataset.
The numbers in the first row show the number of entries in each protein cluster. The performance is reported as Pearson correlation coefficient (root mean squared error in kcal/mol). The median performance of 20 random 5-fold cross validation results is reported for TopBP-ML(Ligand). The results reported for TopBP-KNN(Ligand) are obtained by leave-one-out validation within each protein cluster with $k = 3$ for the KNN model.

In Table 6.1, FFT-BP 5-fold cross validation results were obtained based on multiple additive regression trees and a set of physical descriptors, including geometry, charge, electrostatic interactions, and van der Waals interactions for S1322 set[187]. Since multiple additive regression trees is also an implementation of the GBT used in the present work, it is

appropriate to compare the FFT-BP results with the GBT results in this work to assess representation power of topological features. It is interesting to note that judged by RMSE, both sets of current topological descriptors have more predictive power than the physical descriptors built on protein-ligand complexes constructed in our earlier work [187]. These physical descriptors were constructed from sophisticated surface areas, molecular volumes, van der Waals interactions, charges computed by quantum mechanics, and Poisson-Boltzmann theory based electrostatics [187]. The success of topological descriptors implies the existence of an alternative and potentially more powerful description of the complex biomolecular world.

6.4.2 Complex based protein-ligand binding affinity prediction

In this section, we focus on topological representations of protein-ligand complexes.

Data sets

The PDBBind database provides a comprehensive collection of structures of protein-ligand complexes and their binding affinity data [38, 119]. The original experimental data inProtein Data Bank (PDB) [17] are selected to PDBBind database based on certain quality requirements and are curated for applications. As shown in Table 6.2, this database is expanding on a yearly basis. It has become a common resource for benchmarking computational methods and algorithms for protein-ligand binding analysis and drug design. Popular data sets include version 2007 (v2007), v2013, and v2015. Among them, v2013 core set and v2015 core set are identical. A large number of scoring functions has been tested on these data sets. The latest version, v2016, has an enlarged core set, which contains 290 protein-ligand complexes from 58 protein families. Therefore, this test set should be relatively easier than v2015 core set, whose 195 complexes involve 65 protein families. The core sets are constructed by choosing 3 samples with median, maximum, and minimum binding affinity from

each protein family for v2007, v2013, and v2015 sets. The core set for v2016 was constructed similarly but with 5 samples from each protein family.

Version	Refined set	Training set	Core set (test set)	Protein families
v2007	1300	1105	195	65
v2013	2959	2764	195	65
v2015	3706	3511	195	65
v2016	4057	3767	290	58

Table 6.2: Description of the PDBBind datasets.
Number of complexes or number of protein families in PDBBind data sets used in the present binding affinity prediction. Here training sets are set to the corresponding refined sets, excluding the complexes in the corresponding test sets (i.e., core sets). Protein families refer to those in the corresponding core sets.

Model and performance

Two models TopBP-ML(Complex) and TopBP-DL(Complex) are introduced. The results are shown in Table 6.3. All the gradient boosting trees models take the setup described in *Section Methods/Machine learning algorithms/Gradient boosting trees*.

Core set predictions					
Methods	v2007	v2013	v2015	v2016	Average
TopBP(Complex)	0.827 (1.93)	0.808 (1.95)	0.812 (1.92)	0.861 (1.65)	0.827 (1.86)
TopBP-ML(Complex)	0.818 (2.01)	0.804 (2.00)	0.797 (1.99)	0.848 (1.74)	0.817 (1.94)
TopBP-DL(Complex)	0.806 (1.95)	0.781 (1.98)	0.799 (1.91)	0.848 (1.64)	0.809 (1.87)
RF::VinaElem	0.803 (1.94) [115]	0.752 (2.03) [116]	-	-	-
RI-Score[140]	0.803 (1.99)	-	0.762 (2.05)	0.815 (1.85)	-
Refined set 5-fold cross validations					
Methods	v2007	v2013	v2015	v2016	Average
TopBP-ML(Complex)	0.752 (1.95)	0.768 (1.75)	0.781 (1.71)	0.785 (1.71)	0.771 (1.78)
RI-Score[140]	-	-	-	0.747 (1.83)	-

Table 6.3: Pearson correlation coefficients (RMSE in kcal/mol) of different protein-ligand complex based approaches on PDBBind datasets.
Pearson correlation coefficients with RMSE (kcal/mol) in parentheses for predictions by different methods are listed. For the tests on core sets, the models are trained with the corresponding refined set minus the core set. Five-fold cross validation is done on refined sets. Results of TopBP-ML(Complex) are the medians of 50 repeated runs. For TopBP-DL(Complex), 100 independent models are generated at first. A consensus model is built by randomly choosing 50 models out of the 100, and this process is repeated 1000 times with the median reported. TopBP(Complex) is a consensus model combining TopBP-ML(Complex) and TopBP-DL(Complex). Each time, 50 single deep learning models are randomly selected to form TopBP-DL(Complex) and a TopBP-ML(Complex) model is randomly selected. The average of the two is taken as the output for TopBP(Complex). This process is repeated 1000 times with the median reported.

In TopBP-ML(Complex), alpha complex is used to describe the arrangement of carbon and heavy atom networks, while Rips complex with different distance matrices is used to describe the protein-ligand interactions from the perspective of interaction distances and strength of electrostatics interactions. In detail, the features used are R-B0-I-C, R-B0-CI-S, A-B12-E-S as described in *Section Discussion/Complex based protein-ligand binding affinity prediction*, and those used in TopBP-ML(Ligand).

With the idea that a sequence of element combinations ordered by their importance in gradient boosting trees models can make an extra dimension of the description, we build a 2D convolutional neural network with one spatial dimension and one dimension of element combination. We combine this 2D CNN with a 1D CNN with the pairwise interaction inputs. For the construction of 2D input, the reader is referred to *Section Feature generation from topological invariants* . The 1D image-like inputs consists of two parts both generated by the counts in bins method described in *Section Feature generation from topological invariants* . For the 0th dimensional barcodes from interactive persistent homology of the 36 pairs of atom types ({C,N,O,S} from protein and {C,N,O,S,P,F,Cl,Br,I} from ligand), the interval $[0, 50]$ Å is divided into equal length subintervals of length 0.25 Å. For the 0th dimensional barcodes from interactive persistent homology for electrostatics of the 50 pairs of atom types ({C,N,O,S,H} from protein and {C,N,O,S,P,F,Cl,Br,I,H} from ligand), the parameter interval of $[0, 1]$ is divided into equal length subintervals of length 0.01. These two 1D image-like features have sizes 200×36 and 100×50. The network architecture is given in *Section Methods/Machine learning algorithms/Deep convolutional neural networks*.

The final model TopBP(Complex) takes the average of TopBP-ML(Complex) and TopBP-DL(Complex) with the assumption that the errors made by the two approaches are only partially correlated and thus averaging over them may cancel part of the errors. As a result,

TopBP(Complex) delivers the best prediction performance on all four testing sets.

6.4.3 Structure-based virtual screening

In this section, we examine the performance of the proposed method for the main application in this paper, which is structure-based virtual screening which involves protein-compound complexes obtained by attempting to dock the candidates to the target proteins. The dataset is much larger than the two applications on protein-ligand binding affinity prediction which makes parameter tuning very time consuming. Therefore, the best performing procedures in ligand-based binding affinity prediction and protein-ligand-complex-based binding affinity prediction are applied in this virtual screening application.

DUD data set

The directory of useful decoys (DUD) [93, 135] is used to benchmark our topological approach for virtual screening. The DUD data set contains 40 protein targets from six classes, i.e., nuclear hormone receptors, kinases, serine proteases, metalloenzymes, folate enzymes, and other enzymes. A total of about 3000 active ligands were identified from literature. The number of ligands for each target ranges from tens to hundreds. At most 36 decoys were constructed for each ligand, from the ZINC database of commercially available compounds [95]. At the first step, the ZINC database of 3.5 million compounds was reduced to a database of 1.5 million compounds with similarity less than 0.9 to the ligands. The similarity was measured by Tanimoto coefficient on CACTVS type 2 fingerprints. The decoys were selected so that they possess similar physical properties to the ligands but have dissimilar molecular topology (topology in the sense of chemistry, not mathematical topology). A total of 32 physical properties were used including molecular weight, partition coefficient, and number of hydrogen bonding groups. This results in a total of about 100000

compounds. A discrepancy between calculated partial charges for the ligand and decoy sets was reported for the original release 2 of DUD datasets, which makes it trivial for virtual screening methods to distinguish between the two categories using those charges [5]. In this work, we use the data with recalculated Gasteiger charges for both ligand and decoy sets given by Armstrong *et al.* [5] in AutoDock Vina and our electrostatic persistence.

Data processing

In structure-based virtual screening, the possible complex structures of the target protein and the small molecule candidate are required. For the DUD dataset, the structures of the 40 protein targets, the ligands, and the decoys are given, and we generate the protein-compound complexes by using docking software. To this end, we first add missing atoms to the proteins by using the profix utility in Jackal software package [203]. The receptors and ligands or decoys are prepared using the scripts prepare_receptor4.py and prepare_ligand4.py provided by the AutoDockTools module in MGLTools package (version 1.5.6) [130]. The bounding box of the binding site is defined as a cube with edge size equal to 27 Å, centered at the geometric center of the crystal ligand. AutoDock Vina (version 1.1.2) [179] is used to dock the ligands or decoys to the receptors. The option exhaustiveness is set to 16 and all the other parameters are set to their default values. In each docking experiment, the pose having the lowest binding free energy reported by AutoDock Vina, is used by the reranking models.

Evaluation

Two measurements, the enrichment factor (EF) and the area under the receiver operating characteristic curve (AUC), are used to evaluate each method's ability of discriminating

actives from decoys. The AUC is defined as

$$\mathrm{AUC} = 1 - \frac{1}{N_a} \sum_{i=1}^{N_a} \frac{N_d^i}{N_d}, \tag{6.12}$$

where N_a is the number of active ligands, N_d is the total number of decoys, and N_d^i is the number of decoys that are higher ranked than the ith ligand [158]. An AUC value of 0.5 is the expected value of a random selection, whereas a perfect prediction results in an AUC of 1. The EF at $x\%$ denoted by $\mathrm{EF}_{x\%}$ evaluates the quality of the set of top $x\%$ ranked compounds, by comparing the percentage of actives in the top $x\%$ ranked compounds to the percentage of actives in the entire compound set. It is defined as

$$\mathrm{EF}_{x\%} = \frac{N_a^{x\%}}{N^{x\%}} \cdot \frac{N}{N_a}, \tag{6.13}$$

where $N_a^{x\%}$ is the number of active ligands in the top $x\%$ ranked compounds, $N^{x\%}$ is the number of top $x\%$ ranked compounds, N is the total number of compounds, and N_a is the total number of active ligands.

To evaluate the performance of various methods on the DUD data set, the entries associated with one protein target are used as the test set each time [158]. For the selection of the training set of a given protein target, we follow a procedure given in the literature [93], where the entries associated to the rest of the proteins, excluding those that are within the same class of the testing protein and those that have reported positive cross-enrichment with the testing protein, are taken as the training set. The 40 proteins are split into 6 classes and the excluded list of training data for each protein follows [4].

Topology based machine learning models

Our topology based machine learning model, called *TopVS-ML*, relies on manually constructed features and utilizes ensemble of trees methods. For the complex with the small molecules (i.e., ligands and decoys) docked to the receptor, features R-B0-I-BP, R-B0-CI-S, and A-B12-E-S are used (see *Section Discussion/Complex based protein-ligand binding affinity prediction*), whereas features R-B012-M1-S and A-B012-E-S (see *Section Discussion/Ligand based protein-ligand binding affinity prediction*) are used for the small molecules. The gradient boosting trees method, random forest method, and extra trees method are employed as voters. The averaged probabilities output by the three methods are used for the classifier to decide the class of the testing samples. The modules *GradientBoostingClassifier*, *RandomForestClassifier*, and *ExtraTreesClassifier* in the scikit-learn package [152] (version 0.17.1) are used. The parameters for the three modules are listed in Table 6.4. TopVS-ML achieves a performance of AUC= 0.83, $EF_{2\%} = 8.6$, $EF_{20\%} = 3.4$. These values are the median values of 10 repeated experiments.

Method	Parameters
GBT	n=2000, s=0.5, cw=100:1, lr=0.01, mf=sqrt
RF	n=2000, cw=balanced_subsample
ET	n=2000, cw=balanced_subsample

Table 6.4: Parameters used in machine learning.
The parameters used for the ensemble of trees methods while the other parameters are set to default. GBT: gradient boosting trees. RF: random forest. ET: extra trees. n: n_estimators. s: subsample. cw: class_weight. lr: learning_rate. mf: max_feature.

Topology based deep learning model

Our topology based deep learning model, called *TopVS-DL*, relies on 1D image-like inputs for protein-compound complexes and manually constructed features for the compounds. The 2D representation used in binding affinity problem is not used here due to the intractable

data size. The manually constructed features for the compounds are R-B012-M1-S and A-B012-E-S as described in *Section Discussion/Ligand based protein-ligand binding affinity prediction*. The 1D image-like inputs consisted of three parts are all generated by the counts in bins method described in *Section Feature generation from topological invariants* . (1) For the 0th dimensional barcodes from interactive persistent homology of the 36 pairs of atom types ({C, N, O, S} from protein and {C, N, O, S, P, F, Cl, Br, I} from ligand), the interval $[0, 25]$ Å is divided into equal length subintervals of length 0.25 Å. The barcodes used here are identical to the barcodes in feature R-B0-I-BP. This results in a 1D image-like feature with size 100×36. (2) For the 0th dimensional barcodes from interactive persistent homology for electrostatics of the 50 pairs of atom types ({C, N, O, S, H} from protein and {C, N, O, S, P, F, Cl, Br, I, H} from ligand), the parameter interval of $[0, 1]$ is divided into equal length subintervals of length 0.01. The barcodes used are identical to the barcodes in feature R-B0-CI-S. This results in a 1D image-like feature with size 100×50. (3) Alpha complex based persistent homology is applied to all carbon atoms and all heavy atoms. The computation is done on the complex as well as only the protein with a cutoff distance of 12 Å from the ligands. The interval $[0, 12]$ Å is divided into equal length subintervals of length 0.125 Å. Counts in bins method is applied to the 0th, 1st, and 2nd dimensional barcodes. The features are generated for persistent homology computation of the complex and the protein. The features for the complex and the difference between the features for complex and protein are finally used. This results in a 1D image-like feature of size 96×32. The detailed network architecture is listed in *Section Methods/Machine learning algorithms/Deep convolutional neural networks*. A consensus model is constructed by taking the average over 25 single models trained independently. TopVS-DL achieves a performance of AUC= 0.81, $\mathrm{EF}_{2\%} = 9.1$, $\mathrm{EF}_{20\%} = 3.2$.

The final model

Same as the idea of taking the average output of different ensemble of trees models as the final output in TopVS-ML, we add TopVS-DL as another voter to TopVS-ML to construct a final model, called *TopVS*. Such consensus approach takes the average over different models with the hope that different models make partially uncorrelated errors which are possible to cancel out when averaged. The performance on each of 40 protein targets is reported in Table 6.5. We have also generated virtual screening results of AutoDock Vina (ADV) based on the computed binding free energy by ADV and compared them with those of the present TopVS in terms of enrichment factors and the areas under the receiver operating characteristic curve (AUC). A comparison of average AUC with those from a large number of methods is given in Table 6.6.

Target	ADV			TopVS		
	$EF_{2\%}$	$EF_{20\%}$	AUC	$EF_{2\%}$	$EF_{20\%}$	AUC
ACE	4.1	1.4	0.42	5.1	3.1	**0.81**
AChE	4.7	2.8	**0.67**	1.4	1.9	0.65
ADA	0.0	0.4	0.49	7.8	4.5	**0.90**
ALR2	2.0	2.7	**0.74**	4.9	1.5	0.68
AmpC	2.4	0.2	0.34	0.0	1.0	**0.58**
AR	17.0	3.8	0.81	20.1	4.2	**0.90**
CDK2	9.0	2.4	0.64	7.6	4.1	**0.88**
COMT	13.1	1.4	0.56	17.4	2.9	**0.73**
COX1	9.9	2.8	0.76	11.8	3.6	**0.86**
COX2	20.7	3.9	0.86	23.3	4.9	**0.97**
DHFR	6.4	2.8	0.82	12.6	4.7	**0.96**
EGFr	3.4	1.6	0.63	16.4	4.8	**0.95**
$ER_{agonist}$	17.8	3.3	**0.84**	10.0	2.8	0.81
$ER_{antagonist}$	10.2	2.3	0.70	1.3	2.8	**0.83**
FGFr1	0.4	0.8	0.44	15.1	4.8	**0.95**
FXa	1.0	1.3	0.63	2.1	4.4	**0.89**
GART	0.0	1.9	**0.75**	2.6	0.7	0.48
GPB	0.0	0.9	0.48	1.4	1.5	**0.66**
GR	5.7	1.2	0.57	1.3	3.4	**0.84**
HIVPR	5.6	2.6	0.74	8.9	4.4	**0.91**
HIVRT	8.2	1.9	0.64	11.7	4.0	**0.88**
HMGR	0.0	0.9	0.53	14.4	5.0	**0.96**
HSP90	0.0	0.9	0.64	9.6	4.5	**0.93**
InhA	13.4	1.9	0.56	22.7	4.5	**0.95**
MR	16.7	4.0	0.82	0.0	4.3	**0.87**
NA	0.0	0.3	0.37	1.5	3.8	**0.87**
P38 MAP	1.4	1.7	0.59	18.4	4.5	**0.94**
PARP	4.2	2.7	0.71	0.0	1.7	0.71
PDE5	8.0	1.9	0.61	6.9	3.4	**0.86**
PDGFrb	3.5	0.5	0.32	26.5	4.9	**0.97**
PNP	0.0	0.7	0.59	7.9	4.3	**0.89**
PPARg	17.7	3.4	**0.82**	0.6	1.8	0.72
PR	1.9	1.1	0.52	9.4	4.1	**0.91**
RXRa	28.2	4.8	**0.95**	12.8	3.2	0.83
SAHH	10.4	3.0	0.80	4.5	3.9	**0.84**
SRC	5.6	2.3	0.71	24.6	4.9	**0.98**
thrombin	8.3	2.6	0.72	4.1	2.4	**0.79**
TK	0.0	0.9	0.56	6.9	2.5	**0.65**
trypsin	3.1	1.9	0.58	0.0	2.0	**0.78**
VEGFr2	10.2	2.2	0.63	24.9	4.7	**0.96**
Average	6.9	2.0	0.64	9.5	3.5	**0.84**

Table 6.5: Performance on each protein in DUD dataset.
The median results of 10 repeated runs with different random seeds (for the TopVS-ML part) are reported. The best AUC in each row is marked in bold. The left block of AutoDock Vina (ADV) results are acquired from the ADV runs with the binding free energy reported by ADV.

Method	AUC	Ref.
TopVS	0.84	
DeepVS-ADV	0.81	[158]
ICM[a]	0.79	[137]
NNScore1-ADV[b]	0.78	[59]
Glide SP[a]	0.77	[47]
DDFA-ALL	0.77	[4]
DDFA-RL	0.76	[4]
NNScore2-ADV[b]	0.76	[59]
DDFA-ADV	0.75	[4]
DeepVS-Dock	0.74	[158]
DDFA-AD4	0.74	[4]
Glide HTVS[b]	0.73	[59]
Surflex[a]	0.72	[47]
Glide HTVS	0.72	[47]
ICM	0.71	[137]
RAW-ALL	0.70	[4]
AutoDock Vina[b]	0.70	[59]
Surflex	0.66	[47]
Rosetta Ligand	0.65	[4]
AutoDock Vina	0.64	[4]
ICM	0.63	[47]
FlexX	0.61	[47]
Autodock4.2	0.60	[4]
PhDOCK	0.59	[47]
Dock4.0	0.55	[47]

Table 6.6: AUC comparison of different methods on DUD dataset.
[a]Tuned by expert knowledge. [b]Determined using a different data set of decoys.

6.5 Discussion

6.5.1 Ligand based protein-ligand binding affinity prediction

We conduct several experiments on ligand based protein-ligand binding affinity prediction in this section which leads to the final models. To examine the strength and weakness of different sets of features and models, we first show a statistics fact of the S1322 data set of 7 protein clusters in Fig. 6.5. The details of the S1322 data set is given in *Section Results/Ligand based protein-ligand binding affinity prediction*. All the gradient boosting trees models take the setup described in *Section Methods/Machine learning algorithms/Gradient boosting trees*.

Feature vectors for gradient boosting trees.

In this test, Rips complex based and alpha complex based persistent homology computations up to 2nd dimension are performed for a variety of atom collections with different element types using the Euclidean metric and multi-level distance defined in Eq. (6.1). Two types of features are generated and are denoted by F^C, which is a combination of F_b^C, F_d^C, and F_p^C, and F^S, which is a combination of F_b^S, F_d^S, and F_p^S. The construction of features F^C and F^S are described in *Section Feature generation from topological invariants* . For sets of the 0th dimensional bars, only F_d^C and F_d^S are computed. In each protein cluster, 10-fold or 5-fold cross validation is repeated 20 times for each subset of feature vectors depending on selected element type. The median Pearson correlation coefficients and the root-mean-square error (RMSE) in kcal/mol are reported. For Rips complex, both level 0 computation with distance matrix $\mathbf{M}$ and level 1 computation with distance matrix $\widetilde{\mathbf{M}}^1$ as defined in Eq. (6.2) are performed.

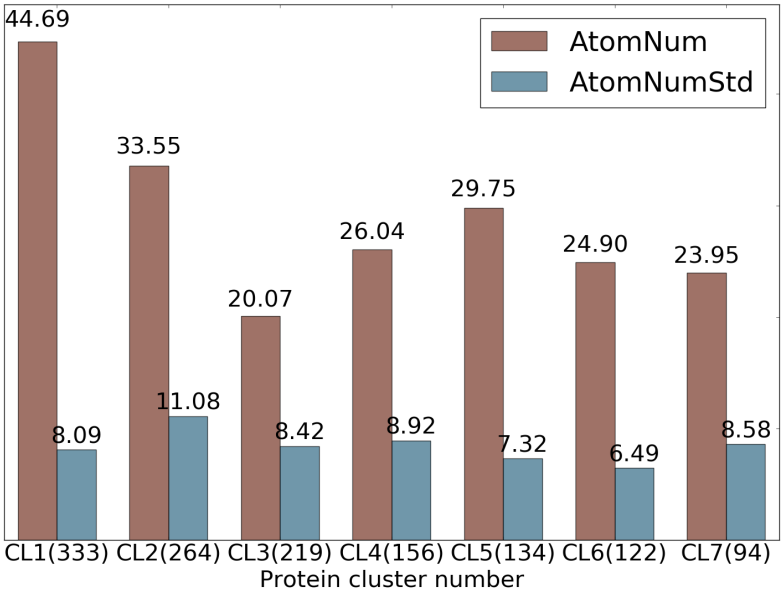

Figure 6.5: Statistics of ligands in 7 protein clusters in S1322 dataset.[20]
The average numbers of heavy atoms of a ligand in each protein cluster are shown in red and the standard deviations of number of heavy atoms across each protein cluster are shown in blue. The number of ligands in each cluster is given in parentheses.

Barcode space metrics for k-nearest neighbor regression.

The barcodes generated using Rips complex with distance matrices $\mathbf{M}$ and $\widetilde{\mathbf{M}}^1$ are collected and the distance between each pair of sets of barcodes are measured using the Wasserstein metric d^2. Leave-one-out prediction for every sample is performed with k-nearest neighbor regression with $k = 3$ within each protein cluster based on the Wasserstein metric. The performance of the best performing and the worst performing protein clusters is shown in Fig. 6.6. The better the performance, the closer the lines are to the semicircle.

The experiments done for this section are summarized in Table 6.7.

Performance of multi-component persistent homology.

It can be noticed from Table 6.8 that topological features generated from barcode statistics typically outperform those created from counts in bins. R-B012-E-S-GBT and R-B012-

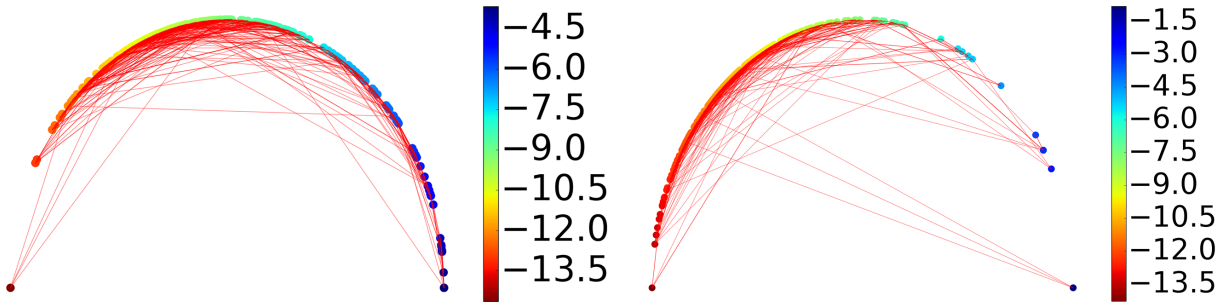

Figure 6.6: An illustration of similarities between ligands measured by their barcode space Wasserstein distances.[20]
Ligands are ordered according to their binding affinities and are represented as dots on the semicircle. Specifically, a sample of binding free energy x is plotted at the angle $\theta = \pi(E_{max} - x)/(E_{max} - E_{min})$ where E_{min} and E_{max} are the lowest and the highest energy in the dataset. Each dot is connected with two nearest neighbors based on their barcode space Wasserstein distances. An optimal prediction would be achieved if lines stay close to the semicircle. The majority of the connections stay near the boundary to the upper half sphere demonstrating that barcode space metric based Wasserstein distance measurement reflects the similarity in function, i.e., the binding affinity in this case. The protein clusters with the best and the worst performance are shown. Left: Protein cluster 2. Right: Protein cluster 3.

M1-S-GBT perform similarly in the majority of the protein clusters whilst R-B012-M1-S-GBT which is based on $\widetilde{\mathbf{M}}^1$ significantly outperforms R-B012-E-S-GBT which is based on Euclidean distance in protein cluster 3 and 6. To assess in what circumstances does the multi-level persistent homology improve the original persistent homology characterization of small molecules, we analyze the statistics of the size of ligands in Fig. 6.5. It turns out that protein cluster 3 has the smallest average number of heavy atoms and protein cluster 6 has the smallest standard deviation of the number of heavy atoms. This observation partially answers the question that in the cases where the small molecules are relatively simple and are relatively of similar size, multi-level persistent homology is able to enrich the characterization of the small molecules which further improves the robustness of the model. Such enrichment or improvement over the original persistent homology approach is mainly realized in higher dimensional barcodes, i.e. the 1st and 2nd dimensions. In Table 6.8, the results with ID through 7 to 12 confirm that the 0th dimensional features from computation with $\widetilde{\mathbf{M}}^1$ are

inferior to the results with Euclidean distance whilst the 1st and 2nd dimensional features based on $\widetilde{\mathbf{M}}^1$ outperforms the best result with Euclidean distance in most cases.

It is interesting to note that although Wasserstein metric based KNN methods are not as accurate as GBT approaches, the consensus result obtained by averaging over various predictions with Wasserstein metric on different sets of barcodes is quite accurate.

Robustness of topological learning models.

Certain elements such as Br are very rare in the data sets studied in this work. Considering only the elements of high occurrence will not hurt the performance on the validations performed. However, omitting the low occurrence elements will sacrifice the capability of the model to handle new data in which such elements play an important role. Therefore, we decide to keep the rare elements that result in a large number of features and redundancy in features. For example, the element combinations CBrH and CH will probably deliver the same performance for most of the samples in the data sets studied in this work. To test whether this redundancy causes degenerated results of the model, the features of one element combination is added to the model at a step and the model is validated with an accumulation of the added features at each step. The performance of the model is measured with Pearson correlation coefficient and is plotted against number of element combinations involved in Fig. 6.7 . For most cases in Fig. 6.7 , the model is robust against the inclusion of more element combinations.

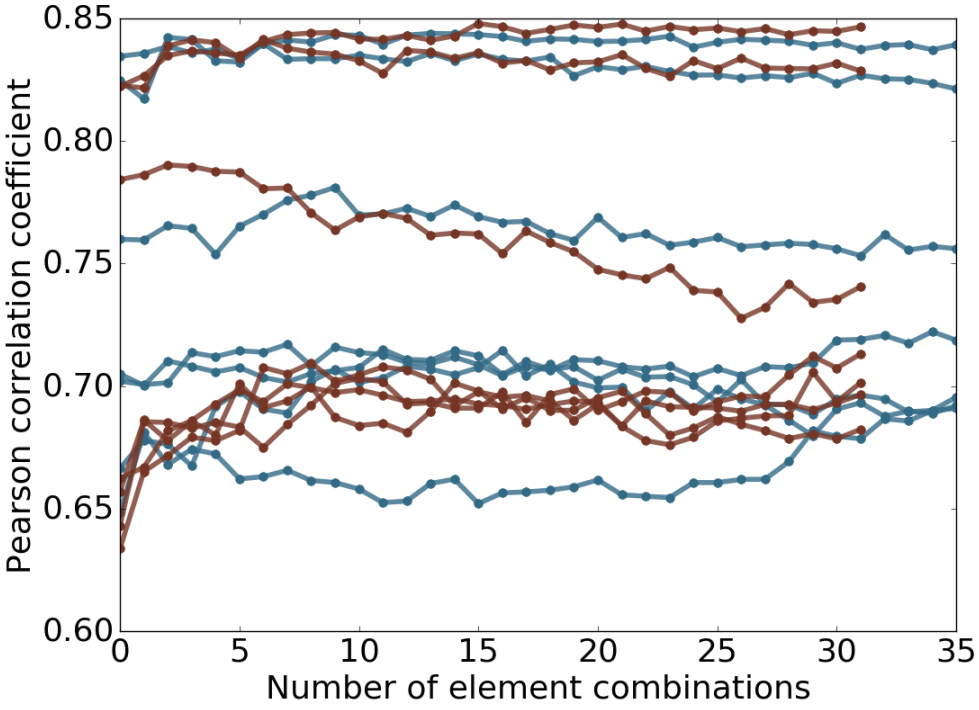

Figure 6.7: Plot of performance against number of element combinations used.[20]
The topological learning model performance against the number of element combinations involved in feature construction for 7 protein clusters in S1322. The horizontal axis corresponds to the number of element combinations used for the features. From left to right, one extra element combination is added at a step. The features are then used in gradient boosting trees method to test if the model is robust against redundant information. The results related to alpha complex are marked in red and Rips complex in blue. The median Pearson correlation coefficient between predicted and experimental results is reported of 10-fold cross-validation within each protein cluster repeated 20 times are reported.

Experiment	Description
A-B012-E-C-GBT	The barcodes are generated using alpha complex on different sets of atoms based on different element combinations. The features are constructed using the 0th, 1st, and 2nd dimensional barcodes following the *counts in bins* method with bins equally dividing the interval $[0, 5]$. Here 32 different element combinations are considered, including {C, N, O, S, CN, CO, CS, NO, NS, OS, CNO, CNS, COS, NOS, CNOS, CNOSPFClBrI, H, CH, NH, OH, SH, CNH, COH, CSH, NOH, NSH, OSH, CNOH, CNSH, COSH, NOSH, CNOSH, CNOSPFClBrIH}. Gradient boosting trees (GBT) with the structured feature matrix are used for this computation.
A-B012-E-S-GBT	The barcodes same as those used in A-B012-E-C-GBT are used. Instead of *counts in bins* , the *Barcode statistics* method is used to generate features.
A-B012-E-SS-GBT	The barcodes same as those used in A-B012-E-C-GBT are used. The *persistence diagram slice and statistics* method is used to generate features. A uniform set of bins by dividing the interval [0,5] into 10 equal length bins is used to slice birth, death, and persistence values.
R-B012-E-S-GBT	Barcodes are generated using Rips complex with Euclidean distances. The features are generated following the *barcode statistics* method. Here 36 element combinations are considered, i.e., {C, N, O, S, CN, CO, CS, NO, NS, OS, CNO, CNS, COS, NOS, CNOS, CNOSPFClBrI, H, CH, NH, OH, SH, CNH, COH, CSH, NOH, NSH, OSH, CNOH, CNSH, COSH, NOSH, CNOSH, CNOSPFClBrIH, CCl, CClH, CBr, CBrH}.
R-B012-M1-S-GBT	The result is obtained with the same setup as R-B012-E-S-GBT except that the first level enrichment distance matrix $\widetilde{\mathbf{M}}^1$ is used instead of Euclidean distance.
R-Bn-E-KNN	The nth dimensional barcodes from Rips complex computation with Euclidean distance are used. K-nearest neighbor (KNN) regression is performed with Wasserstein metric d^2. The leave-one-out validation is performed individually with each element combination and the average prediction of these element combinations is taken as the output result. The element combinations considered are {CNOS, CNOSPFClBrI, NOH, CNO, CNOSPFClBrIH}. These combinations are selected based on their performance in the gradient boosting trees experiments.
R-Bn-M1-KNN	The result is obtained with the same setup as R-Bn-E-KNN except that the distance matrix $\widetilde{\mathbf{M}}^n$ is used instead of Euclidean distance.

Table 6.7: Experiments for ligand-based protein-ligand binding affinity prediction of 7 protein clusters and 1322 protein-ligand complexes.

ID	Experiments	CL 1 (333) CL 5 (134)	CL 2 (264) CL 6 (122)	CL 3 (219) CL 7 (94)	CL 4 (156) Average
1	A-B012-E-C-GBT	0.695(1.63) **0.840(1.30)**	0.836(1.18) 0.647(1.65)	0.690(1.52) 0.730(1.27)	0.642(1.38) 0.726(1.42)
2	A-B012-E-S-GBT	0.695(1.63) 0.828(1.35)	0.845(1.14) 0.702(1.54)	0.678(1.54) 0.739(1.25)	**0.692(1.31)** 0.740(1.39)
3	A-B012-E-SS-GBT	0.704(1.62) 0.834(1.34)	0.846(1.15) 0.715(1.53)	0.681(1.53) 0.741(1.25)	0.668(1.35) 0.741(1.40)
4	R-B012-E-S-GBT	0.712(1.60) 0.808(1.41)	0.837(1.17) 0.635(1.67)	0.659(1.57) **0.757(1.22)**	0.683(1.32) 0.727(1.42)
5	R-B012-M1-S-GBT	**0.716(1.59)** 0.822(1.37)	0.836(1.17) **0.708(1.53)**	**0.706(1.48)** 0.746(1.24)	0.672(1.34) 0.744(1.39)
6	2+5	0.714(1.59) 0.831(1.34)	**0.848(1.13)** **0.717(1.52)**	0.699(1.50) 0.747(1.24)	**0.692(1.31)** **0.750(1.38)**
7	R-B0-E-KNN	0.648(1.73) 0.700(1.70)	0.761(1.39) 0.487(1.89)	0.544(1.76) 0.641(1.43)	0.616(1.42) 0.628(1.62)
8	R-B1-E-KNN	0.547(1.91) 0.535(2.01)	0.684(1.55) 0.634(1.67)	0.444(1.88) 0.649(1.42)	0.536(1.52) 0.576(1.71)
9	R-B2-E-KNN	0.474(2.01) 0.126(2.49)	0.494(1.87) 0.331(2.09)	0.202(2.14) 0.609(1.47)	0.298(1.79) 0.362(1.98)
10	R-B0-M1-KNN	0.581(1.85) 0.672(1.76)	0.771(1.35) 0.485(1.90)	0.516(1.80) 0.644(1.43)	0.601(1.44) 0.610(1.65)
11	R-B1-M1-KNN	0.663(1.70) 0.786(1.49)	0.784(1.33) 0.610(1.71)	0.652(1.59) 0.731(1.30)	0.555(1.50) 0.683(1.52)
12	R-B2-M1-KNN	0.675(1.67) 0.655(1.81)	0.803(1.28) 0.617(1.72)	0.577(1.72) 0.648(1.42)	0.531(1.52) 0.644(1.59)
13	Cons(7+8+9+10+11+12)	0.698(1.66) 0.756(1.68)	0.817(1.28) 0.658(1.68)	0.620(1.68) 0.739(1.31)	0.645(1.41) 0.705(1.49)
14	2+5 (5-fold)	0.713(1.60) 0.831(1.34)	0.843(1.15) 0.698(1.56)	0.693(1.51) 0.737(1.26)	0.670(1.35) 0.741(1.40)

Table 6.8: Performance of different approaches on the S1322 dataset.

Pearson correlation coefficients with RMSE (kcal/mol) in parentheses for binding affinity predictions on 7 protein clusters (CL) in S1322. On the title row, the numbers in parentheses denote the numbers of ligands in the cluster. The median results of 20 repeated runs are reported for the ensemble of trees based methods to account for randomness in the algorithm. For experimental labels, the first letter indicates the complex definition used, 'A' for alpha complex and 'R' for Rips complex. The second part starting with 'B' followed by the integers indicates the dimension of barcode used. The third part indicates the distance function used, 'E' for Euclidean and 'M1' for $\widetilde{\mathbf{M}}^1$. For row 1 through 5, the forth part shows the way of feature construction, 'C' for counts in bins and 'S' for barcode statistics. The last part indicates the regression technique used, 'GBT' for gradient boosting trees and 'KNN' for k-nearest neighbors. The detailed descriptions of the experiments are given in Table 6.7. Row 6 is the results using features of both row 2 and row 5. Row 13 is the consensus results by taking the average of the predictions by row 7 through row 12. Except for specified, all results are obtained from 10-fold cross validations.

6.5.2 Complex based protein-ligand binding affinity prediction

Having demonstrated the representational power of the present topological learning method for characterizing small molecules, we further examine the method on the task of characterizing protein-ligand complex. Biologically, we consider the same task, i.e., the prediction of protein-ligand binding affinity, with a different approach that is based on the structural information of the protein-ligand complexes. Only gradient boosting trees and deep convolutional neural network algorithms are used in this section. All the gradient boosting trees models take the setup described in *Section Methods/Machine learning algorithms/Gradient boosting trees*.

In the present topological learning study, we use four versions of PDBBind core sets as our test sets. For each test set, the corresponding refined set, excluding the core set, is used as the training set.

Groups of topological features and their performance in association with GBT.

The experiments of protein-ligand-complex-based protein-ligand binding affinity prediction for the PDBBind datasets are summarized in Table 6.9.

6.5.2.1 Robustness of GBT algorithm against redundant element combination features and potential overfitting.

It is intuitive that combinations of more than 2 element types are able to enrich the representation especially in the case of higher dimensional barcodes. However, the consideration of combination of more element types rapidly increases the dimension of feature space. In the high dimensional feature space, it is almost inevitable that there exists nonessential and redundant features. Additionally, the importance of a feature varies across different problems and data sets. Therefore, it is preferable to keep all the potentially important features in a

Experiment	Description
R-B0-I-C	0th dimensional barcodes from Rips complex computation with interactive distance matrix based on Euclidean distance are used. Features are generated following *counts in bins* method with bins $\{[0, 2.5), [2.5, 3), [3, 3.5), [3.5, 4.5), [4.5, 6), [6, 12]\}$. Element combinations used are all possible paired choices of one item from {C, N, O, S, CN, CO, NO, CNO} in protein and another item from {C, N, O, S, P, F, Cl, Br, I, CN, CO, CS, NO, NS, OS, CNO, CNS, COS, NOS, CNOS} in ligand, which result in a total of 160 combinations.
R-B0-I-BP	The persistent homology computation and feature generation is the same as R-B0-I-C. However, the element combinations used are all possible paired choices of one item from {C, N, O, S} in protein and another item from {C, N, O, S, P, F, Cl, Br, I} in ligand, which result in a total of 36 element combinations.
R-B0-CI-C	0th dimensional barcodes from Rips complex computation with interactive distance matrix based on the electrostatics correlation function defined in Eq. (6.8) with the parameter $c = 100$. The features are generated following *counts in bins* method with bins $\{(0, 0.1], (0.1, 0.2], (0.2, 0.3], (0.3, 0.4], (0.4, 0.5], (0.5, 0.6], (0.6, 0.7], (0.7, 0.8], (0.8, 0.9], (0.9, 1.0)\}$. The element combinations used are all possible paired choices of one item from {C, N, O, S, H} in protein and another item from {C, N, O, S, P, F, Cl, Br, I, H} in ligand, which result in a total of 50 element combinations.
R-B0-CI-B-S	The barcodes and element combinations are the same as those of R-B0-CI-B-C. The features are generated following the *barcode statistics* method.
A-B12-E-S	1st and 2nd dimensional barcodes from alpha complex computation with Euclidean distance are used. The element combinations considered are all heavy atoms and all carbon atoms. Features are generated following the *barcode statistics* method.

Table 6.9: Experiments for protein-ligand-complex-based protein-ligand binding affinity prediction for the PDBBind datasets.

general model which is expected to cover a wide range of situations. To test the robustness of the model against unimportant features, we select a total of 128 element combinations (i.e., all possible paired choices of one item from {C, N, O, CN, CO, NO, CNO, CNOS} in protein and another item from {C, N, O, S, CN, CO, CS, NO, NS, OS, CNO, CNS, COS, NOS, CNOS, CNOSPFClBrI} in ligand). The 0th, 1st, and 2nd dimensional barcodes are computed for all combinations using alpha complex with Euclidean distance. Features are generated following the barcode statistics method.

A general model with all the features is generated in the first place. The element combinations are then sorted according to their importance scores in the general model. Starting from the most important element combination, one element combination is added to the fea-

ture vector each time and then the resulting feature vector is passed to the machine learning training and testing procedure. The order of adding element combinations is based on their importance scores and thus that a less important feature is added each step.

Fig. 6.8 depicts the changes of Pearson correlation coefficient and RMSE (kcal/mol) with respect to the increase of element combinations in predicting four PDBBind core sets. In all cases, the inclusion of top combinations can readily deliver very good models. The behavior of the present method in PDBBind v2007 is quite different from that in other data sets. The performance of the present method improves almost monotonically as the element combination increases. However, in other three cases, the improvement is unsteady. Nevertheless, the performance fluctuates within a small range, which indicates that the present method is reasonably stable against the increase in element combinations. From a different perspective, the increase in element combinations might lead to overfitting in machine learning. Since the model parameters are fixed before the experiments, it shows that GBT algorithms are not very sensitive to redundant features and are robust against overfitting.

Usefulness of more than 2 element types for interactive 0th dimensional barcodes.

While using element combinations with more than 2 element types with higher dimensional barcodes enriches characterization of geometry, it remains to assess whether interactive 0th dimensional characterization will benefit from element combinations with more element types. As an example, we denote interactive 0th dimensional barcodes for carbon and nitrogen atoms from protein and oxygen atoms from ligand by $\mathbf{B}_{\mathrm{CN-O}}$, barcodes for carbon atoms from protein and oxygen atoms from ligand by $\mathbf{B}_{\mathrm{C-O}}$, and barcodes for nitrogen atoms from protein and oxygen atoms from ligand by $\mathbf{B}_{\mathrm{N-O}}$. In the case of persistent homology barcode representation, $\mathbf{B}_{\mathrm{CN-O}}$ is not strictly the union of $\mathbf{B}_{\mathrm{C-O}}$ and $\mathbf{B}_{\mathrm{N-O}}$.

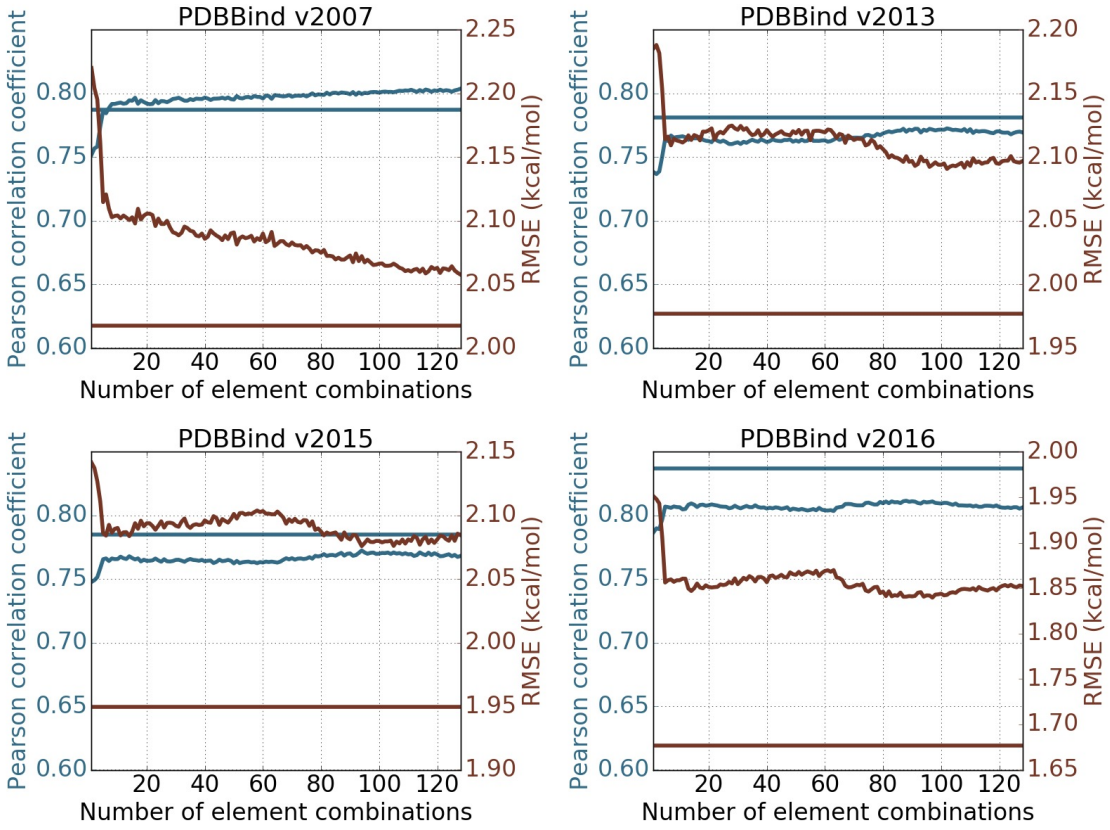

Figure 6.8: Feature robustness tests on PDBBind datasets.[20]
The performance of the topological learning model against the number of included element combinations for predicting on PDBBind core sets and training on PDBBind refined sets minus the core sets. The 1st and 2nd dimensional barcodes computed with alpha complex is used. Features are generated following *barcode statistics* method. Element combinations are all possible paired choices of one item from {C, N, O, CN, CO, NO, CNO, CNOS} in protein and another item from {C, N, O, S, CN, CO, CS, NO, NS, OS, CNO, CNS, COS, NOS, CNOS, CNOSPFClBrI} in ligand, which result in 128 element combinations. The horizontal straight lines represents the performance of the 2D representation with deep convolutional neural network (row 10 in Table 6.10). The blue and red colors correspond to Pearson correlation coefficient and RMSE (kcal/mol) respectively. Each experiment is done by training on refined set minus the core set with the median result of 20 repeated runs reported.

However $\mathbf{B}_{\mathrm{CN-O}}$ might be redundant to $\mathbf{B}_{\mathrm{C-O}}$ and $\mathbf{B}_{\mathrm{N-O}}$. To address this concern, we test features from interactive 0th dimensional barcodes with the 36 element combinations (i.e., { C, N, O, S } for protein and { C, N, O, S, P, F, Cl, Br, I } for ligand) and features for the 160 selected element combinations (i.e., { C, N, O, S, CN, CO, NO, CNO } for protein and { C, N, O, S, P, F, Cl, Br, I, CN, CO, CS, NO, NS, OS, CNO, CNS, COS, NOS, CNOS } for ligand), which are listed as feature group 2 and feature group 1 in Table 6.10. In all the four cases, the features of the 36 combinations (feature group 2) slightly outperforms or performs as well as the features of the 160 combinations (feature group 1) suggesting that element combinations with more than 2 element types are redundant to all the combinations with 2 element types in the case of interactive 0th dimensional characterization.

ID	Experiments	v2007	v2013	v2015	v2016	Average
1	R-B0-I-C	0.799 (2.01)	0.741 (2.14)	0.750 (2.11)	0.813 (1.82)	0.776 (2.02)
2	R-B0-I-BP	**0.816 (1.94)**	0.741 (2.13)	0.750 (2.10)	0.825 (1.78)	0.783 (1.99)
3	R-B0-CI-C	0.791 (2.05)	0.759 (2.10)	0.738 (2.13)	0.801 (1.87)	0.772 (2.04)
4	R-B0-CI-S	0.773 (2.10)	0.762 (2.12)	0.749 (2.13)	0.810 (1.86)	0.774 (2.05)
5	A-B12-E-S	0.736 (2.25)	0.709 (2.26)	0.695 (2.27)	0.752 (2.02)	0.723 (2.20)
6	1+4	0.815 (1.95)	0.780 (2.04)	0.774 (2.04)	0.833 (1.76)	0.801 (1.95)
7	2+4	0.806 (1.99)	0.787 (2.04)	0.770 (2.06)	0.834 (1.77)	0.799 (1.97)
8	1+4+5	0.810 (1.98)	0.792 (2.02)	0.786 (2.02)	0.831 (1.76)	0.805 (1.95)
9	2+4+5	0.802 (2.01)	**0.796** (2.02)	0.782 (2.04)	0.822 (1.79)	0.801 (1.97)
10	2D-CNN-Alpha	0.787 (2.02)	0.781 (**1.98**)	0.785 (1.95)	0.837 (1.68)	0.798 (1.91)
11	1D2D-CNN	0.806 (1.95)	0.781 (**1.98**)	**0.799 (1.91)**	**0.848 (1.64)**	**0.809 (1.87)**

Table 6.10: Performance of different protein-ligand complex based approaches on the PDBBind datasets.

Pearson correlation coefficients with RMSE (kcal/mol) in parentheses for predictions by various groups of features on the four PDBBind core sets. The training sets are the PDBBind refined sets minus the core sets of the same version year. Results of ensemble of trees based methods (rows 1 through 9) are the *median values* of 50 repeated runs to account for randomness in the algorithm. For the deep learning based methods (row 10 and 11), 100 independent models are generated in the first place. A consensus model is built by randomly choosing 50 models out of the 100, and the this process is repeated 1000 times with the median reported. The first letter indicates the definition of complex, 'A' for alpha complex and 'R' for Rips complex. The second part indicates the dimension of barcodes used. The third part indicates the distance function used, 'I' for $\widehat{\mathbf{M}}_{ij}$ defined in Eq. (6.3), 'CI' for the one defined in Eq. (6.8), and 'E' for Euclidean. The last part shows the way of feature construction, 'C' for counts in bins, 'S' for barcode statistics, and 'BP' for only pair of two single elements. The results reported in row 6 through 9 are obtained by combining the features of the rows with the corresponding numbers.

Importance of atomic charge in electrostatic persistence.

In element specific persistent homology, atoms of different element types are characterized separately, which offers a rough and implicit description of the electrostatics of the system. However, such implicit treatment of electrostatics may lose important information because atoms behave differently at different oxidation states. Therefore, we explicitly embed atomic charges in interactive 0th dimensional barcodes as described in Eq. (6.8). The resulting topological features are given in feature group 4 in Table 6.10. It can be seen from Table 6.10 that the combination of feature group 4 and the Euclidean distance based interactive 0th dimensional barcodes (listed as feature group 6 and 7) generally outperforms the results obtained with only Euclidean distance based features. This observation suggests that electrostatics play an important role and should be taken care of explicitly for the protein-ligand binding problem. Additionally, the inclusion of physical interactions in topological invariants opens a promising new direction in topological analysis.

Relevance of elements that are rare with respect to the data sets.

Since the majority of the samples in both training and testing sets only contain atoms of element types, C, N, O, and H, the performance of the model on the samples with rare occurring elements with respect to data sets is hardly reflected by the overall performance statistics. For simplicity, we refer to such rarely occurring elements with respect to data sets simply by rarely occurring elements in the discussion follows. To assess the aspects of the model that potentially affect the performance on the samples containing rarely occurring elements, we picked the samples containing each rarely occurring element from the original testing set as a new testing set. Three experiments are carried out to address two questions: "Are the training samples containing the same rarely occurring element crucial?" and "Are features addressing the rarely occurring element important?". A short answer is yes to

both according to the results shown in Fig. 6.9. Specifically, for each rarely occurring element, the exclusion of samples containing this element in training set and the exclusion of features addressing this element will both cause degenerated results. It is also shown that the exclusion of samples of the rarely occurring element leads to much worse results. Since both modifications of the model deliver worse results, we conclude that including the samples in the training set with similar compositions to the test sample is crucial to the success of the model on this specific test sample. Even the inclusion of features of more element types or element combinations does not deliver better results in the general testing sets, such features should still be kept in the model in case that a sample with a similar element composition comes in as a test sample.

2D persistence for topological deep convolutional neural networks.

Deep learning is potentially more powerful than many other machine learning algorithms when the data size is sufficiently large. In the present work, it is natural to construct a 2D topological representation by incorporating the element combination as an additional dimension, resulting in 16 channels as defined in *Section Feature generation from topological invariants* . Here 128 element combinations (i.e., all possible paired choices of one item from {C, N, O, CN, CO, NO, CNO, CNOS} in protein and another item from {C, N, O, S, CN, CO, CS, NO, NS, OS, CNO, CNS, COS, NOS, CNOS, CNOSPFClBrI} in ligand) are used for 2D analysis. The advantage of introducing this extra dimension with convolutional neural networks is to prevent unimportant features from interacting with important ones at the lower levels of the model whilst generally unimportant features are still kept in the model in case that they are essential to specific problems or a certain portion of the data set.

As shown in Fig. 6.8, for all the data sets except the PDBBind v2007 set, the 2D topological deep learning with convolutional neural networks performs significantly better. The

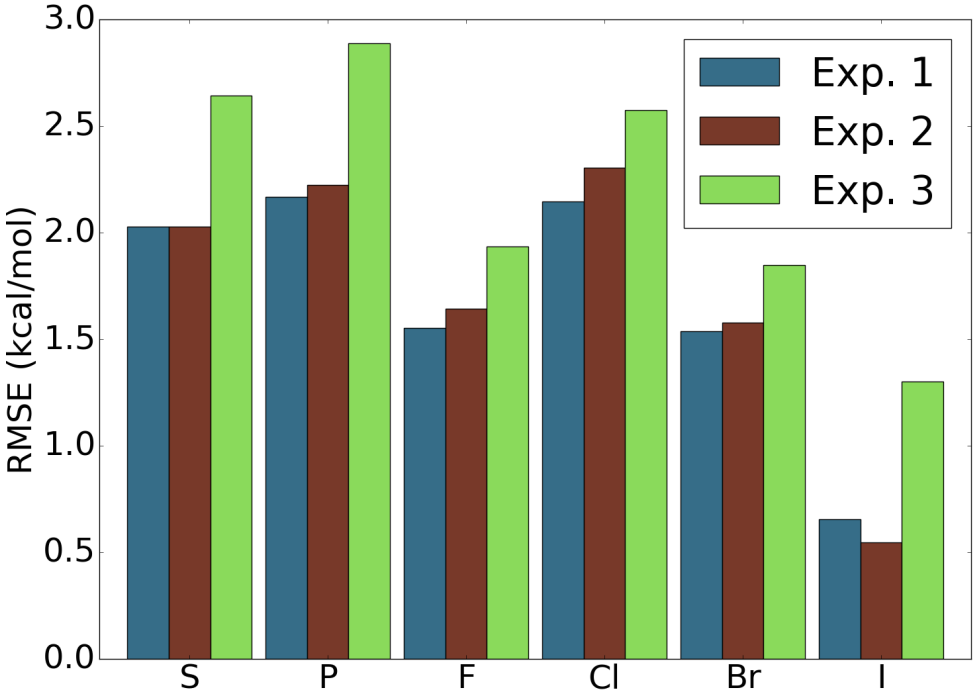

Figure 6.9: Assessment of performance of the model on samples with elements that are rare in the data sets.[20]
For the four data sets PDBBind v2007, v2013, v2015, and v2016 [119], and for each element, the testing set is the subset of the original core sets with only ligands that contain atoms of the particular element type. The features used are features with ID=7 in Table 6.10. The reported RMSE is the average taken over the four data sets. Experiment 1: Training set is the original training set and all the features are used. Experiment 2: Training set is the original training set and only features that do not involve the particular element are used. Experiment 3: Training set is the original training set excluding the samples that contain atoms of the particular element type and all features are used. For most of the elements, experiment 1 achieves the best result and experiment 3 yields the worst performance.

inferior performance of convolutional neural networks in v2007 might be a result of the small data size. Note that v2007 training set has 1105 protein-ligand complexes, whereas other training sets have more than 2700 complexes. Consequently, topological deep convolutional neural networks are able to outperform the topological GBT algorithm in predicting v2013, v2015 and v2016 core sets. Indeed, topological deep convolutional neural networks have advantages in dealing with large data sets.

6.5.3 Structure-based virtual screening

In our final model TopVS reported in Table 6.6, we use topological descriptors of both protein-compound interactions and only the compounds (i.e., ligands and decoys) and take a consensus model on top of several ensemble of trees models and a deep learning model. We have also tested the behavior of our topological learning model TopVS-ML using either one of the aforementioned descriptions. The tests are done with TopVS-ML because that TopVS-DL is much more time consuming. When only topological descriptor of small molecules are used, which falls into the category of ligand-based virtual screening, an AUC of 0.81 is achieved. For the topological learning model using only the descriptions of protein-ligand interactions, an AUC of 0.77 is achieved. An AUC of 0.83 is obtained with a model combining both sets of descriptors which is better than each individual performance, suggesting that the two groups of descriptors are complementary to each other and are both important for achieving satisfactory results. The marginal improvement made by protein-compound complexes maybe due to the various docking quality. Similar situation was encountered by a deep learning method [158]. For the targets with high quality results by Autodock Vina (AUC of ADV > 0.8), the ligand-based features achieve an AUC of 0.81 and the complex-based features achieve an AUC of 0.86. On the other hand, for the targets with low quality

Target	ADV	LIG	COM	ALL
AR	0.81	0.83	0.93	0.90
COX2	0.86	0.97	0.80	0.97
DHFR	0.82	0.95	0.94	0.96
$ER_{agonist}$	0.84	0.69	0.91	0.81
MR	0.82	0.78	0.91	0.89
PPARg	0.82	0.70	0.72	0.72
RXRa	0.95	0.74	0.91	0.79
SAHH	0.80	0.81	0.72	0.84
Average	0.84	0.81	0.86	0.86

Table 6.11: The AUC for autodock vina, TopVS-ML with only compound features, TopVS-ML with only protein-compound complex features, and TopVS-ML with all features. The targets with high quality results by Autodock Vina are reported (AUC > 0.8)

results by Autodock Vina (AUC of ADV < 0.5), the ligand-based features achieve an AUC of 0.82 and the complex-based features achieve an AUC of 0.74. The results of these cases are listed in Table 6.11 and Table 6.12. This observation suggests that the performance of features describing the interactions and the geometry of protein-compounds complexes highly depends on the quality of docking results.

Our model with small molecular descriptors delivers an AUC of 0.81, which is comparably well to the other top performing methods. The performance of this model is also competitive in the regime of protein-ligand binding affinity prediction based on experimentally solved complex structures as is shown in *Section Discussion/Ligand based protein-ligand binding affinity prediction*. These results suggest that topology based small molecule characterization proposed in this work is potentially useful in other applications involving small molecules, such as predictions of toxicity, solubility and partition coefficient of small molecules.

Target	ADV	LIG	COM	ALL
ACE	0.42	0.85	0.78	0.81
ADA	0.49	0.89	0.89	0.89
AmpC	0.34	0.56	0.37	0.53
FGFr1	0.44	0.97	0.71	0.95
GPB	0.48	0.70	0.69	0.71
NA	0.37	0.79	0.82	0.84
PDGFrb	0.32	0.98	0.90	0.96
Average	0.41	0.82	0.74	0.81

Table 6.12: The AUC for autodock vina, TopVS-ML with only compound features, TopVS-ML with only protein-compound complex features, and TopVS-ML with all features. The targets with low quality results by Autodock Vina are reported (AUC < 0.5)

6.6 Conclusion

Persistent homology is a relatively new branch of algebraic topology and is one of the main tools in topological data analysis. The topological simplification of biomolecular systems was a major motivation of the earlier persistent homology development [62, 216]. Persistent homology has been applied to computational biology [99, 75, 48, 156, 75, 198, 195, 202, 201, 200, 186, 117]. However, the predictive power of primitive persistent homology was limited in early topological learning applications [21]. To address this challenge, we have recently introduced element specific persistent homology to retain chemical and biological information during the topological abstraction of biomolecules [23, 24, 26]. The resulting topological learning approach offers competitive predictions of protein-ligand binding affinity and mutation induced protein stability changes. However, persistent homology based approaches for small molecules have not been developed and its representability and predictive powers for the interaction of small molecules with macromolecules have not been extensively studied.

The present work further introduces multi-component persistent homology, multi-level persistent homology and electrostatic persistence for chemical and biological characteriza-

tion, analysis and modeling. Multi-component persistent homology takes a combinatorial approach to create possible element specific topological representations. Multi-level persistent homology allows tailored topological descriptions of any desirable interaction in biomolecules which is especially useful for small molecules. Electrostatic persistence incorporates partial charges that are essential to biomolecules into topological invariants. These approaches are implemented via the appropriate construction of the distance matrix for filtration. The representation power and reduction power of multi-component persistent homology, multi-level persistent homology and electrostatic persistence are validated by two databases, namely PDBBind [119] and DUD [93, 135]. PDBBind involves more than 4,000 high quality protein-ligand complexes and DUD contains near 100,000 small compounds. Two classes of problems are used to test the proposed topological methods, including the prediction of protein-ligand binding affinities and the discrimination of active ligands from decoys (virtual screening). In both problems, we examine the representability of proposed topological learning methods on small molecules, which are somewhat more difficult to describe by persistent homology due to their chemical diversity, variability and sensitivity. Additionally, these methods are tested on their ability to handle the full protein-ligand complexes. Advanced machine learning methods, including Wasserstein metric based k-nearest neighbors (KNNs), gradient boosting trees (GBT), random forest (RF), extra trees (ET) and deep convolutional neural networks (CNN) are utilized in the present work to facilitate the proposed topological methods, rendering advanced topological learning algorithms for quantitative and qualitative biomolecular predictions. The thorough examination of the method on the prediction of binding affinity for experimentally solved protein-ligand complexes leads to a structure-based virtual screening method, TopVS, which outperforms other methods. The feature sets introduced in this work for small molecules and protein-ligand complexes can be extended to

other applications such as 3D-structure based prediction of toxicity, solubility, and partition coefficient for small molecules and complex structure based prediction of protein-nucleic acid binding and protein-protein binding affinities.

Chapter 7

Dissertation contribution

The main contributions of this dissertation are listed as follows.

- In Chapter 3, we develop a quantitative and predictive model based on persistent homology and deep learning for protein-ligand binding affinity prediction and the prediction of protein stability change upon mutation. This model achieves top performance on benchmarks where many other methods have been tested. To the best of our knowledge, this work is the first competitive topological data analysis based predictive model in the field of molecular biology.

- In Chapter 4, motivated by embedding physical properties to persistent homology representations of biomolecules, we propose a method called enriched barcode using cohomology and tools from graph theory. This method has a general application to the situations where data come with multiple heterogeneous dimensions.

- In Chapter 5, we develop a persistent homology construction called evolutionary homology tailored for the analysis of trajectories from coupled dynamical systems. This method is applied to coupled dynamical systems associated to molecular systems and shows competitive results in the modeling of protein flexibility.

- In Chapter 6, we carefully assess the representability of persistent homology of biomolecules. As a result, we propose persistent homology representations specifically designed for

small molecules which we call multi-level persistent homology as well as representations for macromolecules addressing chemical and biological complexities. The method developed in this chapter also leads to a top-performing tool for structural-based virtual screening.

- In addition to the high performance of this work on benchmarks where the testing data are given to the researchers, we have also achieved top performance in world-wide challenges of blind predictions in drug design where testing data are hidden from researchers before the prediction results are submitted.[141]

- The works described in this dissertation are made accessible by attaching source codes to the publications [26, 20]. Several accompanying webservers are made available to users, "TDL-MP" (http://weilab.math.msu.edu/TDL/TDL-MP/) and "TML-MP" (http://weilab.math.msu.edu/TML/TML-MP/) for predictors of protein stability changes upon mutations, and "TDL-BP"(http://weilab.math.msu.edu/TDL/TDL-BP/) and "TML-BP" (http://weilab.math.msu.edu/TML/TML-BP/).

The contents of this dissertation are mostly adopted from the follow publications and preprints:

- **Zixuan Cang** and Guo-wei Wei. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *Plos Computational Biology*, 13(7):e1005690, 2017.

- **Zixuan Cang** and Guo-Wei Wei. Persistent cohomology for data with multicomponent heterogeneous information. *preprint*, 2018.

- **Zixuan Cang**, Elizabeth Munch, and Guo-Wei Wei. Evolutionary homology on coupled dynamical systems. *arXiv preprint arXiv:1802.04677*, 2018.

- **Zixuan Cang**, Lin Mu, and Guo-Wei Wei. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *Plos Computational Biology*, 14(1):e1005929. https://doi.org/10.1371/journal.pcbi.1005929, 2018.

This work led to the following publications/preprints that are not discussed in this dissertation:

- **Zixuan Cang**, Lin Mu, Kedi Wu, Kristopher Opron, Kelin Xia, and Guo-Wei Wei. A topological approach for protein classification. *Molecular Based Mathematical Biology*, 3(1), 2015.

- **Zixuan Cang** and Guo-Wei Wei. Analysis and prediction of protein folding energy changes upon mutation by element specific persistent homology. *Bioinformatics*, 33(22):3549–3557, 2017.

- **Zixuan Cang** and Guo-Wei Wei. Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. *International journal for numerical methods in biomedical engineering*, 34(2):e2914, 2018.

The following publications/preprints are also related to this dissertation:

- Rundong Zhao, **Zixuan Cang**, Yiying Tong, and Guo wei Wei. Protein pocket detection via convex hull surface evolution and associated reeb graph. *accepted by Bioinformatics/proceedings of ECCB 2018*, 2018.

- Duc Duy Nguyen, **Zixuan Cang**, Kedi Wu, Menglun Wang, Yin Cao, and Guo-Wei Wei. Mathematical deep learning for pose and binding affinity prediction and ranking in d3r grand challenges. *arXiv preprint arXiv:1804.10647*, 2018.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] SAMPL6 challenge. https://drugdesigndata.org/about/sampl6. Accessed: 2018-04-10.

[2] Henry Adams, Andrew Tausz, and Mikael Vejdemo-Johansson. Javaplex: A research software package for persistent (co) homology. In *International Congress on Mathematical Software*, pages 129–136. Springer, 2014.

[3] Mamiko Arai, Vicky Brandt, and Yuri Dabaghian. The effects of theta precession on spatial learning and simplicial complex dynamics in a topological model of the hippocampal spatial map. *PLoS Computational Biology*, 10(6):e1003651, 2014.

[4] Marcelino Arciniega and Oliver F Lange. Improvement of virtual screening results by docking data feature analysis. *Journal of chemical information and modeling*, 54(5):1401–1411, 2014.

[5] M. Stuart Armstrong, Garrett M. Morris, Paul W. Finn, Raman Sharma, Loris Moretti, Richard I. Cooper, and W. Graham Richards. Electroshape: fast molecular similarity calculations incorporating shape, chirality and electrostatics. *Journal of computer-aided molecular design*, 24(9):789–801, 2010.

[6] Hossam M. Ashtawy and Nihar R. Mahapatra. A comparative assessment of ranking accuracies of conventional and machine-learning-based scoring functions for protein-ligand binding affinity prediction. *IEEE/ACM Transactions on computational biology and bioinformatics*, 9(5):1301–1313, 2012.

[7] Ivet Bahar, Ali Rana Atilgan, and Burak Erman. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding and Design*, 2(3):173–181, 1997.

[8] Ulrich Bauer. Ripser: a lean c++ code for the computation of Vietoris-Rips persistence barcodes. *Software available at https://github. com/Ripser/ripser*, 2017.

[9] Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Distributed computation of persistent homology. In *2014 proceedings of the sixteenth workshop on algorithm engineering and experiments (ALENEX)*, pages 31–38. SIAM, 2014.

[10] Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Distributed computation of persistent homology. In *2014 Proceedings of the Sixteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 31–38. SIAM, 2014.

[11] Ulrich Bauer, Michael Kerber, Jan Reininghaus, and Hubert Wagner. Phat–persistent homology algorithms toolbox. *Journal of Symbolic Computation*, 78:76–90, 2017.

[12] K. Abdulla Bava, M. Michael Gromiha, Hatsuho Uedaira, Koji Kitajima, and Akinori Sarai. Protherm, version 4.0: thermodynamic database for proteins and mutants. *Nucleic acids research*, 32(suppl_1):D120–D121, 2004.

[13] Nathan Bell and Anil N Hirani. Pydec: software and algorithms for discretization of exterior calculus. *ACM Transactions on Mathematical Software (TOMS)*, 39(1):3, 2012.

[14] Paul Bendich, Herbert Edelsbrunner, and Michael Kerber. Computing robustness and persistence for images. *IEEE transactions on visualization and computer graphics*, 16(6):1251–1260, 2010.

[15] James Bergstra, Daniel Yamins, and David D Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *ICML (1)*, 28:115–123, 2013.

[16] Niklas Berliner, Joan Teyra, Recep Çolak, Sebastian Garcia Lopez, and Philip M. Kim. Combining structural modeling with ensemble machine learning to accurately predict protein fold stability and binding affinity effects upon mutation. *PloS one*, 9(9):e107353, 2014.

[17] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic acids research*, 28(1):35–242, 2000.

[18] Jesse Berwald, Marian Gidea, and Mikael Vejdemo-Johansson. Automatic recognition and tagging of topologically different regimes in dynamical systems. *arXiv preprint arXiv:1312.2482*, 2013.

[19] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[20] Zixuan Cang, Lin Mu, and Guo-Wei Wei. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *Plos Computational Biology*, 14(1):e1005929. https://doi.org/10.1371/journal.pcbi.1005929, 2018.

[21] Zixuan Cang, Lin Mu, Kedi Wu, Kristopher Opron, Kelin Xia, and Guo-Wei Wei. A topological approach for protein classification. *Molecular Based Mathematical Biology*, 3(1), 2015.

[22] Zixuan Cang, Elizabeth Munch, and Guo-Wei Wei. Evolutionary homology on coupled dynamical systems. *arXiv preprint arXiv:1802.04677*, 2018.

[23] Zixuan Cang and Guo-Wei Wei. Analysis and prediction of protein folding energy changes upon mutation by element specific persistent homology. *Bioinformatics*, 33(22):3549–3557, 2017.

[24] Zixuan Cang and Guo-Wei Wei. Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. *International journal for numerical methods in biomedical engineering*, 34(2):e2914, 2018.

[25] Zixuan Cang and Guo-Wei Wei. Persistent cohomology for data with multicomponent heterogeneous information. *preprint*, 2018.

[26] Zixuan Cang and Guowei Wei. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *Plos Computational Biology*, 13(7):e1005690, 2017.

[27] Emidio Capriotti, Piero Fariselli, and Rita Casadio. I-mutant2. 0: predicting stability changes upon mutation from the protein sequence or structure. *Nucleic acids research*, 33(suppl_2):W306–W310, 2005.

[28] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[29] Gunnar Carlsson. Topological pattern recognition for point cloud data. *Acta Numerica*, 23:289–368, 2014.

[30] Gunnar Carlsson, Vin De Silva, and Dmitriy Morozov. Zigzag persistent homology and real-valued functions. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 247–256. ACM, 2009.

[31] Gunnar Carlsson, Tigran Ishkhanov, Vin De Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *International journal of computer vision*, 76(1):1–12, 2008.

[32] Gunnar Carlsson and Afra Zomorodian. The theory of multidimensional persistence. *Discrete & Computational Geometry*, 42(1):71–93, 2009.

[33] Gunnar Carlsson, Afra Zomorodian, Anne Collins, and Leonidas J Guibas. Persistence barcodes for shapes. *International Journal of Shape Modeling*, 11(02):149–187, 2005.

[34] Rich Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[35] D. A. Case, J. T. Berryman, R. M. Betz, D. S. Cerutti, T. E. Cheatham III, T. A. Darden, R. E. Duke, T. J. Giese, H. Gohlke, A. W. Goetz, N. Homeyer, S. Izadi, P. Janowski, J. Kaus, A. Kovalenko, T. S. Lee, S. LeGrand, P. Li, T. Luchko, R. Luo, B. Madej, K. M. Merz, G. Monard, P. Needham, H. Nguyen, H. T. Nguyen, I. Omelyan,

A. Onufriev, D. R. Roe, A. Roitberg, R. Salomon-Ferrer, C. L. Simmerling, W. Smith, J. Swails, R. C. Walker, J. Wang, R.M. Wolf, X. Wu, D. M. York, and P. A. Kollman. Amber 2015. *University of California, San Francisco*, 2015.

[36] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J Guibas, and Steve Y Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 237–246. ACM, 2009.

[37] Frédéric Chazal, Vin De Silva, Marc Glisse, and Steve Oudot. *The structure and stability of persistence modules*. Springer, 2016.

[38] Tiejun Cheng, Xun Li, Yan Li, Zhihai Liu, and Renxiao Wang. Comparative assessment of scoring functions on a diverse test set. *Journal of chemical information and modeling*, 49(4):1079–1093, 2009.

[39] Yongwook Choi, Gregory E Sims, Sean Murphy, Jason R Miller, and Agnes P Chan. Predicting the functional effect of amino acid substitutions and indels. *PloS one*, 7(10):e46688, 2012.

[40] François Chollet. Keras. https://github.com/fchollet/keras, 2015.

[41] Samir Chowdhury and Facundo Mémoli. Persistent path homology of directed networks. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1152–1169. SIAM, 2018.

[42] Fan RK Chung. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[43] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[44] David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. Lipschitz functions have L_p-stable persistence. *Foundations of computational mathematics*, 10(2):127–139, 2010.

[45] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 119–126. ACM, 2006.

[46] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[47] Jason B Cross, David C Thompson, Brajesh K Rai, J Christian Baber, Kristi Yi Fan, Yongbo Hu, and Christine Humblet. Comparison of several molecular docking

programs: pose prediction and virtual screening accuracy. *Journal of chemical information and modeling*, 49(6):1455–1474, 2009.

[48] Yuri Dabaghian, Facundo Mémoli, Loren Frank, and Gunnar Carlsson. A topological paradigm for hippocampal spatial map formation using persistent homology. *PLoS computational biology*, 8(8):e1002581, 2012.

[49] George E Dahl, Navdeep Jaitly, and Ruslan Salakhutdinov. Multi-task neural networks for qsar predictions. *arXiv preprint arXiv:1406.1231*, 2014.

[50] Guillaume Damiand. Combinatorial maps. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.0 edition, 2012. www.cgal.org/Manual/4.0/doc_html/cgal_manual/packages.html #Pkg:CombinatorialMaps.

[51] I. K. Darcy and M. Vazquez. Determining the topology of stable protein-DNA complexes. *Biochemical Society Transactions*, 41:601–605, 2013.

[52] Bhaskar DasGupta and Jie Liang. *Models and Algorithms for Biomolecules and Molecular Networks*. John Wiley & Sons, 2016.

[53] Vin De Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co) homology. *Inverse Problems*, 27(12):124003, 2011.

[54] Vin De Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. *Discrete & Computational Geometry*, 45(4):737–759, 2011.

[55] Yves Dehouck, Aline Grosfils, Benjamin Folch, Dimitri Gilis, Philippe Bogaerts, and Marianne Rooman. Fast and accurate predictions of protein stability changes upon mutations using statistical potentials and neural networks: Popmusic-2.0. *Bioinformatics*, 25(19):2537–2543, 2009.

[56] Omar NA Demerdash, Michael D. Daily, and Julie C. Mitchell. Structure-based predictive models for allosteric hot spots. *PLoS computational biology*, 5(10):e1000531, 2009.

[57] Barbara Di Fabio and Claudia Landi. A Mayer–Vietoris formula for persistent homology with an application to shape recognition in the presence of occlusions. *Foundations of Computational Mathematics*, 11(5):499, 2011.

[58] T. J. Dolinsky, J. E. Nielsen, J. A. McCammon, and N. A. Baker. PDB2PQR: An automated pipeline for the setup, execution, and analysis of Poisson-Boltzmann electrostatics calculations. *Nucleic Acids Research*, 32:W665–W667, 2004.

[59] Jacob D Durrant, Aaron J Friedman, Kathleen E Rogers, and J Andrew McCammon. Comparing neural-network scoring functions and the state of the art: applications to common library screening. *Journal of chemical information and modeling*, 53(7):1726–1735, 2013.

[60] Herbert Edelsbrunner. *Weighted alpha shapes*. University of Illinois at Urbana-Champaign, Department of Computer Science, 1992.

[61] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[62] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 454–463. IEEE, 2000.

[63] Matthew D. Eldridge, Christopher W. Murray, Timothy R. Auton, Gaia V. Paolini, and Roger P. Mee. Empirical scoring functions: I. the development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *Journal of computer-aided molecular design*, 11(5):425–445, 1997.

[64] A Evgeniou and Massimiliano Pontil. Multi-task feature learning. *Advances in neural information processing systems*, 19:41, 2007.

[65] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004.

[66] Xin Feng and Yiying Tong. Choking loops on surfaces. *IEEE transactions on visualization and computer graphics*, 19(8):1298–1306, 2013.

[67] Xin Feng, Kelin Xia, Yiying Tong, and Guo-Wei Wei. Geometric modeling of subcellular structures, organelles, and multiprotein complexes. *International Journal for Numerical Methods in Biomedical Engineering*, 28(12):1198–1223, 2012.

[68] Alan R Fersht. Dissection of the structure and activity of the tyrosyl-trna synthetase by site-directed mutagenesis. *Biochemistry*, 26(25):8031–8037, 1987.

[69] Lukas Folkman, Bela Stantic, Abdul Sattar, and Yaoqi Zhou. EASE-MM: Sequence-based prediction of mutation-induced stability changes with feature-based multiple models. *Journal of molecular biology*, 428(6):1394–1405, 2016.

[70] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[71] Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[72] Patrizio Frosini. A distance for similarity classes of submanifolds of a euclidean space. *Bulletin of the Australian Mathematical Society*, 42(3):407–415, 1990.

[73] Patrizio Frosini and Claudia Landi. Persistent betti numbers for a noise tolerant shape-based approach to image retrieval. *Pattern Recognition Letters*, 34(8):863–872, 2013.

[74] Peter Gabriel. Unzerlegbare darstellungen i. *Manuscripta mathematica*, 6(1):71–103, 1972.

[75] Marcio Gameiro, Yasuaki Hiraoka, Shunsuke Izumi, Miroslav Kramar, Konstantin Mischaikow, and Vidit Nanda. A topological measurement of protein compressibility. *Japan Journal of Industrial and Applied Mathematics*, 32(1):1–17, 2015.

[76] Marcio Gameiro, Konstantin Mischaikow, and William Kalies. Topological characterization of spatial-temporal chaos. *Physical Review E*, 70(3):035203, 2004.

[77] Samuel Genheden and Ulf Ryde. The MM/PBSA and MM/GBSA methods to estimate ligand-binding affinities. *Expert opinion on drug discovery*, 10(5):449–461, 2015.

[78] Ivan Getov, Marharyta Petukh, and Emil Alexov. Saafec: predicting the effect of single point mutations on protein folding free energy using a knowledge-modified mm/pbsa approach. *International journal of molecular sciences*, 17(4):512, 2016.

[79] Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[80] Robert Ghrist and Abubakr Muhammad. Coverage and hole-detection in sensor networks via homology. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 254–260. IEEE, 2005.

[81] Robert W Ghrist. *Elementary applied topology*. Createspace Seattle, 2014.

[82] Michael K Gilson and Huan-Xiang Zhou. Calculation of protein-ligand binding affinities. *Annual review of biophysics and biomolecular structure*, 36, 2007.

[83] Nobuhiro Go, Tosiyuki Noguti, and Testuo Nishikawa. Dynamics of a small globular protein in terms of low-frequency vibrational modes. *Proceedings of the National Academy of Sciences*, 80(12):3696–3700, 1983.

[84] Alexander Grigor'yan, Yong Lin, Yuri Muranov, and Shing-Tung Yau. Homologies of path complexes and digraphs. *arXiv preprint arXiv:1207.2834*, 2012.

[85] Raphael Guerois, Jens Erik Nielsen, and Luis Serrano. Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *Journal of molecular biology*, 320(2):369–387, 2002.

[86] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2001.

[87] Jean-Claude Hausmann et al. On the vietoris-rips complexes and a cohomology theory for metric spaces. *Annals of Mathematics Studies*, 138:175–188, 1995.

[88] Christine Heitsch and Svetlana Poznanović. Combinatorial insights into rna secondary structure. In *Discrete and topological models in molecular biology*, pages 145–166. Springer, 2014.

[89] Anil N Hirani, Kaushik Kalyanaraman, Han Wang, and Seth Watts. Cohomologous harmonic cochains. *arXiv preprint arXiv:1012.2835*, 2010.

[90] Anil Nirmal Hirani. *Discrete exterior calculus*. PhD thesis, California Institute of Technology, 2003.

[91] Danijela Horak, Slobodan Maletić, and Milan Rajković. Persistent homology of complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(03):P03034, 2009.

[92] Gang Hu, Junzhong Yang, and Wenji Liu. Instability and controllability of linearly coupled oscillators: Eigenvalue analysis. *Phys. Rev. E*, 58:4440– 4453, 1998.

[93] Niu Huang, Brian K. Shoichet, and John J. Irwin. Benchmarking sets for molecular docking. *Journal of medicinal chemistry*, 49(23):6789–6801, 2006.

[94] Sheng-You Huang and Xiaoqin Zou. An iterative knowledge-based scoring function to predict protein–ligand interactions: I. derivation of interaction potentials. *Journal of computational chemistry*, 27(15):1866–1875, 2006.

[95] John J. Irwin and Brian K. Shoichet. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1):177–182, 2005.

[96] George A. Jeffrey and George A. Jeffrey. *An introduction to hydrogen bonding*, volume 32. Oxford university press New York, 1997.

[97] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed ¡today¿].

[98] Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek. *Computational homology*, volume 157. Springer Science & Business Media, 2006.

[99] Peter M. Kasson, Afra Zomorodian, Sanghyun Park, Nina Singhal, Leonidas J. Guibas, and Vijay S. Pande. Persistent voids: a new structural metric for membrane fusion. *Bioinformatics*, 23(14):1753–1759, 2007.

[100] Elizabeth H Kellogg, Andrew Leaver-Fay, and David Baker. Role of conformational sampling in computing mutation-induced changes in protein structure and stability. *Proteins: Structure, Function, and Bioinformatics*, 79(3):830–838, 2011.

[101] Michael Kerber, Dmitriy Morozov, and Arnur Nigmetov. Geometry helps to compare persistence diagrams. *Journal of Experimental Algorithmics (JEA)*, 22:1–4, 2017.

[102] Firas A Khasawneh and Elizabeth Munch. Exploring equilibria in stochastic delay differential equations using persistent homology. In *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V008T11A034–V008T11A034. American Society of Mechanical Engineers, 2014.

[103] Firas A. Khasawneh and Elizabeth Munch. Chatter detection in turning using persistent homology. *Mechanical Systems and Signal Processing*, 70:527–541, 2016.

[104] Firas A. Khasawneh and Elizabeth Munch. Utilizing topological data analysis for studying signals of time-delay systems. In *Time Delay Systems*, pages 93–106. Springer, 2017.

[105] Sarah L. Kinnings, Nina Liu, Peter J. Tonge, Richard M. Jackson, Lei Xie, and Philip E. Bourne. A machine learning-based method to improve docking scoring functions and its application to drug repurposing. *Journal of chemical information and modeling*, 51(2):408–419, 2011.

[106] Miroslav Kramár, Rachel Levanger, Jeffrey Tithof, Balachandra Suri, Mu Xu, Mark Paul, Michael F Schatz, and Konstantin Mischaikow. Analysis of kolmogorov flow and rayleigh–bénard convection using persistent homology. *Physica D: Nonlinear Phenomena*, 334:82–98, 2016.

[107] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[108] Brett M. Kroncke, Amanda M. Duran, Jeffrey L. Mendenhall, Jens Meiler, Jeffrey D. Blume, and Charles R. Sanders. Documentation of an imperative to improve methods for predicting membrane protein stability. *Biochemistry*, 55(36):5002–5009, 2016.

[109] Tugba G Kucukkal, Marharyta Petukh, Lin Li, and Emil Alexov. Structural and physico-chemical effects of disease and non-disease nssnps on proteins. *Current opinion in structural biology*, 32:18–24, 2015.

[110] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[111] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[112] Hyekyoung Lee, Hyejin Kang, Moo K Chung, Bung-Nyun Kim, and Dong Soo Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE transactions on medical imaging*, 31(12):2267–2277, 2012.

[113] Michael Lesnick and Matthew Wright. Interactive visualization of 2-d persistence modules. *arXiv preprint arXiv:1512.00180*, 2015.

[114] Hongjian Li, Kwong-Sak Leung, Man-Hon Wong, and Pedro J Ballester. Substituting random forest for multiple linear regression improves binding affinity prediction of scoring functions: Cyscore as a case study. *BMC bioinformatics*, 15(1):291, 2014.

[115] Hongjian Li, Kwong-Sak Leung, Man-Hon Wong, and Pedro J Ballester. Improving autodock vina using random forest: the growing accuracy of binding affinity prediction by the effective exploitation of larger data sets. *Molecular informatics*, 34(2-3):115–126, 2015.

[116] Hongjian Li, Kwong-Sak Leung, Man-Hon Wong, and Pedro J. Ballester. Low-quality structural and interaction data improves binding affinity prediction via random forest. *Molecules*, 20(6):10947–10962, 2015.

[117] Beibei Liu, Bao Wang, Rundong Zhao, Yiying Tong, and Guo-Wei Wei. ESES: software for Eulerian solvent excluded surface. *Journal of computational chemistry*, 38(7):446–466, 2017.

[118] Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient l 2, 1-norm minimization. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 339–348. AUAI Press, 2009.

[119] Zhihai Liu, Yan Li, Li Han, Jie Li, Jie Liu, Zhixiong Zhao, Wei Nie, Yuchen Liu, and Renxiao Wang. PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics*, 31(3):405–412, 2014.

[120] Alessandro Lusci, Gianluca Pollastri, and Pierre Baldi. Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. *Journal of chemical information and modeling*, 53(7):1563–1575, 2013.

[121] Jr. MacKerell, A. D., D. Bashford, M. Bellot, Jr. Dunbrack, R. L., J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, III Reiher,

W. E., B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *Journal of Physical Chemistry B*, 102(18):3586–3616, 1998.

[122] Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The Gudhi library: Simplicial complexes and persistent homology. In *International Congress on Mathematical Software*, pages 167–174. Springer, 2014.

[123] JL Martinez and F Baquero. Mutation frequencies and antibiotic resistance. *Antimicrobial agents and chemotherapy*, 44(7):1771–1777, 2000.

[124] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. Deeptox: toxicity prediction using deep learning. *Frontiers in Environmental Science*, 3:80, 2016.

[125] Marvin Minsky and Seymour A Papert. *Perceptrons: an introduction to computational geometry*. MIT press, 2017.

[126] Konstantin Mischaikow, Marian Mrozek, J Reiss, and Andrzej Szymczak. Construction of symbolic dynamics from experimental time series. *Physical Review Letters*, 82(6):1144, 1999.

[127] Konstantin Mischaikow and Vidit Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete & Computational Geometry*, 50(2):330–353, 2013.

[128] Maria A Miteva, Frederic Guyon, and Pierre Tufféry. Frog2: Efficient 3d conformation ensemble generator for small compounds. *Nucleic acids research*, 38(suppl 2):W622–W627, 2010.

[129] Dmitriy Morozov. Dionysus. *http://www.mrzv.org/software/dionysus/*, 2015.

[130] Garrett M. Morris, Ruth Huey, William Lindstrom, Michel F. Sanner, Richard K. Belew, David S. Goodsell, and Arthur J. Olson. Autodock4 and autodocktools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry*, 30(16):2785–2791, 2009.

[131] Ingo Muegge and Yvonne C Martin. A general and fast scoring function for protein-ligand interactions: a simplified potential approach. *Journal of medicinal chemistry*, 42(5):791–804, 1999.

[132] Daniel Müllner and Aravindakshan Babu. Python mapper: An open-source toolchain for data exploration, analysis and visualization. *See http://danifold. net/mapper*, 2013.

[133] Elizabeth Munch. *Applications of persistent homology to time varying systems*. PhD thesis, 2013.

[134] Elizabeth Munch. A users guide to topological data analysis. *Journal of Learning Analytics*, 4(2):47–61, 2017.

[135] Michael M. Mysinger and Brian K. Shoichet. Rapid context-dependent ligand desolvation in molecular docking. *Journal of chemical information and modeling*, 50(9):1561–1573, 2010.

[136] Vidit Nanda. Perseus: the persistent homology software. *Software available at http://www. sas. upenn. edu/˜ vnanda/perseus*, 2012.

[137] Marco AC Neves, Maxim Totrov, and Ruben Abagyan. Docking and scoring with icm: the benchmarking results and strategies for improvement. *Journal of computer-aided molecular design*, 26(6):675–686, 2012.

[138] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[139] Duc D. Nguyen and Guo-Wei Wei. The impact of surface area, volume, curvature, and lennard–jones potential to solvation modeling. *Journal of computational chemistry*, 38(1):24–36, 2017.

[140] Duc D. Nguyen, Tian Xiao, Menglun Wang, and Guo-Wei Wei. Rigidity strengthening: A mechanism for protein–ligand binding. *Journal of chemical information and modeling*, 57(7):1715–1721, 2017.

[141] Duc Duy Nguyen, Zixuan Cang, Kedi Wu, Menglun Wang, Yin Cao, and Guo-Wei Wei. Mathematical deep learning for pose and binding affinity prediction and ranking in d3r grand challenges. *arXiv preprint arXiv:1804.10647*, 2018.

[142] Monica Nicolau, Arnold J Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, 2011.

[143] Jessica L Nielson, Jesse Paquette, Aiwen W Liu, Cristian F Guandique, C Amy Tovar, Tomoo Inoue, Karen-Amanda Irvine, John C Gensel, Jennifer Kloke, Tanya C Petrossian, et al. Topological data analysis for discovery in preclinical spinal cord injury and traumatic brain injury. *Nature communications*, 6:8581, 2015.

[144] Ippei Obayashi. Volume optimal cycle: Tightest representative cycle of a generator on persistent homology. *arXiv preprint arXiv:1712.05103*, 2017.

[145] Kristopher Opron, Kelin Xia, and Guo-Wei Wei. Fast and anisotropic flexibility-rigidity index for protein flexibility and fluctuation analysis. *The Journal of chemical physics*, 140(23):06B617_1, 2014.

[146] Kristopher Opron, Kelin Xia, and Guo-Wei Wei. Communication: Capturing protein multiscale thermal fluctuations, 2015.

[147] Angel R. Ortiz, M. Teresa Pisabarro, Federico Gago, and Rebecca C. Wade. Prediction of drug binding affinities by comparative binding energy analysis. *Journal of medicinal chemistry*, 38(14):2681–2691, 1995.

[148] Edward Ott, Celso Grebogi, and James A. Yorke. Controlling chaos. *Physical review letters*, 64(11):1196, 1990.

[149] Steve Y. Oudot. *Persistence theory: from quiver representations to data analysis*, volume 209. American Mathematical Society Providence, RI, 2015.

[150] Deepti Pachauri, Chris Hinrichs, Moo K Chung, Sterling C Johnson, and Vikas Singh. Topology-based kernels with application to inference problems in alzheimer's disease. *IEEE transactions on medical imaging*, 30(10):1760–1770, 2011.

[151] Jun-Koo Park, Robert Jernigan, and Zhijun Wu. Coarse grained normal mode analysis vs. refined gaussian network model for protein residue-level structural fluctuations. *Bulletin of mathematical biology*, 75(1):124–160, 2013.

[152] 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*, 12:2825–2830, 2011.

[153] Yunhui Peng, Lexuan Sun, Zhe Jia, Lin Li, and Emil Alexov. Predicting proteinDNA binding free energy change upon missense mutations using modified MM/PBSA approach: SAMPDI webserver. *Bioinformatics*, 34(5):779–786, 2018.

[154] Jose A Perea. Persistent homology of toroidal sliding window embeddings. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6435–6439. IEEE, 2016.

[155] Jose A Perea. Multiscale projective coordinates via persistent cohomology of sparse filtrations. *Discrete & Computational Geometry*, 59(1):175–225, 2018.

[156] Jose A Perea, Anastasia Deckard, Steve B Haase, and John Harer. Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC bioinformatics*, 16(1):257, 2015.

[157] Jose A Perea and John Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3):799–838, 2015.

[158] Janaina Cruz Pereira, Ernesto Raul Caffarena, and Cicero Nogueira dos Santos. Boosting docking-based virtual screening with deep learning. *Journal of chemical information and modeling*, 56(12):2495–2506, 2016.

[159] Douglas EV Pires, David B. Ascher, and Tom L. Blundell. DUET: a server for predicting effects of mutations on protein stability using an integrated computational approach. *Nucleic acids research*, 42(W1):W314–W319, 2014.

[160] Xose S Puente, Luis M Sánchez, Christopher M Overall, and Carlos López-Otín. Human and mouse proteases: a comparative genomic approach. *Nature Reviews Genetics*, 4(7):544, 2003.

[161] Lijun Quan, Qiang Lv, and Yang Zhang. Strum: structure-based prediction of protein stability changes upon single-point mutation. *Bioinformatics*, 32(19):2936–2946, 2016.

[162] V Robins, JD Meiss, and E Bradley. Computing connectedness: Disconnectedness and discreteness. *Physica D: Nonlinear Phenomena*, 139(3-4):276–300, 2000.

[163] Vanessa Robins. Towards computing homology from finite approximations. In *Topology proceedings*, volume 24, pages 503–532, 1999.

[164] Vanessa Robins, James D Meiss, and Elizabeth Bradley. Computing connectedness: An exercise in computational topology. *Nonlinearity*, 11(4):913, 1998.

[165] Michael Robinson. *Topological signal processing*. Springer, 2016.

[166] Lior Rokach and Oded Maimon. Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(4):476–487, 2005.

[167] Erik Rybakken, Nils Baas, and Benjamin Dunn. Decoding of neural data using cohomological learning. *arXiv preprint arXiv:1711.07205*, 2017.

[168] Tamar Schlick and Wilma K Olson. Trefoil knotting revealed by molecular dynamics simulations of supercoiled dna. *Science*, 257(5073):1110–1115, 1992.

[169] X Shi and P Koehl. Geometry and topology for modeling biomolecular surfaces. *Far East J Applied Math*, 50:1–34, 2011.

[170] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[171] Gurjeet Singh, Facundo Mémoli, and Gunnar E Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *SPBG*, pages 91–100, 2007.

[172] Gurjeet Singh, Facundo Memoli, Tigran Ishkhanov, Guillermo Sapiro, Gunnar Carlsson, and Dario L Ringach. Topological analysis of population activity in visual cortex. *Journal of vision*, 8(8):11–11, 2008.

[173] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[174] Bernadette J. Stolz, Heather A. Harrington, and Mason A. Porter. Persistent homology of time-dependent functional networks constructed from coupled time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(4):047410, 2017.

[175] DW Sumners. Knot theory and dna. In *Proceedings of Symposia in Applied Mathematics*, volume 45, pages 39–72, 1992.

[176] Akihiro Takiyama, Takashi Teramoto, Hiroaki Suzuki, Katsushige Yamashiro, and Shinya Tanaka. Persistent homology index as a robust quantitative measure of immunohistochemical scoring. *Scientific reports*, 7(1):14002, 2017.

[177] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[178] Christopher J. Tralie and Jose A. Perea. (quasi) periodicity quantification in video data, using topology. *SIAM Journal on Imaging Sciences*, 11(2):1049–1077, 2018.

[179] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[180] Thomas Unterthiner, Andreas Mayr, Günter Klambauer, and Sepp Hochreiter. Toxicity prediction using deep learning. *arXiv preprint arXiv:1503.01445*, 2015.

[181] Mikael Vejdemo-Johansson, Florian T Pokorny, Primoz Skraba, and Danica Kragic. Cohomological learning of periodic motion. *Applicable Algebra in Engineering, Communication and Computing*, 26(1-2):5–26, 2015.

[182] Hans FG Velec, Holger Gohlke, and Gerhard Klebe. Drugscorecsd knowledge-based scoring function derived from small molecule crystal data with superior recognition rate of near-native ligand poses and better affinity prediction. *Journal of medicinal chemistry*, 48(20):6296–6303, 2005.

[183] G Verkhivker, K Appelt, ST Freer, and JE Villafranca. Empirical free energy calculations of ligand-protein crystallographic complexes. i. knowledge-based ligand-protein interaction potentials applied to the prediction of human immunodeficiency virus 1 protease binding affinity. *Protein Engineering, Design and Selection*, 8(7):677–691, 1995.

[184] Izhar Wallach, Michael Dzamba, and Abraham Heifets. Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*, 2015.

[185] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[186] Bao Wang and Guo-Wei Wei. Object-oriented persistent homology. *Journal of computational physics*, 305:276–299, 2016.

[187] Bao Wang, Zhixiong Zhao, Duc D Nguyen, and Guo-Wei Wei. Feature functional theory–binding predictor (fft–bp) for the blind prediction of binding free energies. *Theoretical Chemistry Accounts*, 136(4):55, 2017.

[188] Bao Wang, Zhixiong Zhao, and Guo-Wei Wei. Automatic parametrization of non-polar implicit solvent models for the blind prediction of solvation free energies. *The Journal of chemical physics*, 145(12):124110, 2016.

[189] Changhao Wang, D'Artagnan Greene, Li Xiao, Ruxi Qi, and Ray Luo. Recent developments and applications of the mmpbsa method. *Frontiers in molecular biosciences*, 4:87, 2017.

[190] Renxiao Wang, Luhua Lai, and Shaomeng Wang. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *Journal of computer-aided molecular design*, 16(1):11–26, 2002.

[191] Sheng Wang, Siqi Sun, Zhen Li, Renyu Zhang, and Jinbo Xu. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS computational biology*, 13(1):e1005324, 2017.

[192] Guo-Wei Wei, Meng Zhan, and Choy Heng Lai. Tailoring wavelets for chaos control. *Phys. Rev. Lett.*, 89:284103, 2002.

[193] Catherine L. Worth, Robert Preissner, and Tom L. Blundell. SDMa server for predicting effects of mutations on protein stability and malfunction. *Nucleic acids research*, 39(suppl_2):W215–W222, 2011.

[194] Chengyuan Wu, Shiquan Ren, Jie Wu, and Kelin Xia. Weighted (co) homology and weighted laplacian. *arXiv preprint arXiv:1804.06990*, 2018.

[195] Kelin Xia, Xin Feng, Yiying Tong, and Guo Wei Wei. Persistent homology for the quantitative prediction of fullerene stability. *Journal of computational chemistry*, 36(6):408–422, 2015.

[196] Kelin Xia, Kristopher Opron, and Guo-Wei Wei. Multiscale multiphysics and multidomain modelsflexibility and rigidity. *The Journal of chemical physics*, 139(19):11B614_1, 2013.

[197] Kelin Xia and Guo-Wei Wei. Molecular nonlinear dynamics and protein thermal uncertainty quantification. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 24:013103, 2014.

[198] Kelin Xia and Guo-Wei Wei. Persistent homology analysis of protein structure, flexibility, and folding. *International journal for numerical methods in biomedical engineering*, 30(8):814–844, 2014.

[199] Kelin Xia and Guo-Wei Wei. Multidimensional persistence in biomolecular data. *Journal of computational chemistry*, 36(20):1502–1520, 2015.

[200] Kelin Xia and Guo-Wei Wei. Persistent topology for cryo-EM data analysis. *International Journal for Numerical Methods in Biomedical Engineering*, 31(8), 2015.

[201] Kelin Xia, Zhixiong Zhao, and Guo-Wei Wei. Multiresolution persistent homology for excessively large biomolecular datasets. *The Journal of chemical physics*, 143(13):10B603_1, 2015.

[202] Kelin Xia, Zhixiong Zhao, and Guo-Wei Wei. Multiresolution topological simplification. *Journal of Computational Biology*, 22(9):887–891, 2015.

[203] Zhexin Xiang and Barry Honig. Extending the accuracy limits of prediction for side-chain conformations1. *Journal of molecular biology*, 311(2):421–430, 2001.

[204] Lee-Wei Yang and Choon-Peng Chng. Coarse-grained models reveal functional dynamics-i. elastic network models–theories, comparisons and perspectives. *Bioinformatics and Biology Insights*, 2:BBI–S460, 2008.

[205] Yang Yang, Biao Chen, Ge Tan, Mauno Vihinen, and Bairong Shen. Structure-based prediction of the effects of a missense variant on protein stability. *Amino Acids*, 44(3):847–855, 2013.

[206] Yuan Yao, Jian Sun, Xuhui Huang, Gregory R Bowman, Gurjeet Singh, Michael Lesnick, Leonidas J Guibas, Vijay S Pande, and Gunnar Carlsson. Topological meth-

ods for exploring low-density states in biomolecular folding pathways. *The Journal of chemical physics*, 130(14):04B614, 2009.

[207] Shuangye Yin, Lada Biedermannova, Jiri Vondrasek, and Nikolay V Dokholyan. Medusascore: an accurate force field-based scoring function for virtual drug screening. *Journal of chemical information and modeling*, 48(8):1656–1662, 2008.

[208] Piotr Zgliczynski and Konstantin Mischaikow. Rigorous numerics for partial differential equations: The kuramotosivashinsky equation. *Foundations of Computational Mathematics*, 1(3):255–288, 2001.

[209] Shunhong Zhang, Qian Wang, Yoshiyuki Kawazoe, and Puru Jena. Three-dimensional metallic boron nitride. *Journal of the American Chemical Society*, 135(48):18216–18221, 2013.

[210] Zhe Zhang, Maria A Miteva, Lin Wang, and Emil Alexov. Analyzing effects of naturally occurring missense mutations. *Computational and mathematical methods in medicine*, 2012, 2012.

[211] Zheng Zheng and Kenneth M Merz Jr. Ligand identification scoring algorithm (lisa). *Journal of chemical information and modeling*, 51(6):1296–1306, 2011.

[212] Zheng Zheng, Melek N Ucisik, and Kenneth M Merz. The movable type method applied to protein–ligand binding. *Journal of chemical theory and computation*, 9(12):5526–5538, 2013.

[213] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Clustered multi-task learning via alternating structure optimization. In *Advances in neural information processing systems*, pages 702–710, 2011.

[214] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Malsar: Multi-task learning via structural regularization. *Arizona State University*, 21, 2011.

[215] Afra Zomorodian. Fast construction of the Vietoris-Rips complex. *Computers & Graphics*, 34(3):263–271, 2010.

[216] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.