PLOS ONE

RESEARCH ARTICLE

Weighted-persistent-homology-based machine learning for RNA flexibility analysis

Chi Seng Pun[1]*, **Brandon Yung Sin Yong**[1], **Kelin Xia**[1,2]*

1 Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, Singapore, **2** School of Biological Sciences, Nanyang Technological University, Singapore, Singapore

* cspun@ntu.edu.sg (CSP); xiakelin@ntu.edu.sg (KX)

Citation: Pun CS, Yong BYS, Xia K (2020) Weighted-persistent-homology-based machine learning for RNA flexibility analysis. PLoS ONE 15 (8): e0237747. https://doi.org/10.1371/journal.pone.0237747

Editor: Ning Cai, Beijing University of Posts and Telecommunications, CHINA

Received: May 3, 2020

Accepted: August 1, 2020

Published: August 21, 2020

Data Availability Statement: S1 Table lists the results for PCC of each RNA chain in training set achieved by the best optimal RF. All our codes can be downloaded from https://github.com/cspun/WPHML-B-Factor-Prediction.

Funding: This research is supported by Nanyang Technological University Startup Grants M4081840 and M4081842 to CSP, Data Science and Artificial Intelligence Research Centre@NTU M4082115 to CSP, and Singapore Ministry of Education Academic Research Fund Tier 1 RG31/18, RG109/19, and Tier 2 MOE2018-T2-1-033 to KX. There

Abstract

With the great significance of biomolecular flexibility in biomolecular dynamics and functional analysis, various experimental and theoretical models are developed. Experimentally, Debye-Waller factor, also known as B-factor, measures atomic mean-square displacement and is usually considered as an important measurement for flexibility. Theoretically, elastic network models, Gaussian network model, flexibility-rigidity model, and other computational models have been proposed for flexibility analysis by shedding light on the biomolecular inner topological structures. Recently, a topology-based machine learning model has been proposed. By using the features from persistent homology, this model achieves a remarkable high Pearson correlation coefficient (PCC) in protein B-factor prediction. Motivated by its success, we propose weighted-persistent-homology (WPH)-based machine learning (WPHML) models for RNA flexibility analysis. Our WPH is a newly-proposed model, which incorporate physical, chemical and biological information into topological measurements using a weight function. In particular, we use local persistent homology (LPH) to focus on the topological information of local regions. Our WPHML model is validated on a well-established RNA dataset, and numerical experiments show that our model can achieve a PCC of up to 0.5822. The comparison with the previous sequence-information-based learning models shows that a consistent improvement in performance by at least 10% is achieved in our current model.

1 Introduction

Biomolecular functions usually can be analyzed by their structural properties through quantitative structure-property relationship (QSPR) models (or quantitative structure-activity relationship (QSAR) models). Among all the structural properties, biomolecular flexibility is of unique importance, as it can be directly or indirectly measured by experimental tools. Debye-Waller factor or B-factor, which is the atomic mean-square displacement, provides a quantitative characterization of the flexibility and rigidity of biomolecular structures. With the strong relationship between structure flexibility and functions, various theoretical and computational methods have been proposed to model the flexibility of a biomolecular. Such methods include

was no additional external funding received for this study.

Competing interests: The authors have declared that no competing interests exist.

molecular dynamics (MD) [1], normal mode analysis (NMA) [2–5], graph theory [6], elastic network models (ENMs) [7–12], Gaussian network model (GNM) [7, 8], anisotropic network model (ANM) [9], local density model (LDM) [13], local contact model (LCM) [14], weighted contact number (WCN) model [15], molecular nonlinear dynamics [16], stochastic dynamics [17] and flexibility-rigidity index (FRI) [18, 19]. In these models, biomolecular structures are usually modeled as graphs or networks, and a deterministic relationship is established between experimental B-factors and certain network properties, such as node degree, centrality, pseudo-inverse Laplacian matrix and pseudo-inverse Hessian matrixes.

Other than the above deterministic models, data-driven machine learning models are also considered in flexibility analysis [20–29], thanks to the accumulation of ever-increasing experimental data. In these learning models, biomolecular genetic, epigenetic, evolutional and structural information are extracted and used as features in machine learning models, such as support vector machine (SVM), random forest (RF), gradient boost tree (GBT) and artificial neural network (ANN). Among these learning models, an evolution-information-based learning model has been used in RNA flexibility analysis [27]. In this model, position-specific iterative basic local alignment search tool (PSI-BLAST) [30] is considered for homologous sequence identification. For each sample, a position-specific scoring matrix (PSSM) profile is calculated. The properties of the matrix are used as feature vectors and fed into various machine learning models. A Pearson correlation coefficient (PCC) value of 0.5028 between the test and predicted B-factor values has been achieved [27]. Further, more features from sequence-based information, including nucleotide acid one hot vector, predicted secondary structure, and predicted solvent accessibility, are considered in RNAbval model [29]. Combined with random forest, RNAbval can significantly improve the performance and achieve a PCC of 0.6061 [29]. Moreover, multiscale weighted colored graphs (MWCGs) based learning model is proposed to blindly predict protein B-factors [28]. These MWCGs provide a series of graph features, that characterize the intrinsic flexibility of protein structure very well. The model can be used in the blind prediction of protein B-factor with a PCC value of 0.66.

More recently, a persistent-homology (PH)-based machine learning model is proposed [28]. In this model, PH, which is a tool for data simplification and dimension reduction, is used for protein structure featurization. Different from conventional topology tools, which tend to oversimplify structural information and thus can only be used in qualitative modeling, PH manages to retain the important geometrical properties through a filtration process. Essentially, a series of simplicial complexes are generated and their topological information are characterized by homology groups [31, 32]. The "birth" and "death" of these homology generators are recorded and can be represented in either persistent diagrams (PDs) or persistent barcodes (PBs) [33]. Further, atom-specific PH and element-specific PH are used to classify the structures into different point sets with more detailed structural information [28]. Moreover, two types of matrices, one based on Euclidean distance and another on multiscale interaction, are considered. Machine learning models can achieve a PCC value up to 0.73 for a dataset of 364 proteins using the topological features extracted from their corresponding PBs [28].

Motivated by the great success of the PH-based machine learning models in protein B-factor prediction, we propose weighted-persistent-homology (WPH)-based machine learning (WPHML) models for RNA B-factor prediction. WPH incorporates physical, chemical and biological information into the topological measurements with a weight function [34, 35]. In general, different weights are assigned to k-simplexes with k starting from 0. In particular, by assigning a weight value of 0 or 1 to each point, we can naturally arrive at a local PH model and element-specific PH model [36–38]. Similarly, an interactive PH is derived by assigning weight values only to the edges between the interaction atoms [36–38]. More importantly, a

weighted boundary operator can be designed to embed higher-level relations into topological invariants.

In this paper, we only consider weight values on points, i.e., atoms, to select a local region around a certain atom-of-interest, whose flexibility is to be evaluated. PH analysis is then applied to the selected atoms within the local region. Features will be generated from the corresponding PBs using a binning approach before the features are fed into learning models. To test and compare the performance of our models, the same dataset and preprocessing steps as described by Guruge et al. [27] are used. Our results show that WPH-based learning models can consistently outperform this sequence-based model in RNA B-factor prediction [27]. However, it should be noticed that higher accuracy can be achieved with more sophisticated feature engineering of sequence information [29]. A combination of features from both structure and sequence may achieve even better accuracy. Essentially, the importance of featurization and feature engineering in material, chemical and biological learning models can not be overemphasized.

The paper is organized as follows. Weighted persistent homology based featurization and the combination with different types of machine learning approaches are introduced in Section "Methodology". In Section "Results", we present the findings of our numerical results, including the comparison between the benchmark and our WPHML approaches and the sensitivity analysis of the model settings. The paper ends with a conclusion.

2 Methodology

In this section, we give a brief introduction to persistent homology and weighted persistent homology. Then, topology-based featurization is discussed in great details. After that, we briefly discuss the four main learning models that are considered.

2.1 Topology-based feature engineering

Data-driven sciences are widely regarded as the fourth paradigm that can fundamentally change sciences and pave the way for a new industrial revolution [39]. The past decade has witnessed a great boom of learning models in areas such as data mining, natural language processing, image analysis, animation and visualization. In contrast, the application of learning models in materials, chemistry and biology is far behind this trend.

One of the most important reasons is featurization or feature engineering [40–42]. Compared to text, image or audio data, molecular structural data from material, chemistry and biology are highly irregular and differ greatly from each other. Essentially, each molecule can have not only different numbers or types of atoms but also very different and complicated spatial connectivity. The structural complexity and high data dimensionality have significantly hampered the progress of the application of learning models in these fields.

To solve the problems, various ways of featurization have been proposed and a series of molecular descriptors (features) are generated. In general, molecular descriptors can be divided into three groups, i.e., structural measurements, physical measurements, and genetic features [40–42]. Structural measurements come from structural geometry, chemical conformation, chemical graph, structure topology, etc. Physical descriptors come from molecular formula, hydrophobicity, steric properties, and electronic properties, etc. Genetic features can be derived from gene sequences, gene expression, genetic interaction, evolution information, epigenetic information, etc.

Recently, persistent homology has been used in molecular characterization. With the unique attribute that balances geometric complexity and topological simplification, PH provides a unique structure featurization that can be naturally combined with machine learning

models. PH-based learning models have been successfully used in various aspects of drug design [36–38], including protein-ligand binding affinity prediction, solubility, toxicity, and partition coefficient. More recently, PH-based learning models have been used in protein B-factor blind prediction and a remarkable high accuracy is obtained [28]. These great successes have inspired us to propose WPHML for RNA B-factor prediction. To have a better understanding of our WPHML, a brief introduction of PH and WPH is given below.

2.1.1 Persistent Homology (PH). General speaking, persistent homology can be analyzed from three aspects—graph and simplicial complex; geometric measurements and topological invariants; and a bridge between geometry and topology.

Graph and simplicial complex. A simplex is a generalization of the notion of a triangle or tetrahedron to arbitrary dimensions and it is the building block for the simplicical complex. A simplicial complex K is a finite set of simplices that satisfy two essential conditions. First, any face of a simplex in K is also in K. Second, the intersection of any two simplices in K is either empty or shares faces. Geometrically, a 0-simplex is a vertex, a 1-simplex is an edge, a 2-simplex is a triangle, and a 3-simplex represents a tetrahedron. Graphs and networks, composed of only vertices and edges, are special cases of simplicial complexes.

Geometric measurements and topological invariants. Geometry models consider geometrical information such as coordinates, distances, angles, areas, various curvatures and vector bundles. Graph models study measurements such as degree, shortest path, clique, cluster coefficient, closeness, centrality, betweenness, Cheeger constant, modularity, graph Laplacian, graph spectral, Erdős number and percolation. These geometric and graph descriptors characterize local and non-intrinsic information very well. In contrast, PH explores the intrinsic connectivity information measured by Betti number, which is a type of topological invariants that is unchanged under deformation. Geometrically, we can regard β_0 as the number of isolated components; β_1 the number of one-dimensional loops, circles, or tunnels, and; β_2 the number of two-dimensional voids or holes.

Bridge between geometry and topology. Different from geometry and topology models, PH manages to incorporate geometrical measurements into topological invariants, thus provides a balance between geometric complexity and topological simplification. The key idea of PH is a process called filtration [31, 32]. By varying the value of a filtration parameter, a series of simplicial complexes are generated. These nested simplicial complexes encode topological information of a structure from different scales. Some topological invariants "live longer" in these simplicial complexes whereas others disappear very quickly when the filtration value increases. In this way, topological invariants can be quantified by their "lifespans" or "persisting times", which are directly related to geometric properties. A PB can be generated from the birth, death and persistence of the topological invariants of the given dataset [33]. An example of PBs can be found in Fig 1.

2.1.2 Weighted Persistent Homology (WPH). Recently, we have systematically studied WPH models and their applications in biomolecular data analysis [34, 35, 43]. General speaking, we can define weight values, which represent physical, chemical and biological properties, on k-simplices, such as vertices (atom centers), edges (bonds), or higher order simplices (motif or domains). That is to say WPH can be characterized into three major categories—vertex-weighted [44–47]; edge-weighted [36, 38, 48, 49], and; general-simplex-weighted models [35, 50, 51]. These weighted values can be viewed as certain distance measurements, and PH analysis can be applied. In this way, these properties are naturally incorporated into topological measurements.

On the other hand, we can define a weighted boundary map, which can embed deeper interaction relationships into a topology. Note that to ensure the consistency of the homology definition, weight values on different simplexes need to satisfy certain constraints [35, 50, 51].

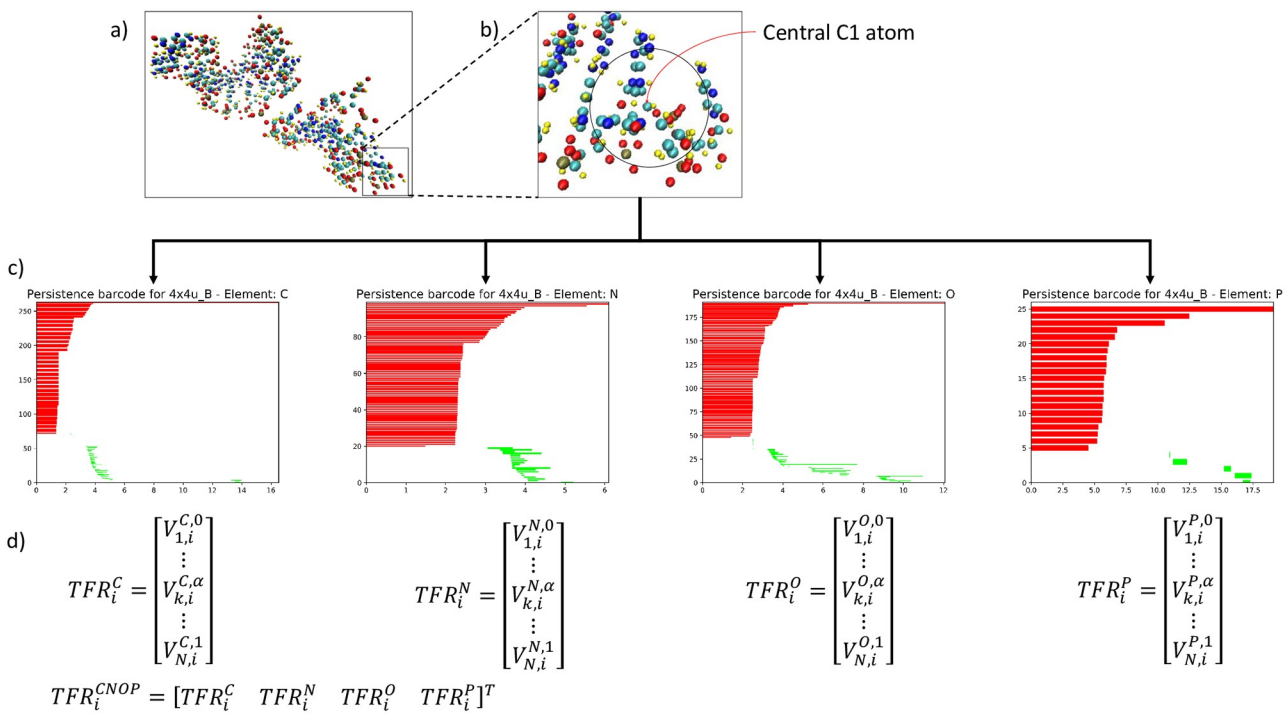

Fig 1. (a) Chain B of RNA 4X4U with each element (C, N, O, P and H) highlighted in a different colour. (b) In order to apply LPH, a local region and all the atoms within the local region for each central C1 atom is determined. However, only the elements of C, N, O and P are considered. (H highlighted in yellow is ignored) (c) Localized ESPH is applied and a persistent barcode is plotted for each of the four elements. (d) Binning approach is applied to each PB and the corresponding topological feature representation (TFR) of a sample *i* is obtained. The feature vector of four-element is a concatenated vector of all four TFRs. The feature vectors are then used for ML training.

https://doi.org/10.1371/journal.pone.0237747.g001

Previous PH models, including element-specific PH (ESPH) [36–38] and local persistent homology (LPH) [34] can be regarded as special cases of vertex-weighted PH. The multi-level PH, interactive PH, and electrostatic persistence [38] are essentially edge-weighted PH.

In this paper, LPH is used for RNA local structure characterization. Biologically, an RNA chain is made up of a set of nucleotides, in which the size of the set of nucleotides can range from low tens to a few thousands and above. In our LPH model, only atoms that are located within a specific Euclidean cut-off distance *E* from each C1 atom in each chain are considered. Note that only the B-factor for C1 atoms are predicted and evaluated against experimental data in the same manner as by Guruge et al. [27].

As a nucleotide constitutes of heavy atoms C, N, O, and P, in our LPH model, the localized ESPH is considered by using each of the four elements individually. That is, each element would eventually generate its own unique set of topological features representation for the specific local region. Note that for each ESPH, the central C1 atom is always included. Their topological features are drastically different from one another as shown in Fig 1(c). Indeed, ESPH is capable of retaining crucial biological information during topological simplification [52].

2.1.3 Topological Features Representation (TFR). Results from PH analysis are pairs of "birth" and "death" values for different dimensions of Betti numbers. They can be represented as PDs or PBs. However, PH results are notorious for meaningful metric definition and statistic interface. Various methods are proposed [53] including barcode statistics, tropical coordinates, binning approach, persistent image, persistent landscapes and image representations to construct topological features.

In this paper, we only consider topological features constructed using a binning approach [53]. More specifically, the filtration interval [0, F] is divided into N bins of equal size f. The number of barcodes which are located on each bin are then counted and used as feature vector [54, 55]. More specifically, the feature vector of a sample i is defined as:

$$\mathbf{V}_i = ||\{(b_j, d_j) \in \mathbf{B}(\alpha, D) | b_j \leq kF/N \leq d_j\}|| \qquad 1 \leq k \leq N$$

where $\|\cdot\|$ is cardinality i.e., the number of elements, of sets. Here b_j and d_j are referring to birth and death of bar j. $\mathbf{B}(\alpha, D)$ is referring to the collections of barcodes with α referring to the selection of atoms and D referring to the dimension of the Betti numbers. Essentially, for each C1 atom, we have a $N * 1$ topological vector for each element and dimension of the Betti numbers.

2.2 Machine Learning (ML) models

After the topological features are represented as a feature vector, it can serve as input to predict B-factor values with ML algorithms. We consider four main ML models, namely regularized linear regression, tree-based methods (including random forest and extreme gradient boosting), support vector regression, and artificial neural networks. All our ML algorithms are implemented in Python (packages mentioned below refer to the packages in Python).

In the following descriptions of the ML models, we assume that we train our models with n data $\{(x_i, y_i)\}_{i=1}^n$, where $y_i \in \mathbb{R}$ is the normalized B-factor value of the ith sample (details of B-factor normalization will be discussed in Section Results), $x_i \in \mathbb{R}^p$ is the structured topological feature vector of the ith sample, and p is the number of structured features. Conventionally, we denote by $\hat{y}$ the predicted normalized B-factor value of a sample.

2.2.1 Regularized linear regression. Linear regression is a straightforward yet efficient approach to model the relationship between a quantitative response and features. The incorporation of regularization can effectively address the high dimensionality setting where the number of features is larger than the sample size. The variable selection feature of the regularized linear regression makes it particularly suitable for our task as our feature vector is usually lengthy. The general formulation of regularized linear regression can be read as the following regularized minimization problem:

$$\min_{\beta_0 \in \mathbb{R}, \beta \in \mathbb{R}^p} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^\top \beta)^2 + \mathcal{R}_\alpha(\beta), \tag{1}$$

where $\mathcal{R}_\alpha(\cdot)$ is a regularization term. Once we obtain the minimizer of (1), denoted by $(\hat{\beta}_0, \hat{\beta})$, we predict the B-factor value of the test data with structured feature vector x by $\hat{y} = \hat{\beta}_0 + x^\top \hat{\beta}$. The specification of $\mathcal{R}_\alpha$ determines the shrinkage of $\hat{\beta}$ and statistical accuracy of $\hat{y}$ [56–60].

In our study, we consider the two typical choices of $\mathcal{R}_\alpha$, namely L2-norm ($\alpha\|\beta\|_2^2$) and L1-norm ($\alpha\|\beta\|_1$), where α is the tuning parameter that strikes the balance between efficiency and regularization. The regression problem with these two types of regularization are also known as **Ridge regression** [56] and **least absolute shrinkage and selection operator (LASSO)** [57], respectively. The advantage of LASSO over Ridge regression is its variable selection feature, which has strong interpretable power. From the LASSO results, one can tell which part of the structural information of the element is important. Both Ridge regression and LASSO are implemented with the package "`scikit-learn`" [61].

2.2.2 Tree-based methods. Classification And Regression Tree (CART) [62] or decision tree learning is a common method used in ML. Many variations of trees have been proposed with the pruning and ensemble methods. The simple and interpretable tree-based methods

have the advantage of handling high-dimensional data without further adjustments. It addresses our concern with the lengthy feature vector deduced from topological features representation. Among many candidates of tree-based methods, we consider **Random Forest (RF)** [63, 64] and **Extreme Gradient Boosting (XGBoost)** [65].

RH. RH is an ensemble learning method that creates a variety of decision (regression) trees independently during training, where each decision tree is constructed using a random subset of the features as split candidates. During the training of each tree, the split at each node is determined by the least-square method. In other words, for each region of each tree, we predict the B-factor value with the average of the B-factor values of the samples fallen in the region. In a regression RF, the final prediction is the average of the predicted values of all individual trees. In the implementation of ensemble trees, the number of trees, minimum number of samples at each leaf node, and the number of split candidates in each splitting, i.e., parameter *mtry*, are all tuning parameters. In our application of RF, we choose $mtry = \lceil \sqrt{p} \rceil$ following Breiman [64] and tune the other two hyperparameters. The RF is also implemented with the package "`scikit-learn`" [61].

XGBoost. Has been one of the popular ML tools used by the winning teams of many ML challenges and competitions, such as the Netflix prize [66] and various Kaggle challenges. Instead of computing the average output of all the individual trees as in a regression RF, each tree in XGBoost contributes a certain value which is added up iteratively. Such additive training or gradient boosting allows the predicted values to approach the actual values as closely as possible. In our study, we tune the number of trees and the maximum tree depth, which affects the number of leaves in the trees, while the remaining parameters are set default as defined by XGBoost. XGBoost is implemented with the package "`xgboost`" [65].

2.2.3 Support Vector Regression (SVR). SVR [67], as a version of the well-known support vector machine (SVM) [68] for regression, is another popular ML algorithm. The goal of an SVM model is to find a function $\beta_0 + x^\top \beta$ that has at most ϵ deviation from the actual target values y_i for all the training data while trying to be as flat as possible [69]. Sometimes, the convex optimization problem is not feasible and a "soft margin" loss function is introduced [68]. The SVR model (β_0, β) is determined by the following minimization problem:

$$\min_{\beta_0,\beta,\xi,\xi^*} \frac{1}{2}\|\beta\|_2^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \quad s.t. \quad \begin{cases} y_i - \beta_0 - x_i^\top \beta_i & \leq & \epsilon + \xi_i \\ \beta_0 + x_i^\top \beta_i - y_i & \leq & \epsilon + \xi_i^*, \\ \xi_i,\ \xi_i^* & \geq & 0 \end{cases}$$

where ξ and ξ_i^* are slack variables to cope with the otherwise infeasible constraints of the optimization problem and the hyperparameter C determines the trade-off between the efficiency and the amount up to which deviation larger than ϵ is tolerable. Typically, we adopt kernel methods to transform the input features from a lower to a higher dimensional space, where a linear fit is feasible. Common choices of kernel include polynomial kernel, Gaussian kernel, and radial basis function (RBF) kernel. In our study, we have opted to use RBF kernel, i.e., $K(x, x') = \exp(-\gamma\|x - x'\|_2^2)$, in our SVR model. The SVR is implemented with the package "`scikit-learn`" [61].

2.2.4 Artificial Neural Network (ANN). ANN has been proved to be capable of learning to recognize patterns or categorize input data after training on a set of sample data from the domain [70]. The ability to learn through training and to generalize broad categories from specific examples is the unique intelligence for ANN [71]. Different from other ML algorithms, ANN requires the user to determine the architecture of the network, such as the number of hidden layers, the number of nodes, and the specification of activation function in each layer.

The hidden layers in ANN architecture allow the ANN to deal with nonlinear and complex problems more robustly and therefore can operate on more interesting problems [72]. The number of hidden layers enables a trade-off between smoothness and closeness of fit [73]. The number of nodes within a hidden layer determines the trade-off between training time and training accuracy. The weights of each layer are optimized via the use of a learning algorithm called "backpropagation" [74]. Since the ANN will involve the learning of a vast amount of weights, from the statistical perspective, the overfitting problem arises. We adopt a recently proposed regularization technique called "dropout" [75], which is empirically proven magical. This approach also addresses the curse of dimensionality due to the lengthy topological feature vector in our study.

In our study, the number of hidden layers, number of nodes in each hidden layer, and the number of epochs are treated as hyperparameters. The hidden and output activation functions are set as sigmoid and leaky ReLU functions respectively. The dropout rate is set to 20% and the remaining hyperparameters are set to default values as defined by the package. ANN is implemented with the package "`keras`" [76].

2.3 Model setting

RNA dataset. We consider the same RNA dataset and data preprocessing by Guruge et al. [27]. The chains are randomly split in the same manner with 75% of the chains go into a training set and 25% go into the test set. The B-factor of each nucleotide is represented by its C1 atom.

B-factor normalization and outlier detection. The values of B-factors may differ significantly from chain to chain due to reasons such as a relatively small number of residues in a protein chain or differences in refinement methods used [77]. Thus, the B-factors of each chain are normalized to have zero mean and unit variance [27]. The range of normalized B-factor falls approximately between -3.00 and 4.00. Further, before the raw B-factors are normalized, values of outliers are first detected and removed using a median-based approach [78]. This is to eliminate raw B-factor values that are located on the extreme ends of the distribution.

Hyperparameter setting. In our dataset, cut-off distance E, F/E ratio, and bin size f are the hyperparameters to be optimized. We chose the value of E to be in the range from 10 Å to 45 Å with a stepsize of 5 Å, i.e., E = {10 Å, 15 Å, 20 Å, 25 Å, 30 Å, 35 Å, 40 Å, 45 Å}. The filtration interval F is defined such that the ratio of F/E is between 0.5 to 1.0 with a stepsize of 0.1, i.e., F/E = {0.5, 0.6, 0.7, 0.8, 0.9, 1.0}. Bin size f is chosen to be in the range from 0.15 Å to 1.50 Å, i.e., f = {0.15 Å, 0.50 Å, 1.00 Å, 1.50 Å}. A total of 32,823 PBs are generated based on the Vietoris-Rips complex for each combination of element type, E and F/E ratio. Both "`GUDHI`" [79] and "`Dionysus`" [80] packages are used.

To determine the optimal hyperparameter values for each ML model, we conduct a five-fold cross validation (CV) using the training set. Specifically, the training set is randomly divided into five folds with a similar number of chains. In each fold, for each combination of the hyperparameters, we find the predicted B-factor values for the left-out training set with the ML model trained by the remaining training set. The optimal hyperparameter set maximizes the out-of-sample PCC between the predicted and actual values across all folds. The optimal hyperparameter values for each ML model can be found in Table 1.

Once the hyperparameter values of the dataset and models have been optimized, the trained models are evaluated using a test set that was non-overlapping with the training set. The PCC between the predicted and actual normalized B-factor values in the test set is calculated for

Table 1. Optimal hyperparameter values for each ML model. ESPH—χ and ESPH—CNOP refer to the optimal hyperparameter values of the ML model under single-element and four-element-combined dataset respectively.

ML model	Hyperparameters	ESPH—χ	ESPH—CNOP
Ridge	Alpha	500	500
LASSO	Alpha	0.01	1
RF	No of trees	500	2000
	No of min samples at nodes	5	5
XGBoost	No of trees	50	50
	Tree depth	3	3
SVM	Kernel	RBF	RBF
	Gamma	0.01	0.001
	C	0.1	0.1
ANN	No of hidden layers	4	3
	No of nodes per hidden layer	68	900
	Activation type for hidden layer	Sigmoid	Sigmoid
	Dropout rate	20%	20%
	No of epochs	15	10

https://doi.org/10.1371/journal.pone.0237747.t001

each model

$$PCC(y_i, \hat{y}_i) = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}},$$

where $\hat{y}_i$ is the predicted i-th B-factor value, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\bar{\hat{y}} = \frac{1}{n}\sum_{i=1}^{n} \hat{y}_i$.

3 Results

In this section, we demonstrate the performance of our WPHML model. Table 2 shows the best performance achieved by each ML model on test set. The best performance reported by Guruge et al. [27] is used as a benchmark performance (benchmark PCC = 0.5028). The

Table 2. Best test set performance for each ML model using the optimal hyperparameter values for dataset and ML model. PSSM stands for Position Specific Scoring Matrix, which is the benchmark performance by Guruge et al. [27].

Feature type	ML model	Test set PCC	Improvement (%)
PSSM	SVM (RBF)	0.5028	
ESPH—O	Ridge	0.4283	-14.8%
ESPH—O	LASSO	0.4667	-7.2%
ESPH—P	**RF**	**0.5788**	**15.1%**
ESPH—P	XGBoost	0.5748	14.3%
ESPH—P	SVM (RBF)	0.5520	9.8%
ESPH—P	ANN	0.5732	14.0%
ESPH—CNOP	Ridge	0.4849	-3.6%
ESPH—CNOP	LASSO	0.4157	-17.3%
ESPH—CNOP	**RF**	**0.5822**	**15.8%**
ESPH—CNOP	XGBoost	0.5657	12.5%
ESPH—CNOP	SVM (RBF)	0.5560	10.6%
ESPH—CNOP	ANN	0.5609	11.6%
PSSM, etc [29]	**RF**	**0.6061**	**20.5%**

https://doi.org/10.1371/journal.pone.0237747.t002

Table 3. Best test performance conditions.

Element type(s)	ML model	Cut-off (Å)	F/E ratio	Bin size (Å)	PCC
ESPH—O	Ridge	25	0.7	1.50	0.4283
ESPH—O	LASSO	25	0.5	0.50	0.4667
ESPH—P	RF	45	0.7	0.15	0.5788
ESPH—P	XGBoost	45	0.9	1.00	0.5748
ESPH—P	SVM (RBF)	40	0.5	1.00	0.5520
ESPH—P	ANN	45	1.0	1.00	0.5732
ESPH—CNOP	Ridge	35	0.6	0.50	0.4849
ESPH—CNOP	LASSO	25	0.5	0.50	0.4157
ESPH—CNOP	RF	40	0.5	0.15	0.5822
ESPH—CNOP	XGBoost	45	0.7	0.15	0.5657
ESPH—CNOP	SVM (RBF)	35	0.5	0.15	0.5560
ESPH—CNOP	ANN	45	0.5	0.15	0.5609

https://doi.org/10.1371/journal.pone.0237747.t003

conditions in which the best test performances are obtained can be found in Table 3. In RNAbval model, sequence-based information, including PSSM, nucleotide acid one hot vector, predicted secondary structure, and predicted solvent accessibility, are considered [29]. By the use of extensive sequence-based features, they can achieve better result.

For both single-element and four-element-combined models, it can be seen that WPHML models are able to consistently outperform the evolution-based method (PSSM) by at least approximately 10% with only the exception of linear regression models (Ridge and LASSO). Among all the models, RF achieves the best result with PCC = 0.5788 (15.1% improvement). Moreover, the performance of the RF model further improves to 0.5822 when the topological features for all four elements were used, which is about 15.8% improvement.

The comparison between the results from single-element and four-element-combined models shows that generally there is no significant improvement. In fact, SVM improves only slightly (approximately 0.8%), while XGBoost and ANN models even show some small reduction of accuracy (1.8% and 2.4% respectively). The results seem to be different from previous studies that concluded that element-specific models always deliver better results [28, 36–38]. Note that previous models are based on protein structures.

Comparably speaking, RNA structures are more regular and relatively simple. Similar topological features may be embedded in different types of element models. In this way, the additional features do not incorporate new information, instead they will contribute more noises, which causes the drop in performances. Noted that the best test performance of all the models except linear regression using a single element are all based on element P.

Effect of Euclidean cut-off distance

Fig 2 shows the effect of cut-off distance. It can be seen that the PCCs of the fivefold cross validation using the topological features from both element P and all four elements gradually improve and eventually plateaus off at approximately 35 Å. Note that 35 Å is larger than the generally used cut-off distance in the Gaussian network model, anisotropic network model, and other graph-based models, which are usually around 8 Å to 20 Å.

One of the reasons that larger cut-off distance delivers good results is that our predicted PCC values are predominantly determined by the several larger-sized RNAs. From S1 Table, it can be seen that there is a wide range of chain PCC distribution in the test set, which ranges from -0.50 to 0.80 although our RF model has a fairly good PCC of 0.5822. Moreover,

approximately 70% of the test data points come from 4 out of the 34 chains, of which these 4 chains have a chain PCC higher than the overall PCC achieved by the RF model. With that said, the performance of the test set is heavily based on these 4 chains. As long as the predictions on these 70% data points continue to improve, the overall performance of the model would continue to improve although there may be a reduction in performance on the remaining 30% of data points. This indicates that the evaluation method [27, 28] may have certain limitations. However, for a fair comparison, we still use it in the current paper.

Effect of F/E ratio

Fig 3 shows the effect of F/E ratio on the fivefold cross validation performance. At a low cut-off distance, the improvement in the fivefold cross validation performance improves more significantly when F/E ratio increases from 0.5 to 0.7. Beyond 0.7, the improvement in performance is very minimal. However, at a large cut-off distance, the performance is rather consistent from 0.5 to 1.0. This shows that the F/E ratio is not a significant hyperparameter to generate the dataset and it is more than sufficient to use an F/E ratio of 0.5 so as to minimize the number of unnecessary features generated especially as a large cut-off distance is required as discussed previously.

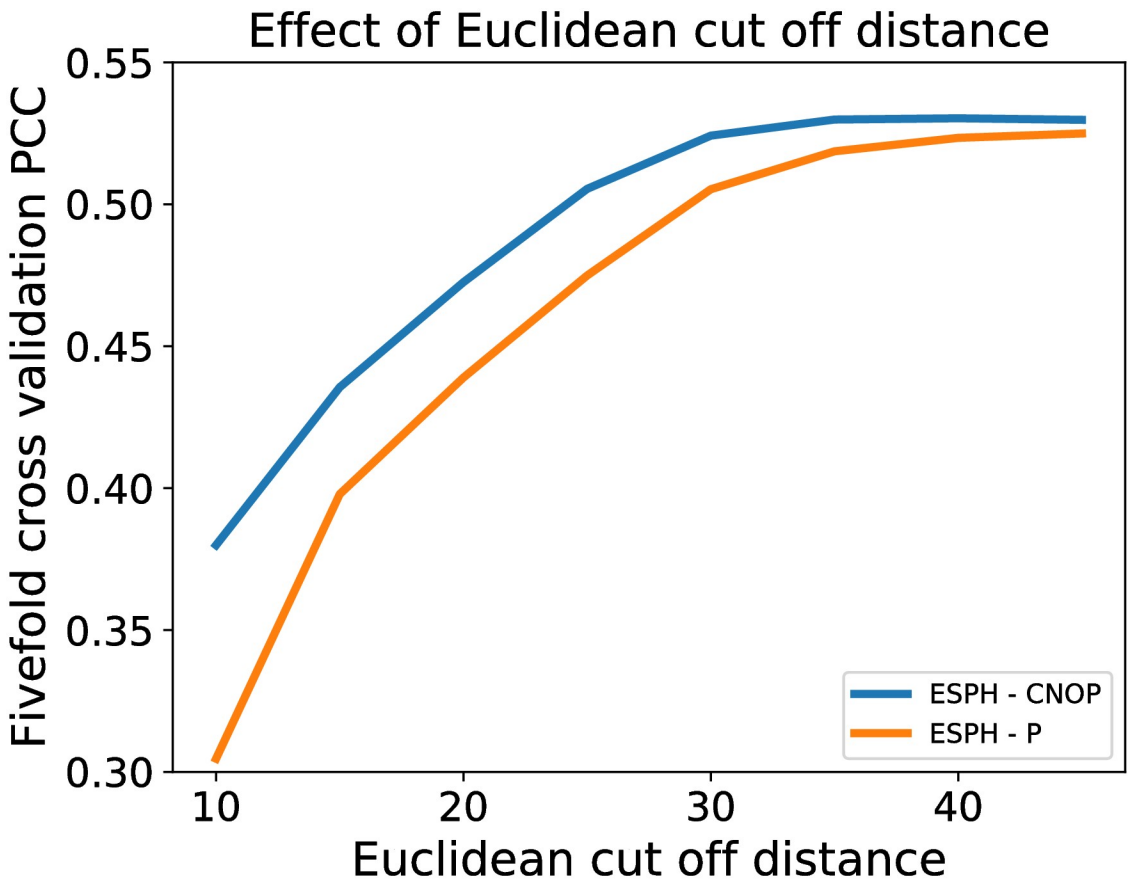

Fig 2. Effect of Euclidean cut-off distance on RF using the topological features from element P and all four elements with a fixed F/E ratio of 1.0 and bin size of 0.15Å.

https://doi.org/10.1371/journal.pone.0237747.g002

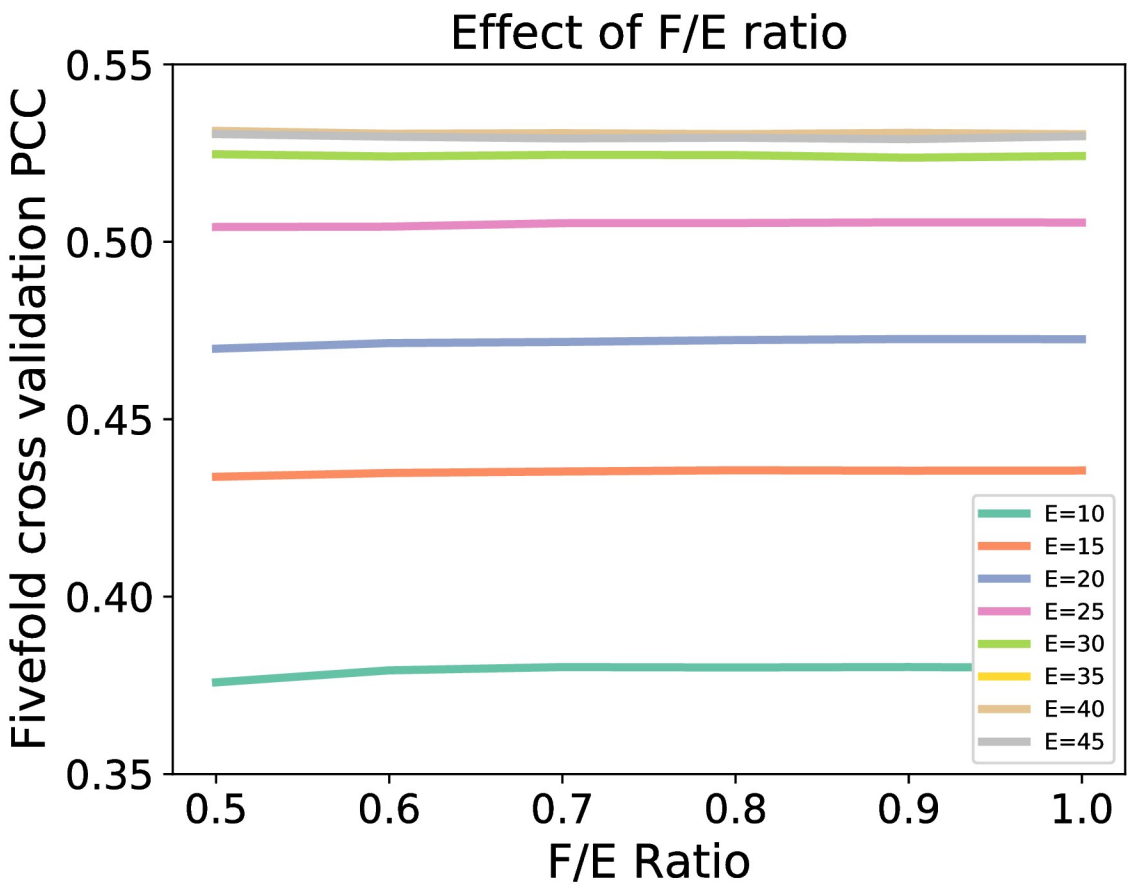

Fig 3. Effect of F/E ratio on RF using the topological features from all four elements and bin size of 0.15Å.

https://doi.org/10.1371/journal.pone.0237747.g003

Effect of bin size

Fig 4 shows the changes in five-fold CV performance with respect to bin size. As the bin size decreases from 1.5 Å to 0.15 Å, the performance improves for all Euclidean cut-off distance. This indicates that with a smaller bin size, the finer details of topological features are detected especially topological invariants that exist for a very short moment. The geometric information, embedded in the topological invariants, are key to the success of WPHML models.

4 Conclusion

In this paper, we propose the weighted-persistent-homology-based machine learning (WPHML) models and use them in the RNA B-factor prediction. We found that our WPHML models can consistently deliver a better performance than the evolution-based learning models. In particular, local persistent homology and element-specific persistent homology are considered for topological feature generation. These topological-feature-based random forest models can deliver a PCC up to 0.5822, which is 15.8% increase as compared to the performance of the previous model. Our WPHML models are suitable for any biomolecular-structure-based data analysis. Note that more sophisticated feature engineering of sequence-based information can further improve the accuracy to 0.61 [29]. This again demonstrates the great importance of featurization for material, chemical and biological learning models.

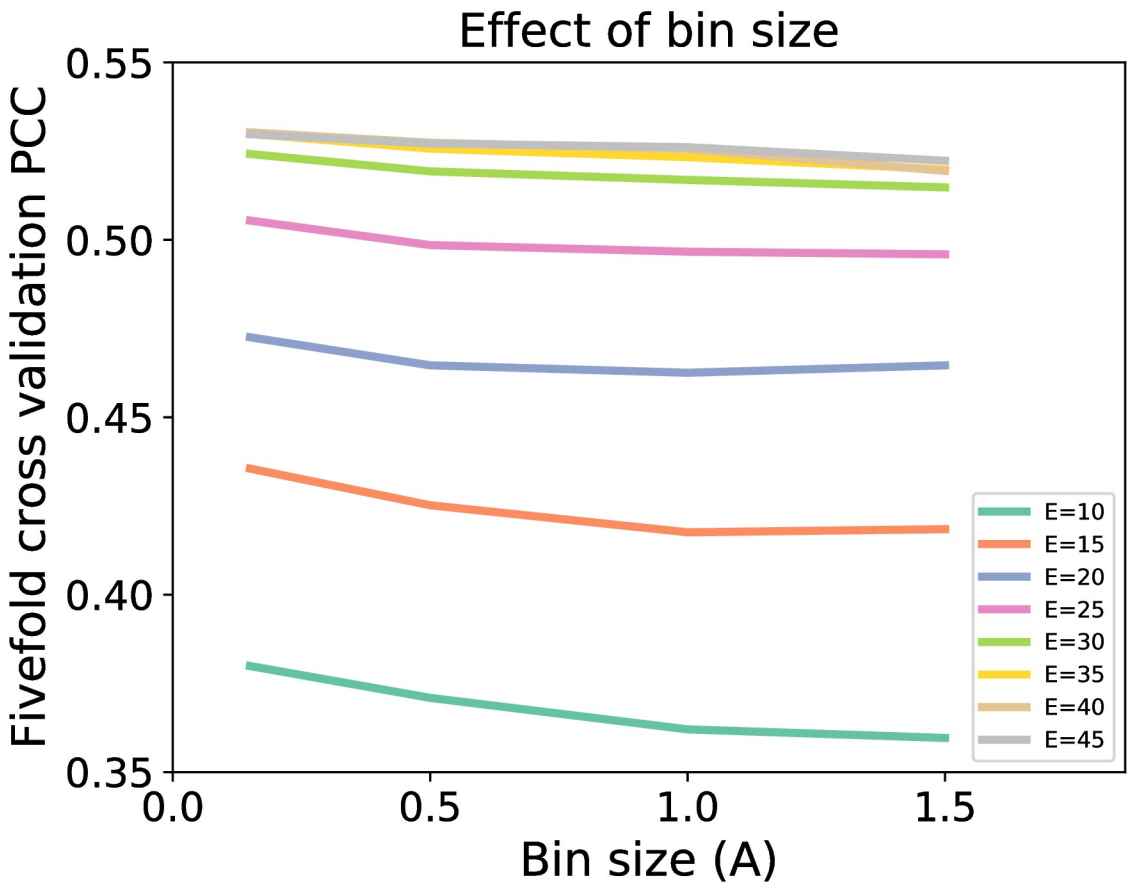

Fig 4. Effect of bin size on RF performance with a fixed F/E ratio of 1.0.

https://doi.org/10.1371/journal.pone.0237747.g004

Supporting information

S1 Table. PCC of each RNA chain in training set achieved by the best optimal RF.
(PDF)

S2 Table. PCC of each RNA chain in test set achieved by the best optimal RF.
(PDF)

Author Contributions

Conceptualization: Chi Seng Pun, Kelin Xia.

Data curation: Brandon Yung Sin Yong.

Formal analysis: Kelin Xia.

Funding acquisition: Chi Seng Pun, Kelin Xia.

Investigation: Brandon Yung Sin Yong.

Methodology: Kelin Xia.

Project administration: Kelin Xia.

Supervision: Chi Seng Pun, Kelin Xia.

Validation: Kelin Xia.

Visualization: Brandon Yung Sin Yong.

Writing – original draft: Brandon Yung Sin Yong, Kelin Xia.

Writing – review & editing: Brandon Yung Sin Yong, Kelin Xia.

References

1. McCammon J. A., Gelin B. R., and Karplus M. Dynamics of folded proteins. *Nature*, 267:585–590, 1977. https://doi.org/10.1038/267585a0 PMID: 301613
2. Go N., Noguti T., and Nishikawa T. Dynamics of a small globular protein in terms of low-frequency vibrational modes. *Proc. Natl. Acad. Sci.*, 80:3696–3700, 1983. https://doi.org/10.1073/pnas.80.12.3696 PMID: 6574507
3. Tasumi M., Takenchi H., Ataka S., Dwidedi A. M., and Krimm S. Normal vibrations of proteins: Glucagon. *Biopolymers*, 21:711–714, 1982. https://doi.org/10.1002/bip.360210318 PMID: 7066480
4. Brooks B. R., Bruccoleri R. E., Olafson B. D., States D.J., Swaminathan S., and Karplus M. Charmm: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, 4:187–217, 1983. https://doi.org/10.1002/jcc.540040211
5. Levitt M., Sander C., and Stern P. S. Protein normal-mode dynamics: Trypsin inhibitor, crambin, ribonuclease and lysozyme. *J. Mol. Biol.*, 181(3):423–447, 1985. https://doi.org/10.1016/0022-2836(85)90230-X PMID: 2580101
6. Jacobs D. J., Rader A. J., Kuhn L. A., and Thorpe M. F. Protein flexibility predictions using graph theory. *Proteins-Structure*, *Function*, *and Genetics*, 44(2):150–165, AUG 1 2001. https://doi.org/10.1002/prot.1081
7. Bahar I., Atilgan A. R., and Erman B. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding and Design*, 2:173–181, 1997. https://doi.org/10.1016/S1359-0278(97)00024-2 PMID: 9218955
8. Bahar I., Atilgan A. R., Demirel M. C., and Erman B. Vibrational dynamics of proteins: Significance of slow and fast modes in relation to function and stability. *Phys. Rev. Lett*, 80:2733–2736, 1998. https://doi.org/10.1103/PhysRevLett.80.2733
9. Atilgan A. R., Durrell S. R., Jernigan R. L., Demirel M. C., Keskin O., and Bahar I. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophys. J.*, 80:505–515, 2001. https://doi.org/10.1016/S0006-3495(01)76033-X PMID: 11159421
10. Hinsen K. Analysis of domain motions by approximate normal mode calculations. *Proteins*, 33:417–429, 1998. https://doi.org/10.1002/(SICI)1097-0134(19981115)33:3%3C417::AID-PROT10%3E3.0.CO;2-8 PMID: 9829700
11. Tama F. and Sanejouand Y. H. Conformational change of proteins arising from normal mode calculations. *Protein Eng.*, 14:1–6, 2001. https://doi.org/10.1093/protein/14.1.1 PMID: 11287673
12. Li G. H. and Cui Q. A coarse-grained normal mode approach for macromolecules: an efficient implementation and application to Ca(2+)-ATPase. *Bipohys. J.*, 83:2457–2474, 2002. https://doi.org/10.1016/S0006-3495(02)75257-0
13. Halle B. Flexibility and packing in proteins. *PNAS*, 99:1274–1279, 2002. https://doi.org/10.1073/pnas.032522499 PMID: 11818549
14. Zhang F. L. and Brüschweiler R. Contact model for the prediction of nmr nh order parameters in globular proteins. *Journal of the American Chemical Society*, 124(43):12654–12655, 2002. https://doi.org/10.1021/ja027847a PMID: 12392400
15. Lin C. P., Huang S. W., Lai Y. L., Yen S. C., Shih C. H., Lu C. H., et al. Deriving protein dynamical properties from weighted protein contact number. *Proteins: Structure*, *Function*, *and Bioinformatics*, 72(3):929–935, 2008. https://doi.org/10.1002/prot.21983
16. Xia K. L. and Wei G. W. Molecular nonlinear dynamics and protein thermal uncertainty quantification. *Chaos*, 24:013103, 2014. https://doi.org/10.1063/1.4861202 PMID: 24697365
17. Xia K. L. and Wei G. W. A stochastic model for protein flexibility analysis. *Physical Review E*, 88:062709, 2013. https://doi.org/10.1103/PhysRevE.88.062709
18. Xia K. L., Opron K., and Wei G. W. Multiscale multiphysics and multidomain models—Flexibility and Rigidity. *Journal of Chemical Physics*, 139:194109, 2013. https://doi.org/10.1063/1.4830404 PMID: 24320318

19. Opron K., Xia K. L., and Wei G. W. Fast and anisotropic flexibility-rigidity index for protein flexibility and fluctuation analysis. *Journal of Chemical Physics*, 140:234105, 2014. https://doi.org/10.1063/1.4882258 PMID: 24952521

20. de Brevern Alexandre G, Bornot Aurelie, Craveur Pierrick, Etchebest Catherine, and Gelly Jean-Christophe. PredyFlexy: flexibility and local structure prediction from sequence. *Nucleic acids research*, 40 (W1):W317–W322, 2012. https://doi.org/10.1093/nar/gks482 PMID: 22689641

21. Jing R, Wang Y, Wu Y, Hua Y, and Dai X. A research of predicting the b-factor based on the protein sequence. *J. Theor. Comput. Sci*, 1:1000111, 2014. https://doi.org/10.4172/2376-130X.1000111

22. Yuan Zheng, Bailey Timothy L., and Teasdale Rohan D. Prediction of protein b-factor profiles. *Proteins: Structure, Function, and Bioinformatics*, 58(4):905–912, jan 2005. https://doi.org/10.1002/prot.20375

23. Pan Xiao-Yong and Shen Hong-Bin. Robust prediction of b-factor profile from sequence using two-stage SVR based on random forest feature selection. *Protein & Peptide Letters*, 16(12):1447–1454, dec 2009. https://doi.org/10.2174/092986609789839250

24. Sonavane Shrihari, Jaybhaye Ashok A, and Jadhav Ajaykumar G. Prediction of temperature factors from protein sequence. *Bioinformation*, 9(3):134–140, jan 2013. https://doi.org/10.6026/97320630009134 PMID: 23422595

25. Radivojac P. Protein flexibility and intrinsic disorder. *Protein Science*, 13(1):71–80, jan 2004. https://doi.org/10.1110/ps.03128904 PMID: 14691223

26. Vihinen Mauno, Torkkila Esa, and Riikonen Pentti. Accuracy of protein flexibility predictions. *Proteins: Structure, Function, and Genetics*, 19(2):141–149, jun 1994. https://doi.org/10.1002/prot.340190207

27. Guruge Ivantha, Taherzadeh Ghazaleh, Zhan Jian, Zhou Yaoqi, and Yang Yuedong. B-factor profile prediction for RNA flexibility using support vector machines. *Journal of Computational Chemistry*, 39 (8):407–411, nov 2017. https://doi.org/10.1002/jcc.25124 PMID: 29164646

28. Bramer David and Wei Guo-Wei. Blind prediction of protein b-factor and flexibility. *The Journal of chemical physics*, 149(13):134107, 2018. https://doi.org/10.1063/1.5048469 PMID: 30292224

29. Wei Hong, Wang Boling, Yang Jianyi, and Gao Jianzhao. RNA flexibility prediction with sequence profile and predicted solvent accessibility. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2019. https://doi.org/10.1109/TCBB.2019.2956496

30. Altschul S. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, sep 1997. https://doi.org/10.1093/nar/25.17.3389 PMID: 9254694

31. Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, nov 2002. https://doi.org/10.1007/s00454-002-2885-2

32. Zomorodian Afra and Carlsson Gunnar. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, nov 2004. https://doi.org/10.1007/s00454-004-1146-y

33. Ghrist Robert. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45(01):61–76, oct 2007. https://doi.org/10.1090/S0273-0979-07-01191-3

34. Meng Z. Y., Anand D. V., Lu Y. P., Wu J., and Xia K. L. Weighted persistent homology for biomolecular data analysis. *Scientific Report*, 10, 2079, 2020.

35. Wu C. Y., Ren S. Q., Wu J., and Xia K. L. Weighted (co) homology and weighted laplacian. *Science China Mathematics*, https://doi.org/10.1007/s40840-020-00904-z, 2020.

36. Cang Z. X. and Wei G. W. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLOS Computational Biology*, 13(7):e1005690, 2017. https://doi.org/10.1371/journal.pcbi.1005690 PMID: 28749969

37. Cang Z. X. and Wei G. W. Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. *International journal for numerical methods in biomedical engineering*, page, 2017. PMID: 28677268

38. Cang Z. X., Mu L., and Wei G. W. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLoS computational biology*, 14(1):e1005929, 2018. https://doi.org/10.1371/journal.pcbi.1005929 PMID: 29309403

39. Hey Tony. The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, 2009.

40. Lo Y. C., Rensi S. E., Torng W., and Altman R. B. Machine learning in chemoinformatics and drug discovery. *Drug discovery today*, 23(8):1538–1546, 2018. https://doi.org/10.1016/j.drudis.2018.05.010 PMID: 29750902

41. Bajorath J. and Bajorath J. *Chemoinformatics and computational chemical biology*. Springer, 2011.

42. Libbrecht Maxwell W and Noble William Stafford. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321, 2015. https://doi.org/10.1038/nrg3920 PMID: 25948244

43. Anand D. V., Meng Z. Y., Xia K. L., and Mu Y. G. Weighted persistent homology for osmolyte molecular aggregation and hydrogen-bonding network analysis. *Scientific Report*, 10, 9685, 2020.

44. H. Edelsbrunner. *Weighted alpha shapes*, volume 92. University of Illinois at Urbana-Champaign, Department of Computer Science, 1992.

45. G. Bell, A. Lawson, J. Martin, J. Rudzinski, and C. Smyth. Weighted persistent homology. *arXiv preprint arXiv:1709.00097*, 2017.

46. Guibas L., Morozov D., and Mérigot Q. Witnessed k-distance. *Discrete & Computational Geometry*, 49 (1):22–45, 2013. https://doi.org/10.1007/s00454-012-9465-x

47. Buchet M., Chazal F., Oudot Steve Y., and Sheehy D. R. Efficient and robust persistent homology for measures. *Computational Geometry*, 58:70–96, 2016. https://doi.org/10.1016/j.comgeo.2016.07.001

48. Petri G., Scolamiero M., Donato I., and Vaccarino F. Topological strata of weighted complex networks. *PloS one*, 8(6):e66506, 2013. https://doi.org/10.1371/journal.pone.0066506 PMID: 23805226

49. Binchi J., Merelli E., Rucco M., Petri G., and Vaccarino F. jholes: A tool for understanding biological complex networks via clique weight rank persistent homology. *Electronic Notes in Theoretical Computer Science*, 306:5–18, 2014. https://doi.org/10.1016/j.entcs.2014.06.011

50. Dawson R. J. M. Homology of weighted simplicial complexes. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 31(3):229–243, 1990.

51. Ren S. Q., Wu C. Y., and Wu J. Weighted persistent homology. *Rocky Mountain Journal of Mathematics*, 48(8):2661–2687, 2018. https://doi.org/10.1216/RMJ-2018-48-8-2661

52. Cang Zixuan and Wei Guo-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, aug 2017. https://doi.org/10.1002/cnm.2914

53. Pun Chi Seng, Xia Kelin, and Lee Si Xian. Persistent-homology-based machine learning and its applications—a survey. *SSRN Electronic Journal*, 2018. https://doi.org/10.2139/ssrn.3275996

54. Cang Zixuan, Mu Lin, and Wei Guo-Wei. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLOS Computational Biology*, 14(1):e1005929, jan 2018. https://doi.org/10.1371/journal.pcbi.1005929 PMID: 29309403

55. Cang Zixuan and Wei Guo-Wei. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLOS Computational Biology*, 13(7):e1005690, jul 2017. https://doi.org/10.1371/journal.pcbi.1005690 PMID: 28749969

56. Hoerl A. E. Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58 (3):54–59, 1962.

57. Tibshirani Robert. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, jan 1996.

58. Fan Jianqing and Li Runze. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, dec 2001. https://doi.org/10.1198/016214501753382273

59. Zou Hui and Hastie Trevor. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, apr 2005. https://doi.org/10.1111/j.1467-9868.2005.00503.x

60. Zou Hui. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, dec 2006. https://doi.org/10.1198/016214506000000735

61. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, et al. Scikit-learn: Machine learning in python.

62. Breiman Leo, Friedman Jerome H., Olshen Richard A., and Stone Charles J. *Classification And Regression Trees*. Routledge, oct 1984.

63. Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*. IEEE Comput. Soc. Press, 1995.

64. Breiman Leo. Random forests. *Machine Learning*, 45(1):5–32, 2001. https://doi.org/10.1023/A:1010933404324

65. Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD 16*. ACM Press, 2016.

66. James Bennett and Stan Lanning. The netflix prize. *Proceedings of KDD Cup and Workshop 2007*, pages 3–6, August 2007.

67. Harris Drucker, Chris J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS'96, pages 155–161, Cambridge, MA, USA, 1996. MIT Press.

68. Cortes Corinna and Vapnik Vladimir. Support-vector networks. *Machine Learning*, 20(3):273–297, sep 1995. https://doi.org/10.1023/A:1022627411411

69. Smola Alex J. and Schölkopf Bernhard. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, aug 2004. https://doi.org/10.1023/B:STCO.0000035301.49549.88

70. Walczak Steven and Cerpa Narciso. Heuristic principles for the design of artificial neural networks. *Information and Software Technology*, 41(2):107–117, jan 1999. https://doi.org/10.1016/S0950-5849(98)00116-5

71. Hertz John, Krogh Anders, Palmer Richard G., and Horner Heinz. Introduction to the theory of neural computation. *Physics Today*, 44(12):70–70, dec 1991. https://doi.org/10.1063/1.2810360

72. Medsker Larry R. *Hybrid Neural Network and Expert Systems*. Springer US, 1994.

73. Barnard Etienne and Wessels L. F. A. Extrapolation and interpolation in neural network classifiers. *IEEE Control Systems*, 12(5):50–53, oct 1992. https://doi.org/10.1109/37.158898

74. Cherkassky V. and Lari-Najafi H. Data representation for diagnostic neural networks. *IEEE Expert*, 7(5):43–53, oct 1992. https://doi.org/10.1109/64.163672

75. Srivastava Nitish, Hinton Geoffrey, Krizhevsky Alex, Sutskever Ilya, and Salakhutdinov Ruslan. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

76. François Chollet et al. Keras. https://keras.io, 2015.

77. Tronrud D. E. Knowledge-based b-factor restraints for the refinement of proteins. *Journal of Applied Crystallography*, 29(2):100–104, apr 1996. https://doi.org/10.1107/S002188989501421X

78. Smith David K., Radivojac Predrag, Obradovic Zoran, Keith Dunker A., and Zhu Guang. Improved amino acid flexibility parameters. *Protein Science*, 12(5):1060–1072, may 2003. https://doi.org/10.1110/ps.0236203 PMID: 12717028

79. The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2015.

80. Dionysus: the persistent homology software. Software available at http://www.mrzv.org/software/dionysus.