THE JOURNAL OF PHYSICAL CHEMISTRY B
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

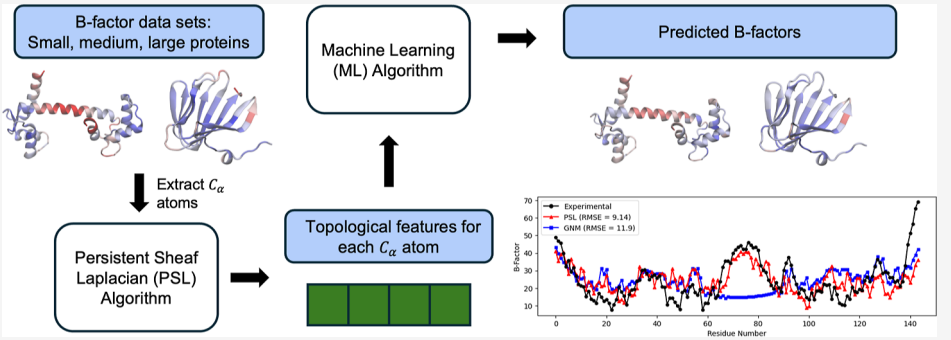
pubs.acs.org/JPCB
Article

Persistent Sheaf Laplacian Analysis of Protein Flexibility

Published as part of The Journal of Physical Chemistry B special issue "At the Cutting Edge of Theoretical and Computational Biophysics".

Nicole Hayes, Xiaoqi Wei, Hongsong Feng, Ekaterina Merkurjev,* and Guo-Wei Wei*

Cite This: J. Phys. Chem. B 2025, 129, 4169−4178
Read Online



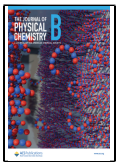
ABSTRACT: Protein flexibility, measured by the *B*-factor or Debye−Waller factor, is essential for protein functions such as structural support, enzyme activity, cellular communication, and molecular transport. Theoretical analysis and prediction of protein flexibility are crucial for protein design, engineering, and drug discovery. In this work, we introduce the persistent sheaf Laplacian (PSL), an effective tool in topological data analysis, to model and analyze protein flexibility. By representing the local topology and geometry of protein atoms through the multiscale harmonic and nonharmonic spectra of PSLs, the proposed model effectively captures protein flexibility and provides accurate, robust predictions of protein *B*-factors. Our PSL model demonstrates an increase in accuracy of 32% compared to the classical Gaussian network model (GNM) in predicting *B*-factors for a data set of 364 proteins. Additionally, we construct a blind machine learning prediction method utilizing global and local protein features. Extensive computations and comparisons validate the effectiveness of the proposed PSL model for *B*-factor predictions.

1. INTRODUCTION

Proteins are pivotal to life, playing an essential role in many biological processes, including signaling, gene regulation, transcription, translation, interaction with a protein or substrate molecule, etc.[1] They are composed of amino acids, which form polypeptide chains and fold into specific three-dimensional (3D) structures. There are four levels of protein structures: primary, secondary, tertiary, and quaternary. The primary structure is the linear sequence of amino acids, whereas the secondary structure refers to α-helices and β-sheets due to hydrogen bonds and electrostatic interactions. The tertiary structure corresponds to the 3D shape of a single polypeptide chain, while the quaternary structure describes the global arrangement of multiple polypeptide chains into a functional complex.[2]

Proteins have various functions; most notably, some of the functions of proteins include catalyzing metabolic reactions (enzymes), providing structural support (e.g., collagen in connective tissues), facilitating cellular communication (e.g., receptors and signaling molecules), and transporting molecules (e.g., hemoglobin for oxygen transport). These functions originate from their 3D structures. In particular, protein structure flexibility is a vital characteristic of protein structure that is essential to protein functions.[3] Specifically, protein flexibility enables proteins to adapt to various shapes and conditions, which facilitate their interactions with other molecules, such as DNA, RNA, ions, cofactors, ligands, and other small molecules. Under physiological conditions, proteins undergo constant thermal fluctuation, which enables the proteins to bind substrates, catalyze reactions, and transmit signals. Enzymes, for example, exhibit an induced fit mechanism, where their active sites adapt complementary shapes to accommodate substrates, improving the catalytic

Received: February 25, 2025
Revised: April 11, 2025
Accepted: April 14, 2025
Published: April 22, 2025



https://doi.org/10.1021/acs.jpcb.5c01287
J. Phys. Chem. B 2025, 129, 4169−4178

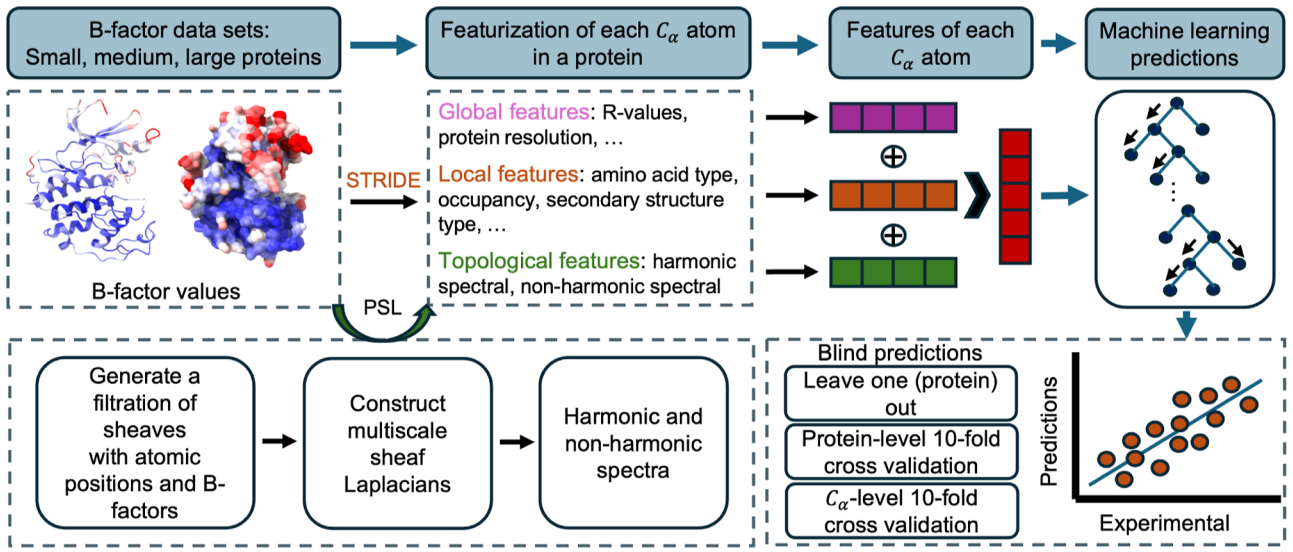


Figure 1. Outline of the methods used in our work. The blind *B*-factor prediction in Section 2.3 utilizes all pictured features, while the protein subset results from Section 2.1 include only the topological features generated using the persistent sheaf Laplacian (PSL) model.

efficiency. In a similar way, molecular motors, such as myosins and kinesins, utilize flexibility to enable directed movement during muscle contraction and intracellular transport.

Protein flexibility can be measured by the *B*-factor, also known as the Debye−Waller factor, which measures the attenuation of X-ray or neutron scattering due to thermal motion of atoms in protein crystallography. Specifically, the *B*-factor is defined according to the mean displacement of a scattering center in X-ray diffraction data.[4,5] The *B*-factor is used to describe the flexibility of atoms and/or amino acids within a protein structure, and it further provides valuable information about the protein's thermal motion, structural stability, activity, and other protein functions.[6]

Protein flexibility has been intensively studied in computational biophysics in recent decades.[7−10] In addition to the thoroughly investigated flexibility of proteins involved in folding, folded proteins (i.e., proteins in their native conformations) are also flexible and, in fact, exhibit internal motion in neighborhoods of their native conformations.[11,12] In a seminal work, McCammon et al.[11] investigated such local motion in a small folded globular protein using a molecular dynamics (MD) approach, demonstrating the fluid-like characteristics of the internal motions. However, analyzing the dynamics of a large protein would require simulations at time scales that are intractable for the MD approach.[13] Consequently, other methods have since emerged using a time-harmonic approximation[14] to the protein's potential energy function used in MD, resulting in time-independent techniques. Such methods include normal-mode analysis (NMA)[14−18] and elastic network models (ENMs).[19−24]

Some of the most popular methods[13,25,26] for protein flexibility analysis include the Gaussian network model (GNM)[21,27,28] and anisotropic network model (ANM),[19] both of which are types of ENMs. The GNM approach treats the protein as a network, with the residues representing the junctions. *B*-Factors are then approximated using the first few eigenvalues of the connectivity matrix, which correspond to the long-time dynamics of proteins that MD simulations are unable to capture.[29] Moreover, multiple methods have emerged as modifications of the original GNM and ANM models, including generalized GNM (gGNM), multiscale GNM (mGNM), and multiscale ANM (mANM).[26] Such methods attempt to improve the efficiency and accuracy of GNM and ANM. Due to their ability to capture multiscale information intrinsic to protein structures, mGNM and mANM models have been shown[26] to significantly improve *B*-factor predictions of proteins compared to the original GNM and ANM methods.

Other algorithms, such as the flexibility-rigidity index (FRI),[13] which relies on the theory of continuum elasticity with atomic rigidity (CEWAR), have also improved results for *B*-factor prediction over the original GNM method. The FRI is based on the assumption that protein functions depend solely upon the protein's structure and environment, and therefore it assesses flexibility and rigidity by analyzing the topological connectivity and geometric compactness of protein structures. A benefit of the flexibility-rigidity index is that it bypasses the Hamiltonian interaction matrix and matrix diagonalization. Consequently, the FRI has significantly reduced computational complexity compared to other algorithms for protein flexibility analysis. Additional modifications, including fast FRI (fFRI),[25] anisotropic FRI (aFRI),[25] and multiscale FRI (mFRI),[30] have been developed to further improve the efficiency of FRI as well as its accuracy on structures that are difficult for the NMA, GNM, and FRI algorithms.[30]

Recently, many machine learning approaches have been developed for protein flexibility analysis. For example, sequence-based predictions have been reported,[31−33] and other machine-learning-based predictions of protein flexibility have also been proposed.[33−35] More recently, a method that utilizes both sequence information and structure information has been developed for protein *B*-factor prediction.[36]

In 2019, persistent topological Laplacians (PTLs)[37,38] were first introduced to overcome certain drawbacks of persistent homology, a key technique used in topological data analysis (TDA).[39,40] Many PTLs have been proposed in the past few years, including the persistent combinatorial Laplacian, the persistent path Laplacian, the persistent sheaf Laplacian (PSL),[41] the persistent directed graph Laplacian, and the persistent hyperdigraph Laplacian.[42] Most of these algorithms are global, offering the topological and geometric descriptions of all objects in their topological space. In other words, they

https://doi.org/10.1021/acs.jpcb.5c01287
J. Phys. Chem. B 2025, 129, 4169−4178

Table 1. Average Pearson Correlation Coefficients for the PSL Model Compared to Other Methods[a]

protein set	PSL	ASPH (B)[5]	ASPH (W)[5]	opFRI[25]	pfFRI[25]	GNM[14]	NMA[14]
small	0.927	0.85	0.86	0.667	0.594	0.541	0.480
medium	0.728	0.69	0.69	0.664	0.605	0.550	0.482
large	0.643	0.61	0.62	0.636	0.591	0.529	0.494
superset	0.751	0.65	0.66	0.673	0.626	0.565	NA

[a]Experiments were conducted on the full set of 364 proteins as well as three subsets of small, medium, and large protein structures as described by Park et al.[14] ASPH denotes the atom-specific persistent homology method developed by Bramer et al.,[5] with results using Bottleneck (B) and Wasserstein (W) metrics displayed. Both sets of ASPH results used both an exponential and Lorenz kernel for least-squares fitting. opFRI and pfFRI results are from Opron et al.,[25] and GNM and NMA results are from Park et al.[14]

generate information about the protein as a whole. However, for protein flexibility analysis, one must have a method to describe the local properties of individual atoms. The PSL model serves such a function, as it allows the assignment of a specific weight at each node (or atom); thus, it provides local topological and geometric information in its spectra, making it suitable for protein flexibility analysis.

The aim of the present work is to demonstrate the utility of the PSL model for protein flexibility analysis via the prediction of protein *B*-factors. The remainder of this manuscript is organized in the following manner: all results of this work are given in Section 2. Section 2.1 summarizes our results on protein subsets from the literature, and Section 2.2 presents the performance of the PSL model on individual proteins that are challenging for the GNM. Section 2.3 details the results for blind machine learning prediction using the PSL model. In Section 3, we describe the algorithms used in this manuscript, including some background on persistent homology and cellular sheaves.

2. RESULTS

In this section, we present our results for experiments applying the persistent sheaf Laplacian (PSL) model as outlined in the previous section. Figure 1 summarizes the methods used to generate the results throughout this section.

2.1. Results on Protein Subsets. *2.1.1. Data Sets.* To demonstrate the persistent sheaf Laplacian model's performance on proteins of various sizes, we conducted computational experiments on four data sets. Three of these data sets were constructed by Park et al.[14] as sets of relatively small-, medium-, and large-sized protein structures. There are 33 proteins in the set of small-sized proteins, 36 in the set of medium-sized proteins, and 35 in the set of large-sized proteins. The fourth data set is a superset constructed by Opron et al.[25,30] consisting of (1) the three aforementioned sets, (2) 40 proteins of varying sizes randomly selected from the Protein Data Bank (PDB),[43] and (3) 263 high-resolution protein structures used by Xia et al.[13] in tests of their FRI algorithm, with the duplicates subsequently removed (note that in their earlier paper, Opron et al.[25] used a set of 365 proteins, but their later manuscript[30] excluded the protein with PDB ID 1AGN due to an unrealistic *B*-factor. The present paper utilizes the updated set consisting of 364 proteins).

Additionally, all protein data sets used for *B*-factor prediction in the present study were preprocessed to contain only the C_α atoms from their respective proteins. As discussed by Xia et al.,[13] the *B*-factor for an arbitrary atom in a protein is associated with that atom's flexibility, but its *B*-factor may be affected by diffraction in data collection, preventing a direct interpretation of flexibility. However, the *B*-factors of C_α atoms correlate directly with their atomic flexibility. Accordingly, our *B*-factor predictions in this work can be interpreted as atomic flexibility predictions.

Table 1 displays the results of the PSL model compared to other methods on the data sets of small, medium, and large proteins as well as the superset.

2.1.2. Parameters and Results. For all PSL results in this section and Section 2.2, we utilized a filtration induced by three radii: 6, 9, and 12 Å. For each radius, we generate a zeroth persistent sheaf Laplacian matrix L_0 and compute its eigenvalues, then compute the maximum, minimum, mean, and median of the set of nonzero eigenvalues, as well as the number of zero eigenvalues. These quantities comprise five features for each radius, resulting in 15 features in total for each residue. To obtain the *B*-factor predictions in this section, we performed linear regression using the set of PSL features for the full set of 364 proteins as well as the subsets.

To better assess the performance of the PSL method relative to other approaches and to avoid overfitting, we did not perform an extensive search for the optimal filtration radii and eigenvalue statistic parameters for each task below. Rather, we conducted experiments on the set of 364 proteins with a few sets of parameters and chose those that yielded a good average Pearson correlation coefficient over the entire set. The above parameters may be tuned to further improve model performance for a given task—higher-order persistent sheaf Laplacian matrices and their respective eigenvalues may also be used to generate such features, and other statistics may be used as well, such as the standard deviation of the nonzero eigenvalues. Moreover, suitable filtration radii may be chosen to capture desired multiscale information for a given protein. Another example of PSL feature generation can be seen in Section 2.3.2.

The PSL model achieves improved performance over all other compared methods on all data sets shown in Table 1. In particular, the PSL model improves the benchmark GNM by 32%.

2.2. Individual Protein Case Studies. As Opron et al. discussed in their 2015 work,[30] the Gaussian network model (GNM) experiences difficulty in predicting *B*-factors for certain protein structures. In addition to the comparison shown in Table 1, in this section, we examine a few case studies of particular proteins to demonstrate the success of the PSL model on such structures. All protein structural visualizations were generated using the visual molecular dynamics software (VMD),[44] and residues of each protein are assigned colors based on their experimental or predicted *B*-factors. Lower *B*-factors are shown as blue (corresponding to "colder" or more rigid residues), and higher *B*-factors are shown as red (corresponding to "warmer" or more flexible residues). All GNM results were obtained using the default GNM model with a cutoff of 7 Å.

https://doi.org/10.1021/acs.jpcb.5c01287
J. Phys. Chem. B 2025, 129, 4169−4178

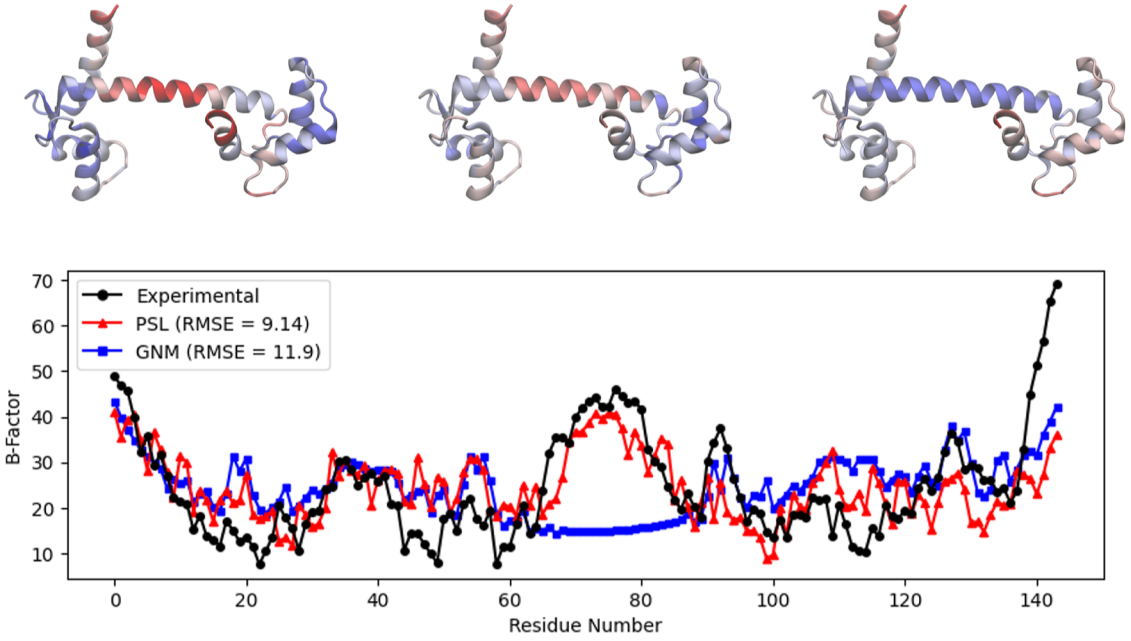


Figure 2. Top: visualization of the protein calmodulin (PDB ID: 1CLL) using visual molecular dynamics (VMD),[44] with residues colored by experimental *B*-factors (left), *B*-factors predicted by PSL (center), and *B*-factors predicted by GNM (right). Bottom: experimental and predicted *B*-factors for each residue of the protein. The GNM result uses the default cutoff of 7 Å. The GNM underestimates the *B*-factors for residues between about 65 and 85.

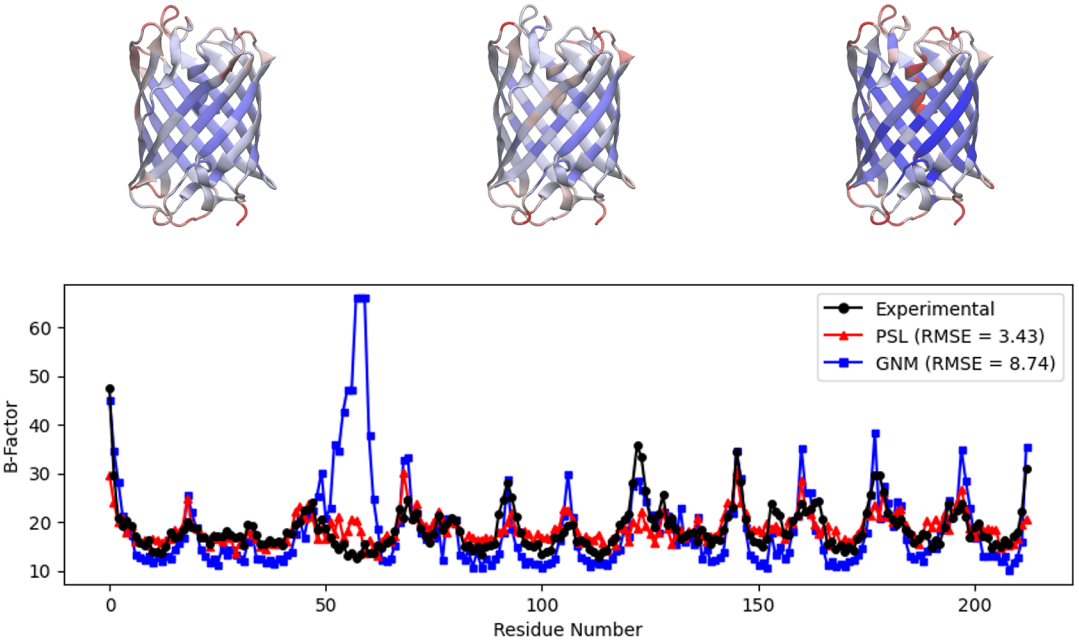


Figure 3. Top: visualization of the protein mTFP1 (PDB ID: 2HQK) using VMD,[44] with residues colored by experimental *B*-factors (left), *B*-factors predicted by PSL (center), and *B*-factors predicted by GNM (right). Bottom: experimental and predicted *B*-factors for each residue of the protein. The GNM result uses the default cutoff of 7 Å. The GNM vastly overestimates the *B*-factors of residues around 50−60.

Calmodulin is a calcium detector within the cells and plays a significant role in numerous cellular pathways. Its flexibility allows it to interact with varied target proteins. Figure 2 displays the predicted and experimental *B*-factors for the calcium-binding protein calmodulin (PDB ID: 1CLL)[43] using our persistent sheaf Laplacian model as well as the Gaussian network model. We observe that the Gaussian network model produces a large error in *B*-factor prediction for residues from about 65−85. These residues correspond to a flexible hinge region of the protein.[30] The root-mean-square error (RMSE) for the PSL model is 9.14 for calmodulin, a 23% decrease from the GNM model's RMSE of 11.9.

Next, we consider a monomeric cyan fluorescent protein (mTFP) that emits cyan light. It is used in biological experiments to visualize specific targets. Figure 3 shows experimental *B*-factors and predicted *B*-factors of the protein mTFP1 (PDB ID: 2HQK). Again, the predicted *B*-factors shown were computed using the Gaussian network model and our PSL model. As in the results for the protein calmodulin, the GNM is unable to correctly predict *B*-factors for one range

https://doi.org/10.1021/acs.jpcb.5c01287
J. Phys. Chem. B 2025, 129, 4169−4178

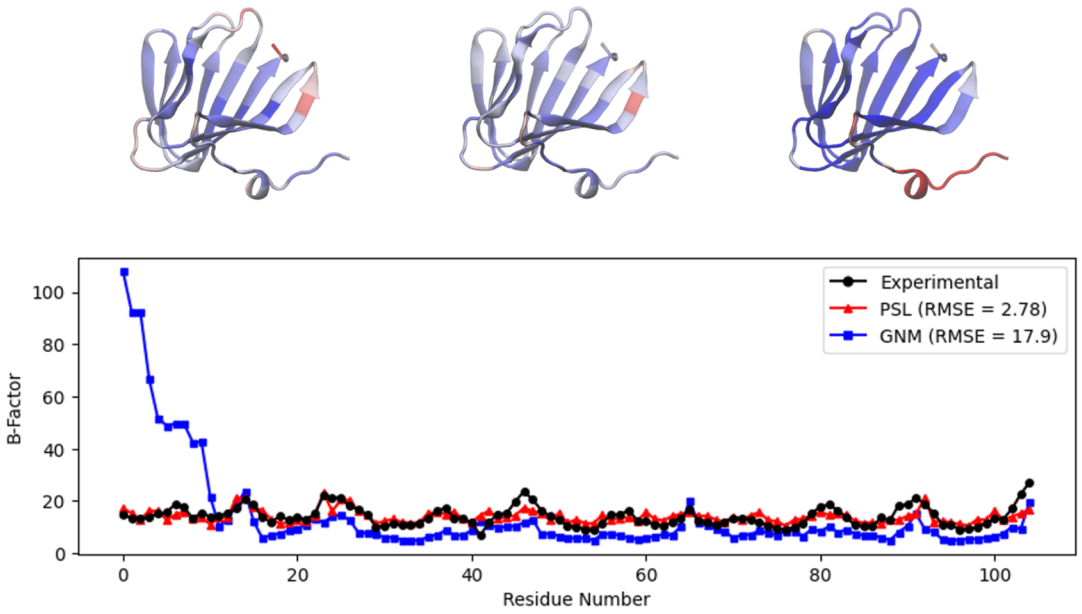


Figure 4. Top: visualization of the protein with PDB ID 1V70 using VMD,[44] with residues colored by experimental *B*-factors (left), *B*-factors predicted by PSL (center), and *B*-factors predicted by GNM (right). Bottom: experimental and predicted *B*-factors for each residue of the protein. The GNM result uses a cutoff of 7 Å. The GNM vastly overestimates the *B*-factors for residues from about 0−10.

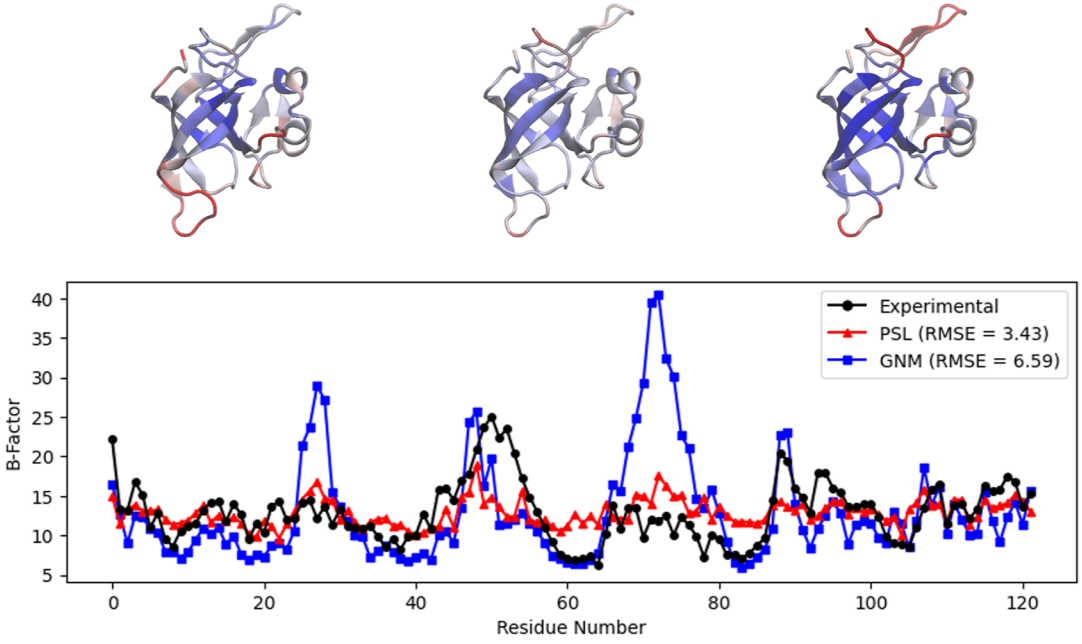


Figure 5. Top: visualization of the ribosomal protein L14 (PDB ID: 1WHI) using VMD,[44] with residues colored by experimental *B*-factors (left), *B*-factors predicted by PSL (center), and *B*-factors predicted by GNM (right). Bottom: experimental and predicted *B*-factors for each residue of the protein. The GNM result uses a cutoff of 7 Å. The GNM overestimates the *B*-factors for residues between 60 and 80.

of residues (around residues 50−60) in the protein mTFP1. Here, however, the Gaussian network model overestimates the *B*-factors in this region, visible in the GNM structural representation as the red α-helix in the center of the β-barrel.[30] Opron et al.[30] observed that using a cutoff of 8 Å for GNM somewhat resolves this error, and they suggested that the GNM may experience difficulty in this region due to its use of hard thresholds based on connectivity parameters. The persistent sheaf Laplacian model is significantly more accurate in this region, likely due to the fact that it captures atom-specific information as well as molecular information at multiple scales. Overall, the PSL model improves the RMSE on mTFP1 to 3.43 from 8.74 for the GNM, a nearly 61% decrease.

We further consider a probable antibiotics synthesis protein from *Thermus thermophilus*. In Figure 4, we investigate the experimental and predicted *B*-factors of this protein (PDB ID: 1V70). On this protein, our persistent sheaf Laplacian model is able to predict the *B*-factors accurately across all residues of the protein, while the Gaussian network model experiences a high level of inaccuracy on residues from about 0−10. This vast over-prediction contributes to a very high RMSE value for the GNM, at 17.9. Our PSL model achieves a significantly lower

https://doi.org/10.1021/acs.jpcb.5c01287
J. Phys. Chem. B 2025, 129, 4169−4178

RMSE of 2.78 on the protein 1V70, 84% lower than that of the GNM.

Finally, we studied the ribosomal protein L14 (PDB ID: 1WHI),[30] one of the most conserved ribosomal proteins. It functions as an organizational component of the translational apparatus. In Figure 5, we show the experimental and predicted *B*-factors for the ribosomal protein L14. Again, we observe that the GNM overestimates the flexibility of some regions of this protein, most significantly for the residues around 60−80. The RMSE for the PSL model on this protein is nearly half that of the GNM model, whose RMSE is 6.59.

2.3. Blind Machine Learning Prediction. *2.3.1. Data Sets.* Two data sets, one from Opron et al.[25,30] and the other from Park et al.[14] are used in our work. The first data set contains 364 proteins,[25,30] and the second[14] has three sets of proteins with small, medium, and large sizes, which are the subsets of the 364 protein set.

In our blind predictions, proteins 1OB4, 1OB7, 2OXL, and 3MD5 from the superset are excluded because the STRIDE software cannot generate features for these proteins. We exclude protein 1AGN due to the known problems with this protein data.[25,30] Additional proteins from the superset are also excluded. Proteins 1NKO, 2OCT, and 3FVA are excluded because these proteins have unphysical *B*-factors (i.e., zero values). We also excluded proteins 3DWV, 3MGN, 4DPZ, 2J32, 3MEA, 3A0M, 3IVV, 3W4Q, 3P6J, and 2DKO due to inconsistent protein data processed with STRIDE compared to original PDB data. A total of 346 proteins are used for blind predictions. Those data can be found in our provided GitHub repository.

2.3.2. PSL Features. The second approach to *B*-factor prediction that we examined is a blind prediction for protein *B*-factors. We use PSL features as local descriptors of protein structures, applying three cutoff distances, i.e., 7, 10, and 13 Å, to define the atom groups used to construct a sheaf Laplacian matrix. For each cutoff distance, we generate a sheaf Laplacian matrix, L_1, with a filtration radius matching the cutoff distance. From each matrix, we extract five features: the count of zero eigenvalues, and the maximum, minimum, mean, and standard deviation of the nonzero eigenvalues. Together, these provide 15 PSL features for blind machine learning predictions.

2.3.3. Additional Features. In addition to PSL features, we extract a range of global and local protein features for building machine learning models. Each PDB structure is associated with global features, such as the *R*-value, resolution, and the number of heavy atoms, which are extracted from the PDB files. These features enable the comparison of the *B*-factors in different proteins. The local characteristics of each protein consist of packing density, amino acid type, occupancy, and secondary structure information generated by STRIDE.[45] STRIDE provides comprehensive secondary structure details for a protein based on its atomic coordinates from a PDB file, classifying each atom into categories such as α-helix, 3−10-helix, π-helix, extended conformation, isolated bridge, turn, or coil. Furthermore, STRIDE provides ϕ and ψ angles and residue solvent-accessible area, contributing a total of 12 secondary features. In our implementation, we use one-hot encoding for both amino acid types and the 12 secondary features. The packing density of each C_α atom in a protein is calculated based on the density of surrounding atoms, with short, medium, and long-range packing density features defined for each C_α atom. The packing density of the ith C_α atom is defined as

$$p_i^d = \frac{N_d}{N} \tag{1}$$

where d represents the specified cutoff distance in Å, N_d denotes the number of atoms within the Euclidean distance d from the ith atom, and N is the total number of heavy atoms in the protein. The packing density cutoff values used in this study are provided in Table 2.

Table 2. Packing Density Parameter in Distance *d* Å

short	medium	long
$d < 3$	$3 \geq d < 5$	$5 \leq d$

Our PSL features, combined with the global and local features provided for each PDB file, offer a comprehensive feature set for each C_α atom in the protein. For blind predictions, we integrate these features with machine learning algorithms to build regression models. To evaluate the performance of our machine learning model on blind predictions, we conducted two validation tasks: 10-fold cross-validation and leave-one-(protein)-out validation. For 10-fold cross-validation, we designed two types of experiments—one based on splitting by PDB files and another on splitting by all C_α atoms collected from the PDB files. Our modeling and predictions are centered on the *B*-factors of C_α atoms.

2.3.4. Evaluation Metrics. To assess our method for *B*-factor prediction, we use the Pearson correlation coefficient (PCC)

$$\mathrm{PCC}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{m=1}^{M} (B_m^e - \overline{B}^e)(B_m^t - \overline{B}^t)}{\sqrt{\sum_{m=1}^{M} (B_m - \overline{B}^e)^2 \sum_{m=1}^{M} (B_m^t - \overline{B}^t)^2}}$$

where B_m^t, $m = 1, 2, ..., N$ are the predicted *B*-factors and B_m^e, $m = 1, 2, ..., N$ are the experimental *B*-factors from the PDB file. Here $\overline{B}^e$ and $\overline{B}^t$ are the averaged *B*-factors.

2.3.5. Machine Learning Algorithms. For the blind predictions, instead of using more sophisticated methods,[46−48] we consider two simple machine learning algorithms, namely gradient-boosting decision trees (GBDT) and random forests (RF), to highlight the proposed PSL method. The hyperparameters of these two types of algorithms are given in Table 3.

Table 3. Hyperparameters of the Random Forest (RF) and Gradient Boosting Decision Tree (GBDT) Algorithms Used for the *B*-Factor Predictions

RF parameters	GBDT parameters
	n_estimators = 1000
n_estimators = 1000	max_depth = 7
max_depth = 8	min_samples_split = 5
min_samples_split = 4	subsample = 0.8
min_samples_leaf = 0.8	learning_rate = 0.002
	max_features = "sqrt"

2.3.6. Machine Learning Results. We carried out several experiments, the first of which is a leave-one-(protein)-out prediction using the four data sets described above. We trained models five times independently with different random seeds and calculated the average Pearson correlation coefficients from the ten sets of modeling predictions. Our results are

https://doi.org/10.1021/acs.jpcb.5c01287
J. Phys. Chem. B 2025, 129, 4169−4178

shown in Table 4, where the GBDT-based models yield better predictions than the RF-based models, as expected.

Table 4. Average Pearson Correlation Coefficients (PCC) of Leave-One (Protein)-Out Predictions for the Four *B*-Factor Datasets[a]

protein set	RF	GBDT
small	0.478	0.433
medium	0.518	0.590
large	0.508	0.582
superset	0.542	0.588

[a]The PCC results obtained with random forest (RF) and gradient boosting decision tree (GBDT) models are compared.

In our study, we additionally carried out 10-fold cross-validation at the protein level. In each fold, we use nine out of the ten subsets of the 346 proteins to train our model, while the remaining subset is reserved for testing. Specifically, features of C_α atoms in the training proteins are pooled together to train the models, while those in the test proteins are used for evaluation. This process is repeated across ten different splits. Table 5 shows the average PCC values for two types of machine learning models. Again, the GBDT model gives better predictions than the RF model.

Table 5. Average Pearson Correlation Coefficient (PCC) From Protein-Level 10-Fold Cross Validation Predictions With the Collected 346 Proteins[a]

protein set	RF	GBDT
superset	0.397	0.452

[a]The *B*-factor values of C_α atoms in each protein are predicted. The average PCC value is calculated from five independent experiments. The PCC results with random forest (RF) and gradient boosting decision tree (GBDT) modeling are compared.

We also performed an alternative C_α-level 10-fold cross-validation. The data set consists of more than 74,000 C_α atoms from 364 proteins. In each of ten independent models, nine out of ten subsets of C_α atoms are used to train the models, while the remaining subset is used for testing. As shown in Table 6, GBDT modeling yields slightly better predictions than RF-based modeling.

Table 6. Average Pearson Correlation Coefficient (PCC) From C_α-Level 10-Fold Cross Validation Predictions With all C_α Atoms in the Collected 346 Proteins[a]

protein set	RF	GBDT
superset	0.839	0.840

[a]The average PCC value is calculated from five independent experiments. The PCC results with random forest (RF) and gradient boosting decision tree (GBDT) models are compared.

3. METHODS

3.1. Persistent Homology and Persistent Laplacians.

As one of the most abstract mathematical subjects, homology excessively simplifies complex geometry. In contrast, persistent homology balances simplification and information retrieval in data analysis and is widely used in topological data analysis.[39,40] However, persistent homology has several drawbacks, including its insensitivity to homotopic shape evolution. To address this challenge, the persistent spectral graph, also known as persistent Laplacians, was introduced on simplicial complexes in 2019.[37] Since then, various persistent Laplacians, or persistent topological Laplacians, have been proposed for different topological objects, such as path complexes, directed flag complexes, hyperdigraphs, and cellular sheaves.[42]

Given a finite set V, a simplicial complex X is a collection of subsets of V, such that if a set σ is in X, then any subset of σ is also in X. A set σ that consists of $q+1$ elements is referred to as a q-simplex. If σ is a subset of τ, then we say that σ is a face of τ and denote the face relation by $\sigma \leqslant \tau$. If X and Y are simplicial complexes and $X \subset Y$, then X is referred to as a sub-complex of Y. A simplicial complex X gives rise to a simplicial chain complex

$$\cdots \xrightarrow{\partial_3} C_2(X) \xrightarrow{\partial_2} C_1(X) \xrightarrow{\partial_1} C_0(X) \longrightarrow 0.$$

The real vector space $C_q(X)$ is generated by q-simplices. An element of $C_q(X)$ is called a q-chain. The boundary operator ∂_q is a linear map defined by

$$\partial_q[v_{a_0}, ..., v_{a_q}] = \sum_i (-1)^i [v_{a_0}, ..., \hat{v}_{a_i}, ..., v_{a_q}]$$

where the symbol $\hat{v}_{a_i}$ means that $\hat{v}_{a_i}$ is deleted. The total ordering of V ensures that the boundary operator is well-defined. The q-th homology group $H_q = \ker \partial_q / \mathrm{im}\partial_{q+1}$ is well-defined since $\partial^2 = 0$. Now suppose X is a sub-complex of Y. Then we have the following diagram

$$\begin{array}{ccccccccc} \cdots & \xrightarrow{\partial^X_{q+2}} & C_{q+1}(X) & \xrightarrow{\partial^X_{q+1}} & C_q(X) & \xrightarrow{\partial^X_q} & C_{q-1}(X) & \xrightarrow{\partial^X_{q-1}} & \cdots \\ & & \downarrow & & \downarrow & & \downarrow & & \\ \cdots & \xrightarrow{\partial^Y_{q+2}} & C_{q+1}(Y) & \xrightarrow{\partial^Y_{q+1}} & C_q(Y) & \xrightarrow{\partial^Y_q} & C_{q-1}(Y) & \xrightarrow{\partial^Y_{q-1}} & \cdots \end{array}$$

where hooked dashed arrows represent inclusion maps $\imath$: $C_q(X) \to C_q(Y)$. The inclusion $\imath$ induces a map $\imath^\bullet$: $H_q(X) \to H_q(Y)$. The q-th persistent homology for the pair (X, Y) is the image

$$\imath^\bullet(H_q(X))$$

Usually the ranks of persistent homology groups are represented by barcodes, where each bar represents a topological feature that persists in the filtration, offering a multiscale topological characterization of the input point cloud.[39,40]

Recently, the theory of persistent Laplacians[37] has been proposed to extract additional information from a point cloud. A persistent Laplacian is a positive semidefinite operator whose kernel is isomorphic to the corresponding persistent homology group. The additional information provided by the nonzero eigenvalues of persistent Laplacians can be learned by machine learning algorithms. Since $C_q(X)$ is generated by q-simplices, it is equipped with a canonical inner product. Let $C^{X,Y}_{q+1} = \{c \in C_{q+1}(Y) | \partial^Y_{q+1}(c) \in C_q(X)\}$ and $\partial^{X,Y}_{q+1}$ the restriction of ∂^Y_{q+1} to $C^{X,Y}_{q+1}$. The q-th persistent Laplacian $\Delta^{X,Y}_q$ is defined by

$$\partial^{X,Y}_{q+1}(\partial^{X,Y}_{q+1})^\dagger + (\partial^X_q)^\dagger \partial^X_q \tag{2}$$

where † denotes the adjoint of a linear morphism. Using basic linear algebra we can prove that the kernel of $\Delta^{X,Y}_q$ is isomorphic to $\imath^\bullet(H_q(X))$. Generally speaking, any method that utilizes multiscale Laplacians to analyze data can be referred to as a persistent Laplacian method.

https://doi.org/10.1021/acs.jpcb.5c01287
J. Phys. Chem. B 2025, 129, 4169−4178

3.2. Cellular Sheaves and Persistent Sheaf Laplacians. Molecular structures often contain important nonspatial information, and many applications of topological methods in analyzing molecular data require integration of nonspatial information. For example, we can use generalized distance to model the biochemical interaction between atoms or only use specific types of atoms as input to persistent homology[49] or persistent Laplacians.[37] An alternative approach is to integrate biological information through the construction of (co)chain complexes and extend persistent homology and persistent Laplacians to new settings. For example, one can construct a filtration of cellular sheaves and consider the persistence module of sheaf cochain complexes instead of simplicial complexes and simplicial chain complexes.[50]

Roughly speaking, a cellular sheaf $\mathcal{F}$ is a simplicial complex X with an assignment to each simplex σ of X a finite-dimensional vector space $\mathcal{S}(\sigma)$ (referred to as the stalk of $\mathcal{S}$ over σ) and to each face relation $\sigma \leqslant \tau$ (i.e., $\sigma \subset \tau$) a linear morphism of vector spaces denoted by $\mathcal{S}_{\sigma\leqslant\tau}$ (referred to as the restriction map of the face relation $\sigma \leqslant \tau$), satisfying the rule

$$\rho \leqslant \sigma \leqslant \tau \Rightarrow \mathcal{S}_{\rho\leqslant\tau} = \mathcal{S}_{\sigma\leqslant\tau}\mathcal{S}_{\rho\leqslant\sigma}$$

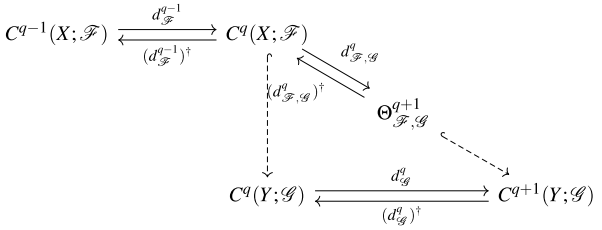
and $\mathcal{S}_{\sigma\leqslant\sigma}$ is the identity map of $\mathcal{S}(\sigma)$. We can view stalks as information stored for each simplex, and restriction maps as the way this information interacts. A cellular sheaf gives rise to a sheaf cochain complex

$$0 \longrightarrow C^0(X;\mathscr{S}) \xrightarrow{d} C^1(X;\mathscr{S}) \xrightarrow{d} C^2(X;\mathscr{S}) \xrightarrow{d} \cdots.$$

The q-th sheaf cochain group $C^q(X; \mathcal{S})$ is the direct sum of stalks over q-dimensional simplices. To define coboundary maps d, we can globally orient the simplicial complex X and obtain a signed incidence relation, an assignment to each $\sigma \leqslant \tau$ an integer $[\sigma:\tau]$. The co-boundary map d^q: $C^q(X; \mathcal{S}) \to C^{q+1}(X; \mathcal{S})$ is defined by

$$d^q|_{S(\sigma)} = \sum_{\sigma\leqslant\tau} [\sigma: \tau]\mathcal{S}_{\sigma\leqslant\tau}$$

Now suppose we have $\mathcal{F}$ on X and $\mathcal{G}$ on Y such that $X \subseteq Y$ and stalks and restriction maps of X are identical to those of Y. If each stalk is an inner product space then we have the following diagram



where $\Theta^{q+1}_{\mathcal{F},\mathcal{G}} = \{x \in C^{q+1}(Y; \mathcal{G}) | (d^q_{\mathcal{G}})^\dagger(x) \in C^q(X; \mathcal{F})\}$ and $d^q_{\mathcal{F},\mathcal{G}}$ is the adjoint of $\pi(d^q_{\mathcal{G}})^\dagger|_{\Theta^{q+1}_{\mathcal{F},\mathcal{G}}}$: $\Theta^{q+1}_{\mathcal{F},\mathcal{G}} \to C^q(X; \mathcal{F})$ (π is the projection map from $C^q(Y; \mathcal{G})$ to its subspace $C^q(X; \mathcal{F})$). We define the q-th persistent sheaf Laplacian $\Delta^{\mathcal{F},\mathcal{G}}_q$ by

$$\Delta^{\mathcal{F},\mathcal{G}}_q = (d^q_{\mathcal{F},\mathcal{G}})^\dagger d^q_{\mathcal{F},\mathcal{G}} + d^{q-1}_{\mathcal{F}}(d^{q-1}_{\mathcal{F}})^\dagger$$

When $\mathcal{F} = \mathcal{G}$, the persistent sheaf Laplacian is equal to the sheaf Laplacian of $\mathcal{F}$. When $\mathcal{F}$ and $\mathcal{G}$ are constant sheaves, persistent sheaf Laplacians coincide with persistent Laplacians. Since a sheaf co-chain complex is constructed through stalks and restriction maps, we expect that persistent sheaf cohomology and persistent sheaf Laplacians contain additional information besides the underlying simplicial complex.

If a simplicial complex X is labeled (each vertex is associated with a quantity q), then a sheaf can be constructed as follows. Let F: $X \to \mathbb{R}$ be a nowhere-zero function. We let each stalk be $\mathbb{R}$, and for the face relation $[v_0, ..., v_n] \leqslant [v_0, ..., v_n, v_{n+1}..., v_m]$ (here orientation is not relevant), the linear morphism $\mathcal{S}([v_0, ..., v_n] \leqslant [v_0, ..., v_n, v_{n+1}..., v_m])$ is the scalar multiplication by

$$\frac{F([v_0, ..., v_n])q_{n+1}\cdots q_m}{F([v_0, ..., v_n, v_{n+1}, ..., v_m])}$$

For a labeled point cloud (a point cloud where each point is associated with a quantity), if we construct a filtration of the point cloud, then for each complex in the filtration we can construct a sheaf as described above. This leads to a filtration of sheaves such as in persistent sheaf co-homology[51] and persistent sheaf Laplacians.[41] The harmonic spectra of PSLs reveal the topological invariants, while the nonharmonic spectra represent geometric information on the data.[41,42] In this work, we use sheaf Laplacians to construct features for individual C_α atoms. For a given atom A, we first pick a cutoff distance and only consider the nearby C_α atoms within the cutoff. Then we choose a radius and build an alpha complex X out of these C_α atoms. A cellular sheaf on X is constructed as follows. We denote an atom in X by v_i. We assign a label q_i to v_i, then, we let each stalk be $\mathbb{R}$. For face relation $v_i \leqslant v_iv_j$, the restriction map is the scalar multiplication by q_j/r_{ij}, where r_{ij} is the length of v_iv_j. For face relation $v_iv_j \leqslant v_iv_jv_k$, the restriction map is the scalar multiplication by $q_k/(r_{ik}r_{jk})$. Since we want to distinguish the C_α atom A from the other atoms, we let the label of A be 0, and the labels of other nearby C_α atoms be 1. The features are then obtained from the spectra of sheaf Laplacians for this specific C_α atom A. In this manner, we can construct sheaf Laplacian features for all C_α atoms.

4. CONCLUSION

Protein flexibility is crucial for protein functions, and its prediction is essential for understanding protein properties, protein design, and protein engineering. However, the intrinsic complexity of proteins and their interactions present challenges in understanding protein flexibility. To address this, many effective computational approaches have been developed to predict B-factor values, which reflect protein flexibility. In the literature, a variety of techniques have been proposed, including NMA,[16] GNM,[20,21] pfFRI,[25] ASPH,[5] opFRI,[25] and EH.[52]

In this study, we propose a persistent sheaf Laplacian (PSL) model for protein B-factor prediction. Sheaf theory, a branch of algebraic geometry, serves as the foundation for PSL, a novel approach to topological data analysis (TDA). Unlike many global TDA tools, PSL is a localized method that captures the local topology of a point within the data. Similarly to other TDA methods, PSL also provides a multiscale analysis of the system under study.

The multiscale nature of PSL allows it to capture atomic interactions across different distance ranges, enabling a more effective analysis of protein flexibility. This characteristic makes the proposed method superior to traditional approaches, such as GNM, which fail to account for atomic interactions beyond a specific cutoff distance.

https://doi.org/10.1021/acs.jpcb.5c01287
J. Phys. Chem. B 2025, 129, 4169−4178

For cross-protein prediction, we further enhance the PSL by integrating additional global and local features intrinsic to protein structures and structure determination conditions. This integration enables the blind prediction of protein *B*-factors, which is particularly valuable for assessing protein flexibility when experimental *B*-factors are unavailable. The proposed PSL model has been validated using various data sets, demonstrating its effectiveness and robustness in protein flexibility analysis.

■ ASSOCIATED CONTENT

Data Availability Statement

Code is available at https://github.com/weixiaoqimath/persistent_sheaf_Laplacians. Data is available at https://github.com/fenghon1/MDG_bfactor.

■ AUTHOR INFORMATION

Corresponding Authors

Ekaterina Merkurjev − *Department of Mathematics and Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, Michigan 48824, United States;* orcid.org/0000-0002-2489-8332; Email: merkurje@msu.edu

Guo-Wei Wei − *Department of Mathematics, Department of Electrical and Computer Engineering, and Department of Biochemistry and Molecular Biology, Michigan State University, East Lansing, Michigan 48824, United States;* Email: weig@msu.edu

Authors

Nicole Hayes − *Department of Mathematics, Michigan State University, East Lansing, Michigan 48824, United States*

Xiaoqi Wei − *Department of Mathematics, Michigan State University, East Lansing, Michigan 48824, United States;* Present Address: Department of Mathematics, North Carolina State University, Raleigh, NC 27695, United States

Hongsong Feng − *Department of Mathematics, Michigan State University, East Lansing, Michigan 48824, United States;* orcid.org/0000-0001-8039-3059

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jpcb.5c01287

Author Contributions

Conception and design: Guo-Wei Wei. Sample preparation and collection of data: Nicole Hayes. Algorithm implementation: Xiaoqi Wei, Hongsong Feng. Analysis and interpretation of data: Nicole Hayes, Guo-Wei Wei. Supervision: Ekaterina Merkurjev, Guo-Wei Wei. Manuscript preparation: Nicole Hayes, Xiaoqi Wei, Hongsong Feng, Ekaterina Merkurjev, Guo-Wei Wei. All authors contributed to the article and approved the submitted version.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported in part by NIH grants R01AI164266 and R35GM148196, NSF grant DMS-2052983, MSU Research Foundation, and Bristol-Myers Squibb 65109.

■ REFERENCES

(1) Petsko, G. A.; Ringe, D. *Protein Structure and Function*; New Science Press, 2004.

(2) Ivar Branden, C.; Tooze, J. *Introduction to Protein Structure*; Garland Science, 2012.

(3) Radivojac, P.; Obradovic, Z.; Smith, D. K.; Zhu, G.; Vucetic, S.; Brown, C. J.; Lawson, J. D.; Dunker, A. K. Protein flexibility and intrinsic disorder. *Protein Sci.* **2004**, *13* (1), 71−80.

(4) Sun, Z.; Liu, Q.; Qu, G.; Feng, Y.; Reetz, M. T. Utility of b-factors in protein science: Interpreting rigidity, flexibility, and internal motion and engineering thermostability. *Chem. Rev.* **2019**, *119* (3), 1626−1665.

(5) Bramer, D.; Wei, G.-W. Atom-specific persistent homology and its application to protein flexibility analysis. *Comput. Math. Biophys.* **2020**, *8* (1), 1−35.

(6) Yuan, Z.; Bailey, T. L.; Teasdale, R. D. Prediction of protein B-factor profiles. *Proteins: Struct., Funct., Bioinf.* **2005**, *58* (4), 905−912.

(7) Ma, J. Usefulness and limitations of normal mode analysis in modeling dynamics of biomolecular complexes. *Structure* **2005**, *13* (3), 373−380.

(8) Vihinen, M.; Torkkila, E.; Riikonen, P. Accuracy of protein flexibility predictions. *Proteins: Struct., Funct., Bioinf.* **1994**, *19* (2), 141−149.

(9) Jacobs, D. J.; Rader, A. j.; Kuhn, L. A.; Thorpe, M. f. Protein flexibility predictions using graph theory. *Proteins: Struct., Funct., Bioinf.* **2001**, *44* (2), 150−165.

(10) Camps, J.; Carrillo, O.; Emperador, A.; Orellana, L.; Hospital, A.; Rueda, M.; Cicin-Sain, D.; D'Abramo, M.; Gelpí, J. L.; Orozco, M. Flexserv: an integrated tool for the analysis of protein flexibility. *Bioinformatics* **2009**, *25* (13), 1709−1710.

(11) McCammon, J. A.; Gelin, B. R.; Karplus, M. Dynamics of folded proteins. *Nature* **1977**, *267* (5612), 585−590.

(12) Huber, R.; Bennett, W. S. Functional significance of flexibility in proteins. *Biopolymers* **1983**, *22* (1), 261−279.

(13) Xia, K.; Opron, K.; Wei, G. W. Multiscale multiphysics and multidomain models—flexibility and rigidity. *J. Chem. Phys.* **2013**, *139* (19), 194109.

(14) Park, J.-K.; Jernigan, R.; Wu, Z. Coarse grained normal mode analysis vs. refined Gaussian network model for protein residue-level structural fluctuations. *Bull. Math. Biol.* **2013**, *75*, 124−160.

(15) Tasumi, M.; Takeuchi, H.; Ataka, S.; Dwivedi, A. M.; Krimm, S. Normal vibrations of proteins: glucagon. *Biopolymers* **1982**, *21* (3), 711−714.

(16) Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.* **1983**, *4* (2), 187−217.

(17) Go, N.; Noguti, T.; Nishikawa, T. Dynamics of a small globular protein in terms of low-frequency vibrational modes. *Proc. Natl. Acad. Sci. U.S.A.* **1983**, *80* (12), 3696−3700.

(18) Levitt, M.; Sander, C.; Stern, P. S. Protein normal-mode dynamics: Trypsin inhibitor, crambin, ribonuclease and lysozyme. *J. Mol. Biol.* **1985**, *181* (3), 423−447.

(19) Atilgan, A. r.; Durell, S. r.; Jernigan, R. l.; Demirel, M. c.; Keskin, O.; Bahar, I. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophys. J.* **2001**, *80* (1), 505−515.

(20) Bahar, I.; Atilgan, A. R.; Demirel, M. C.; Erman, B. Vibrational dynamics of folded proteins: Significance of slow and fast motions in relation to function and stability. *Phys. Rev. Lett.* **1998**, *80*, 2733−2736.

(21) Bahar, I.; Atilgan, A. R.; Erman, B. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Fold. Des.* **1997**, *2* (3), 173−181.

(22) Hinsen, K. Analysis of domain motions by approximate normal mode calculations. *Proteins: Struct., Funct., Bioinf.* **1998**, *33* (3), 417−429.

(23) Li, G.; Cui, Q. A coarse-grained normal mode approach for macromolecules: An efficient implementation and application to Ca^{2+}-ATPase. *Biophys. J.* **2002**, *83* (5), 2457−2474.

(24) Tama, F.; Sanejouand, Y. H. Conformational change of proteins arising from normal mode calculations. *Protein Eng., Des. Sel.* **2001**, *14* (1), 1−6.
(25) Opron, K.; Xia, K.; Wei, G.-W. Fast and anisotropic flexibility-rigidity index for protein flexibility and fluctuation analysis. *J. Chem. Phys.* **2014**, *140* (23), 234105.
(26) Xia, K.; Opron, K.; Wei, G.-W. Multiscale Gaussian network model (mGNM) and multiscale anisotropic network model (mANM). *J. Chem. Phys.* **2015**, *143* (20), 204106.
(27) Paul, J. F. Statistical thermodynamics of random networks. *Proc. R. Soc. A: Math. Phys. Eng. Sci.* **1976**, *351* (1666), 351−380.
(28) Haliloglu, T.; Bahar, I.; Erman, B. Gaussian dynamics of folded proteins. *Phys. Rev. Lett.* **1997**, *79*, 3090−3093.
(29) Yang, L.-W.; Chng, C.-P. Coarse-grained models reveal functional dynamics - i. elastic network models − theories, comparisons and perspectives. *Bioinform. Biol. Insights* **2008**, *2* (S460), BBI.S460.
(30) Opron, K.; Xia, K.; Wei, G.-W. Communication: Capturing protein multiscale thermal fluctuations. *J. Chem. Phys.* **2015**, *142* (21), 211101.
(31) Schlessinger, A.; Rost, B. Protein flexibility and rigidity predicted from sequence. *Proteins: Struct., Funct., Bioinf.* **2005**, *61* (1), 115−126.
(32) de Brevern, A. G.; Bornot, A.; Craveur, P.; Etchebest, C.; Gelly, J. C. Predyflexy: flexibility and local structure prediction from sequence. *Nucleic Acids Res.* **2012**, *40* (W1), W317−W322.
(33) Vander Meersche, Y.; Cretin, G.; de Brevern, A. G.; Gelly, J.-C.; Galochkina, T. MEDUSA: prediction of protein flexibility from sequence. *J. Mol. Biol.* **2021**, *433* (11), 166882.
(34) Masters, M. R.; Mahmoud, A. H.; Wei, Y.; Lill, M. A. Deep learning model for efficient protein−ligand docking with implicit side-chain flexibility. *J. Chem. Inf. Model.* **2023**, *63* (6), 1695−1707.
(35) Song, X.; Bao, L.; Feng, C.; Huang, Q.; Zhang, F.; Gao, X.; Han, R. Accurate prediction of protein structural flexibility by deep learning integrating intricate atomic structures and cryo-em density information. *Nat. Commun.* **2024**, *15* (1), 5538.
(36) Xu, G.; Yang, Y.; Lv, Y.; Luo, Z.; Wang, Q.; Ma, J. Opus-bfactor: Predicting protein b-factor with sequence and structure information. *bioRxiv* **2024**.
(37) Wang, R.; Nguyen, D. D.; Wei, G. Persistent spectral graph. *Int. J. Numer. Methods Biomed. Eng.* **2020**, *36* (9), No. e3376.
(38) Chen, J.; Zhao, R.; Tong, Y.; Wei, G.-W. Evolutionary de rham-hodge method. *Discrete Contin. Dyn. Syst. - B* **2021**, *26* (7), 3785.
(39) Carlsson, G. Topology and data. *Bull. Amer. Math. Soc.* **2009**, *46* (2), 255−308.
(40) Edelsbrunner, H.; Harer, J.; et al. Persistent homology-a survey. *Contemp. Math.* **2008**, *453* (26), 257−282.
(41) Wei, X.; Wei, G.-W. Persistent sheaf Laplacians. *Found. Data Sci.* **2025**, *7* (2), 446−463.
(42) Wei, X.; Wei, G.-W. Persistent topological Laplacians−a survey. *Mathematics* **2025**, *13* (2), 208.
(43) Bernstein, F. C.; Koetzle, T. F.; Williams, G. J. B.; Meyer, E. F.; Brice, M. D.; Rodgers, J. R.; Kennard, O.; Shimanouchi, T.; Tasumi, M. The Protein Data Bank. *Eur. J. Biochem.* **1977**, *80* (2), 319−324.
(44) Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual molecular dynamics. *J. Mol. Graphics* **1996**, *14* (1), 33−38.
(45) Heinig, M.; Frishman, D. STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins. *Nucleic Acids Res.* **2004**, *32* (WebServer), W500−W502.
(46) Merkurjev, E. A fast graph-based data classification method with applications to 3d sensory data in the form of point clouds. *Pattern Recognit. Lett.* **2020**, *136*, 154−160.
(47) Garcia-Cardona, C.; Merkurjev, E.; Bertozzi, A. L.; Flenner, A.; Percus, A. G. Multiclass data segmentation using diffuse interface methods on graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36* (8), 1600−1613.
(48) Merkurjev, E.; Garcia-Cardona, C.; Bertozzi, A. L.; Flenner, A.; Percus, A. G. Diffuse interface methods for multiclass segmentation of high-dimensional data. *Appl. Math. Lett.* **2014**, *33*, 29−34.
(49) Cang, Z.; Wei, G. Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. *Int. J. Numer. Methods Biomed. Eng.* **2018**, *34* (2), No. e2914.
(50) Hansen, J.; Ghrist, R. Learning sheaf laplacians from smooth signals. In *ICASSP 2019−2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE, 2019.
(51) Russold, F. Persistent sheaf cohomology. *arXiv* **2022**, arXiv:2204.13446.
(52) Cang, Z.; Munch, E.; Wei, G.-W. Evolutionary homology on coupled dynamical systems with applications to protein flexibility analysis. *J. Appl. Comput. Topol.* **2020**, *4*, 481−507.

https://doi.org/10.1021/acs.jpcb.5c01287
J. Phys. Chem. B 2025, 129, 4169−4178