Wang Shuyu (Orcid ID: 0000-0002-0038-1347)

BayeStab: Predicting Effects of Mutations on Protein Stability with Uncertainty Quantification

Shuyu Wang[a,b,*], Hongzhou Tang[a], Yuliang Zhao[a,b] and Lei Zuo[c]

[a]Department of Control Engineering, Northeastern University, Qinhuangdao, Hebei, 066001, China.
[b] Department of Mechanical Engineering, Dalian University of Technology, Dalian, Liaoning,116024, China
[c]Department of Marine Engineering, University of Michigan, Ann Arbor, MI,48109, USA.
*To whom correspondence should be addressed. Email: vincentwang622@126.com
Hongzhou Tang and Shuyu Wang are co-first authors.

Abstract

Predicting protein thermostability change upon mutation is crucial for understanding diseases and designing therapeutics. However, accurately estimating Gibbs free energy change of the protein remained a challenge. Some methods struggle to generalize on examples with no homology and produce uncalibrated predictions. Here we leverage advances in graph neural networks for protein feature extraction to tackle this structure-property prediction task. Our method, BayeStab, is then tested on four test datasets, including S669, S611, S350, and Myoglobin, showing high generalization and symmetry performance. Meanwhile, we apply concrete dropout enabled Bayesian neural networks to infer plausible models and estimate uncertainty. By decomposing the uncertainty into parts induced by data noise and model, we demonstrate that the probabilistic method allows insights into the inherent noise of the training datasets, which is closely relevant to the upper bound of the task. Finally, the BayeStab web server is created and can be found at: http://www.bayestab.com. The code for this work is available at: https://github.com/HongzhouTang/BayeStab.

Keywords: protein stability change, graph neural network, concrete dropout, uncertainty quantification, web server

1 Introduction

A critical approach to investigate protein folding is to measure its thermodynamic properties. The folding process might be disturbed in mutated states, leading to changes in Gibbs free energy(ΔΔG). This change is sometimes desired in the pharmaceutical industry, as antibody drugs typically need high thermal stability[1]. Also, such a process is essential to understand how genome variation in drug targets can cause resistance to therapeutic drugs [2, 3].

To predict the stability change of proteins upon mutation with high throughput, computational approaches have been widely used. There were methods based on various evolutionary and physical chemical hypotheses with high performance. Another branch leveraged machine learning for fast identification, using techniques, such as support vector machine (SVM) [4-6], gradient boosting [7-9], artificial neural network (ANN)[10, 11], and combinations of them [12-20]. However, several studies pointed out the significantly biased results of the machine learning-based methods [21-23]. In other words, they predict the destabilizing mutation more than the stabilizing mutation, and the seemingly high linear correlation between predicted and experimental results might not be shown in the stabilizing mutations.

Recent studies based on deep learning techniques, such as the convolution neural network, seem to handle this issue well, showing symmetric prediction [24-27]. Generally, deep learning requires large amounts of training data to improve performance [28]. Currently, deep learning-based approaches have been demonstrated with high performance comparable to classic machine learning methods. With new collected data [29, 30] and potentially more in the future, it is not yet known how deep learning-based methods will perform.

One conundrum in this field is how to further improve the representation learning of the models when limited experimental data is available. The graph neural network (GNN) is a powerful tool for extracting information from graph data [31]. Graph convolutional networks apply spectral convolution in the graph Fourier domain to aggregate neighboring representations for feature

learning [32]. They have been used for protein structure refinement [33] and protein function prediction [34]. These attempts to encode protein context information make the prediction of mutation induced stability changes possible, yet it is still scarcely investigated.

method can be applied to investigate the inherent noise of the dataset, which is related to the upper bound performance[38]. The key difficulty in using Bayesian neural networks (BNNs) is that Bayesian inference is computationally intractable. To reduce computation cost, researchers proposed using dropout at test times

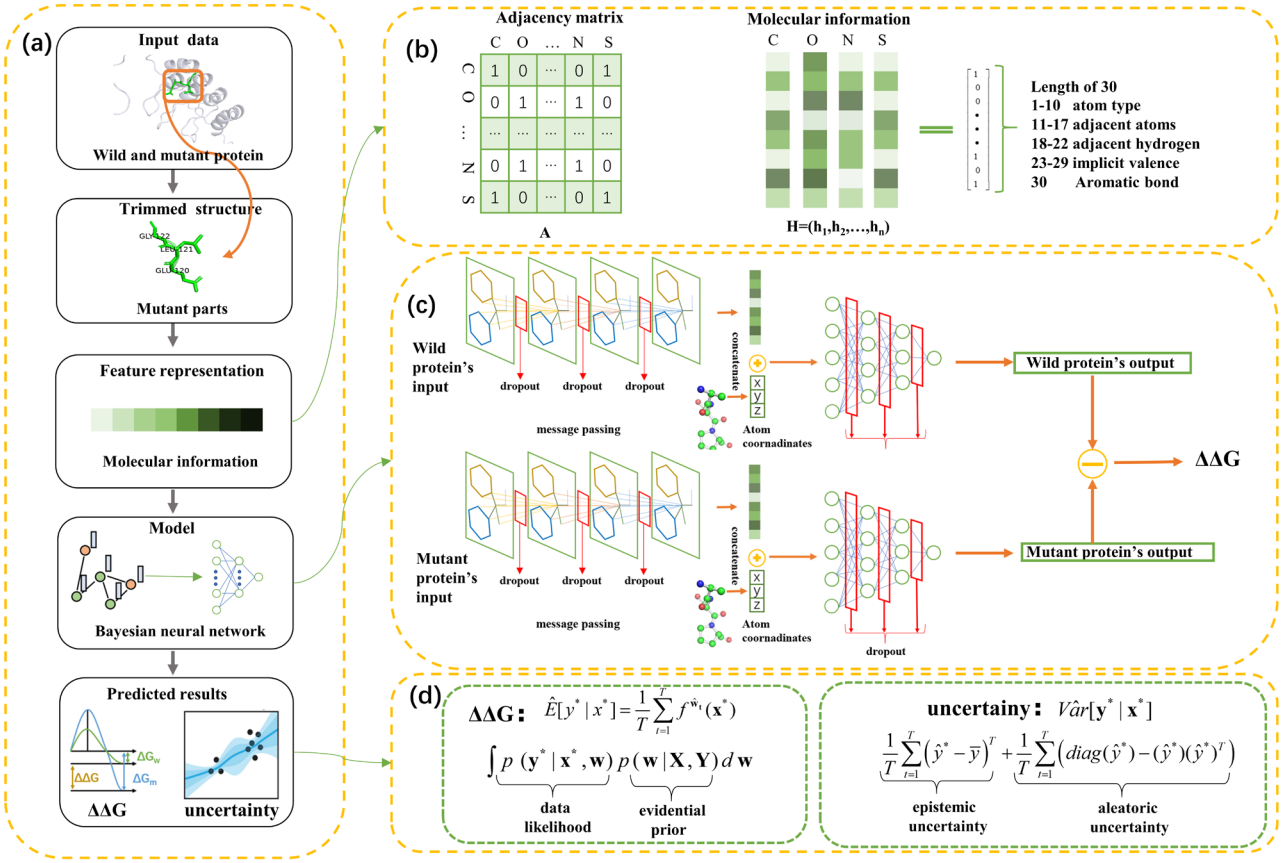

Fig 1. (a) The BayeStab's processing can be summarized into five steps: input the protein data, trim the non-mutant part, encode the protein vector representation, train the BNN, and predict the ΔΔG and uncertainty. (b) Illustration of the adjacency matrix and molecular information in the feature vectors. (c) The structure of the BayeStab model. (d) The underlying theory of Bayesian method to predict ΔΔG and quantify the uncertainty.

Overfitting is another critical challenge to consider in machine learning-based predictions. It happens when only limited experimental data is available, and the well-trained models might not generalize well on unobserved datasets. Thus, the model must be flexible enough to capture all properties of the data [35]. Probabilistic programming offers a way to generalize the models, allowing much richer representations of the model. It addresses this challenge by developing a distribution that encompasses the models using Bayesian theory [36]. The key idea behind the probabilistic machine learning is to infer plausible models from the data with uncertainty. Compared to a pure deep learning model, which predicts a definite output, Bayesian machine learning's prediction corresponds to the aggregation of different neural networks trained on the same dataset [37]. One advantage of the Bayesian approach is less prone to overfitting, since they are averaged over the parameters.

Meanwhile, the uncertainty quantified by the Bayesian

to enable uncertainty quantification of the predictive distribution [39]. Concrete dropout is a dropout variant which can be seen as a continuous relaxation of the discrete dropout. With appropriate regularization terms, this technique allows the dropout probability to be tuned using gradient methods and the uncertainty to be estimated.

Here we demonstrate that BNNs enabled by concrete dropout can be coupled with graph neural networks (GNN) to predict protein mutations' ΔΔGs and estimate the uncertainties. The molecular representations learned by the feature extractor are operated on graph networks. After being combined with the coordination of the atoms, they are then processed by fully connected layers to map the high-dimensional features to the low-dimensional properties. To enable faster training, we retained the mutant part only and trimmed the rest. Our deep learning model is trained end-to-end, from protein feature vectors to the output property (**Fig 1 (a)**).

We test our method on four public datasets, and the model outperforms previous approaches, showing improved generalization ability. Based on the BNN, we estimate the prediction uncertainty and decompose the uncertainty into parts induced by model data noise, which offers significant insights for investigating the upper bound of the performance. Last, BayeStab web server is presented to serve the broad scientific community.

2 Theoretical background

In this section, we first introduce the Bayesian inference model and variational inference as an approximation. Then, we illustrate how to quantify the uncertainty in a BNN. Next, we explain the working principle of our GNN.

2.1 Bayesian inference

Given a training set {**X**, **Y**}, where X is the protein feature and Y is ΔΔG upon mutation. $\boldsymbol{p}$ (**Y**|**X**, $\boldsymbol{w}$) is the likelihood of the model and $\boldsymbol{p}$(**w**) is the prior distribution. **w** = {$\boldsymbol{W_1}$, ..., $\boldsymbol{W_k}$} is the model parameters with a structure of $\boldsymbol{k}$ layers structure. In a Bayesian framework, the posterior is calculated as:

$$p(\mathbf{w}\mid\mathbf{X},\mathbf{Y})=\frac{p(\mathbf{Y}\mid\mathbf{X},\mathbf{w})p(\mathbf{w})}{p(\mathbf{Y}\mid\mathbf{X})} \tag{1}$$

The predictive distribution of the problem can be defined as follows:

$$p(\mathbf{y}^*\mid\mathbf{x}^*,\mathbf{X},\mathbf{Y})=\int p(\mathbf{y}^*\mid\mathbf{x}^*,\mathbf{w})\,p(\mathbf{w}\mid\mathbf{X},\mathbf{Y})\,d\mathbf{w} \tag{2}$$

where $\mathbf{y}^*$ is the output of input $\mathbf{x}^*$ for a given $\boldsymbol{w}$.

Direct application of the formula is impractical due to the high computation cost. Variational inference can approximate the posterior using a tractable distribution $\boldsymbol{q_\theta}$ ($\boldsymbol{w}$) parameterized by the parameter $\boldsymbol{\theta}$. By minimizing the Kullback-Leibler(KL) divergence,

$$\mathrm{KL}(q_\theta(\mathbf{w})\,\|\,p(\mathbf{w}\mid\mathbf{X},\mathbf{Y}))=\int_\Omega q_\theta(\mathbf{w})\log\frac{q_\theta(\mathbf{w})}{p(\mathbf{w}\mid\mathbf{X},\mathbf{Y})}d\mathbf{w} \tag{3}$$

we can combine the intractable posterior distribution in eq (3) with eq (1). Then, the variational approximation of the negative evidence lower-bond becomes:

$$\mathrm{L}_{VI}(\theta)=-\int_\Omega q_\theta(\mathbf{w})\,\log\mathrm{p}(\mathbf{Y}|\mathbf{X},\mathbf{w})d\mathbf{w}+\mathrm{KL}(q_\theta(\mathbf{w})\,\|\,p(\mathbf{w})) \tag{4}$$

To implement a Bayesian model, $\boldsymbol{q_\theta}$ (**w**) is needed. Concrete dropout inside a neural network can approximate the posterior distribution without extra learnable parameters, and the integral across the full parameter space can be retrieved by Monte Carlo (MC) sampling.

2.2 Quantification of uncertainty with BNN

Given a new input $\mathbf{x}^*$, the variational distribution of the output, $\mathbf{y}^*$, can be obtained as:

$$q_\theta(\mathbf{y}^*\mid\mathbf{x}^*)=\int p(\mathbf{y}^*\mid f^{\mathbf{w}}(\mathbf{x}^*))q_\theta(\mathbf{w})d\mathbf{w} \tag{5}$$

where $f^{\mathbf{w}}(\mathrm{x}^*)$ is the output of the model for a given **w**. The predictive mean of this distribution with T times of MC sampling is estimated for regression tasks by:

$$\hat{E}[y^*\mid x^*]=\frac{1}{T}\sum_{t=1}^{T}f^{\hat{\mathbf{w}}_t}(\mathbf{x}^*) \tag{6}$$

and a predictive variance is estimated by:

$$\hat{Var}[\mathbf{y}^*\mid\mathbf{x}^*]=\frac{1}{T}\sum_{t=1}^{T}f^{\hat{\mathbf{w}}_t}(\mathbf{x}^*)^T f^{\hat{\mathbf{w}}_t}(\mathbf{x}^*)-\hat{E}[\mathbf{y}^*\mid\mathbf{x}^*]^T\hat{E}[\mathbf{y}^*\mid\mathbf{x}^*] \tag{7}$$

The uncertainty can be divided in two parts: aleatoric and epistemic uncertainty. The aleatoric uncertainty is inherent in the noise from the datasets, while the epistemic uncertainty is caused by the prediction of the model. The uncertainty's segmentation is as follows:

$$\hat{Var}[\mathbf{y}^*\mid\mathbf{x}^*]=\underbrace{\frac{1}{T}\sum_{t=1}^{T}\left(\hat{\mathbf{y}}_t^*-\bar{\mathbf{y}}\right)\left(\hat{\mathbf{y}}_t^*-\bar{\mathbf{y}}\right)^T}_{\text{epostemic}}+\underbrace{\frac{1}{T}\sum_{t=1}^{T}\left(diag(\hat{\mathbf{y}}_t^*)-(\hat{\mathbf{y}}_t^*)(\hat{\mathbf{y}}_t^*)^T\right)}_{aleatoric} \tag{8}$$

where $\bar{\mathrm{y}}=\sum_{t=1}^{T}\hat{\mathrm{y}}_t^*/\mathrm{T}$. $\hat{\mathrm{y}}_t^*=softmax(\mathrm{f}^{\hat{\mathbf{w}}_t}(\mathrm{x}_t))$, and $\mathrm{f}^{\hat{\mathbf{w}}_t}(\mathrm{x}_t)$ is the neural network's output with input x_t.

2.3 Graph neural network for feature learning

The inputs to the graph neural network $X = H^{(0)}$ are the adjacency matrix, A, and the initial node features, which consisted of atom types, adjacent atoms, number of adjacent hydrogen, implicit valence and aromatic bonds (**Fig1 (b)**).

The GNN's message passing through a single layer is as follows:

$$\boldsymbol{H^{(l+1)}=Leaky\_relu(W^{(l)}AH^{(l)})} \tag{9}$$

where $\boldsymbol{H^{(l)}}$ and $\boldsymbol{W^{(l)}}$ are node features and trainable parameters at the l-th layer, $l \in \{0, \cdots, \mathrm{L}\}$, respectively. The GNN updates the node feature $\boldsymbol{H^{(l+1)}}$ with information from adjacent nodes for representation learning.

To improve the feature extraction performance, we integrated the gating mechanism into the network as:

$$\boldsymbol{H_{gate}^{(l+1)}=GH_{gate}^{(l+1)}+(1-G)H_{gate}^{(l)}} \tag{10}$$

with

$$\boldsymbol{G=leaky\_relu(W_{gate}\left[H_{gate}^{(l)}\;H_{gate}^{(l+1)}\right]+B)} \tag{11}$$

After updating the mode features L-times through feedforward computations, the graph feature $\boldsymbol{h}_\mathrm{G}$ is obtained by summation of all $\boldsymbol{N}$ node:

$$\boldsymbol{h_G}=\sum_{n\in N}\mathrm{NN}(\boldsymbol{H_n^{(L)}}) \tag{12}$$

3 Experiments and Methods

3.1 Datasets

S2648 contains 2648 single point mutations from 131 different globular proteins. The ProTherm database is the source of the dataset. In this dataset, 2,080 of them are destabilizing and 568 are stabilizing. We use S2648 as the training dataset for BayeStab.

Q3421 includes 3421 mutations from 150 proteins. We use the dataset for 10-fold cross-validation.

S350 consists of 350 mutations in 67 different proteins. It is a subset of the S2648 dataset, so the overlapped part needs to be tailored during training.

S611 is developed by DynaMut2[17], which is split from a dataset of 4,633 mutations.

S669 is a latest curated test dataset[40] manually cleaned from the ThermoMutDB database. It consists of 669 variants of protein sequences that do not share homology with the S2648 dataset and Varibench.

Myoglobin is the globular protein that regulates the concentration of cellular oxygen[41]. The dataset consists of 134 mutations scattered throughout the protein chain, which also does not overlap with the training dataset.

S^{sym} contains 684 variations, and half of them are reverse variations with crystal structures of the corresponding mutant proteins[42]. We use the S^{sym} dataset to investigate the uncertainty in the dataset and in the model.

3.2 Implementation and evaluation

The schematic view of the BayeStab is shown in **Fig 1(c)**, and the sizes of each layer in the architecture are listed in **Table 1**.

Table 1 The architecture of the BayeStab

Layer type	Specifications
GNN layer + Dropout ×4	Size:1400
FC layer +Dropout + ReLU	Size:1024
FC layer +Dropout + ReLU	Size:512
FC layer +Dropout + ReLU	Size:256
FC layer +Dropout	Size:1

The two branches for processing wild and mutant proteins are symmetric, with both the GNN module and the FC module. The summation of the atom coordinates is concatenated to the latent feature extracted by the GNN. Finally, the output of the wild protein is subtracted from the mutant protein to obtain the ΔΔG. At each hidden layer, we applied the concrete dropout, which leads to the corresponding uncertainty estimation. The principles for quantifying and decomposing the uncertainty are also illustrated in **Fig 1(d)**.

In the training phase, we used the Adam optimizer with the learning rate of 10^{-3} for 400 epochs. The dropout was performed at the inference phases, sampled with $T = 10$ for Bayesian inference. The model was implemented using Pytorch on a GTX-3070 processor.

To evaluate the prediction accuracy, we use the Pearson correlation coefficient (r) between experimental and predicted ΔΔGs and the root mean squared error (σ) of predictions. To quantify the prediction bias, we adopt r between the predicted results for direct mutations and reverse mutations and the error, $\delta = \Delta\Delta G_{rev} + \Delta\Delta G_{dir}$ [42].

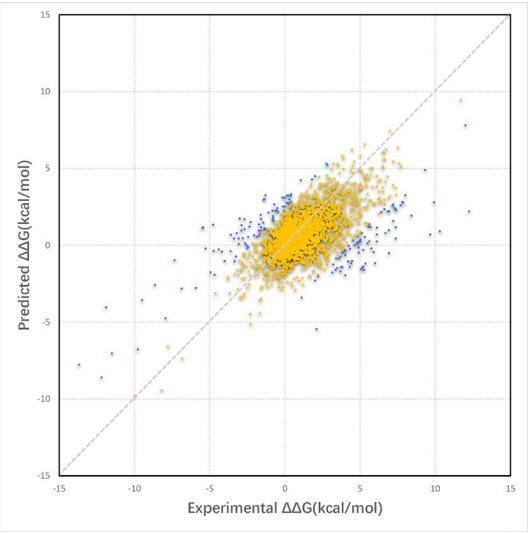

Fig 3. Cross validation results of the Q3421 dataset. With 5% of the outliers removed(blue dots), r=0.68, σ=1.29 kcal/mol.

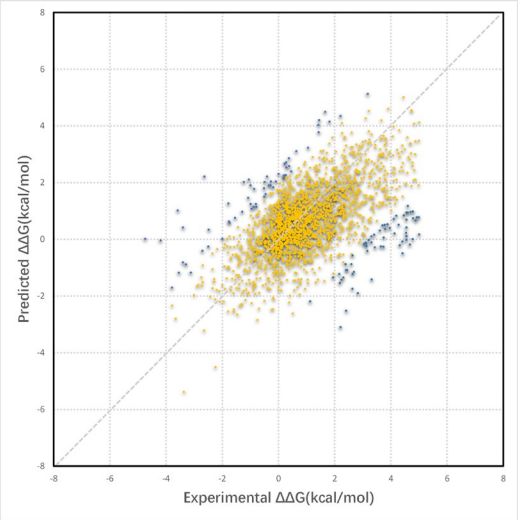

Fig 2. Cross validation results of the S2648 dataset. With 5% of the outliers removed(blue dots), r=0.69, σ=1.06 kcal/mol.

4 Result and Discussion

4.1 Testing results on four datasets

After 10-fold cross-validation of the S2648 dataset, BayeStab showed r = 0.61 and σ = 1.19 kcal/mol. The Pearson correlation coefficient increased to 0.69 and σ decreased to 1.06 kcal/mol after removing 5% of the outliers (**Fig 2**). When we performed a 10-fold cross-validation on the Q3421 dataset, r was 0.68, and σ was reduced to 1.29 kcal/mol, if 5% of the outliers were removed (**Fig 3**).

Then, we tested the trained model on S611, S350, Myoglobin, and S669 datasets, respectively. Before training, the overlap between the training and testing datasets were tailored for assessment. Since BayeStab can predict with the corresponding uncertainty, we marked the data points with various colors to indicate its probability(**Fig4**).

When evaluated using the S611 dataset, BayeStab obtained r = 0.73, σ = 0.99 kcal/mol in the direct mutations, r = 0.73, σ = 0.99 kcal/mol in the reverse mutations, and r = -0.97, δ = 0.01 in direct-reverse prediction (**Fig 4(a)-(c)**). We further analyze the performance of the stabilizing and destabilizing mutations, respectively. BayeStab's performance on destabilizing and stabilizing mutations were r = 0.72, σ = 1.02 kcal/mol and r = 0.48, σ = 1.28 kcal/mol. Comparing with other methods, BayeStab improved performance on the overall(**Table 2**).

Table 2 Comparison of different methods tested on the S611 dataset.

	Overall		Stabilizing mutations		Destabilizing mutations	
Method	σ	r	σ	r	σ	r
BayeStab	**0.99**	**0.73**	**1.28**	**0.48**	**1.02**	**0.72**
DUET	1.40	0.48	1.75	0.09	1.00	0.58
DynaMut2	1.14	0.68	1.02	0.51	0.91	0.62
SDM	1.93	0.35	1.62	0.48	-0.77	0.03
mCSM	1.42	0.46	1.81	0.11	0.98	0.56

MAESTRO	1.55	-0.36	1.17	0..27	1.81	0.43
I-mutant	1.47	0.33	1.83	0.03	1.09	0.49

Wang Shuyu (Orcid ID: 0000-0002-0038-1347)

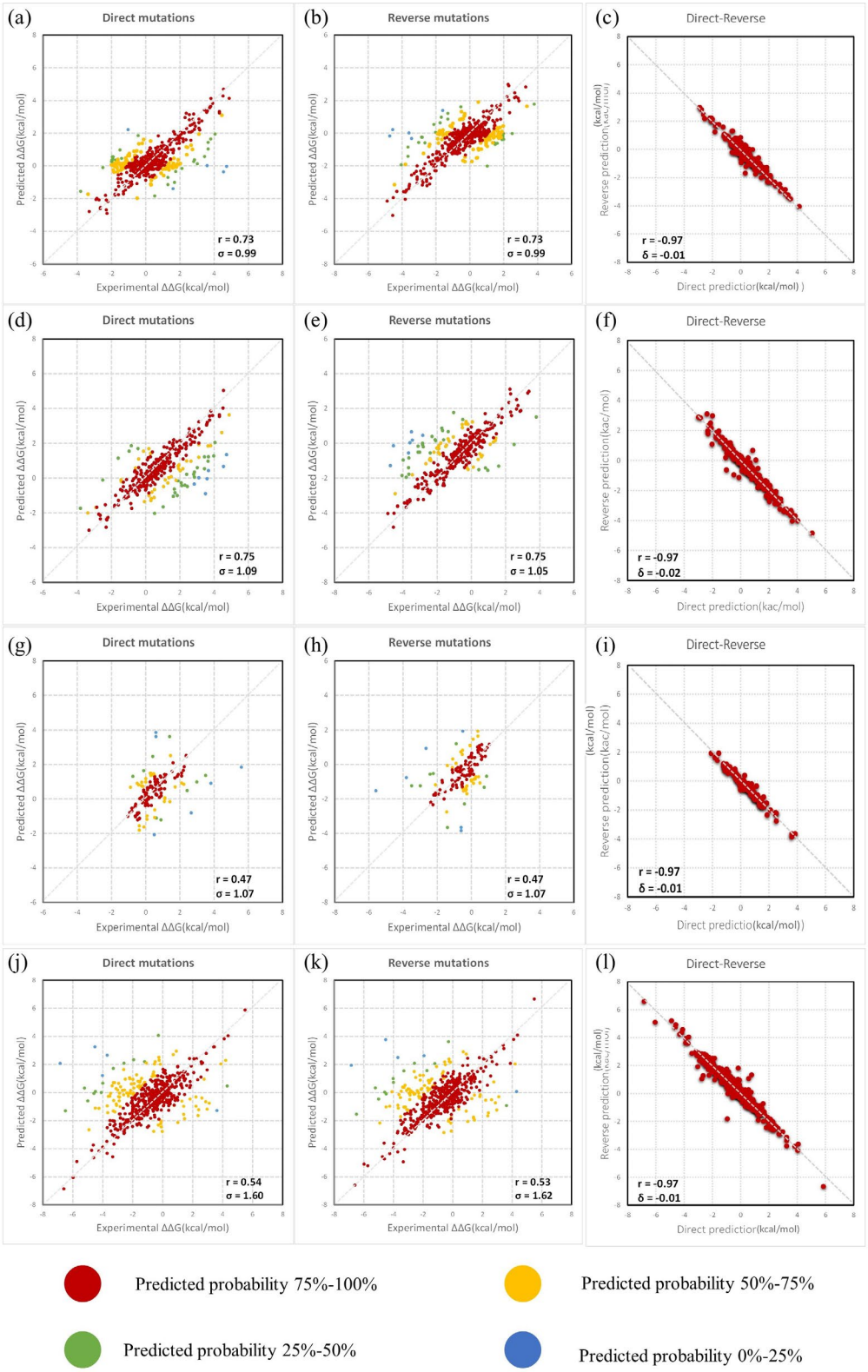

Fig 4. BayeStab's performance when tested on four datasets. The corresponding prediction uncertainty is marked using four different colors. (a) Predicting ΔΔG for direct mutations in S611, (b) reverse mutations in S611, (c) direct versus reverse ΔΔG values in S611. (d) Predicting ΔΔG for direct mutations in S350, (e) reverse mutations in S350, (f) direct versus reverse ΔΔG values in S350. (g) Predicting ΔΔG for direct mutations in Myoglobin, (h) reverse mutations in Myoglobin, (i) direct versus reverse ΔΔG values in Myoglobin, (j) predicting ΔΔG for direct mutations in S669, (k) reverse mutations in S669, (l) direct versus reverse ΔΔG values in S669.

Wang Shuyu (Orcid ID: 0000-0002-0038-1347)

Next, BayeStab was tested on the S350 dataset and achieved r = 0.75, σ = 1.09 kcal/mol in direct mutations, r = 0.75, σ = 1.05 kcal/mol in reverse mutations, and r = -0.97, δ = -0.02 kcal/mol in direct-reverse prediction (**Fig 4(d)-(f)**). Meanwhile, we split the results into stabilizing and destabilizing mutations. We found that BayeStab's strong performances on stabilizing mutations were r = 0.66, σ = 1.29 kcal/mol, and destabilizing mutations showed r = 0.62, σ = 1.37 kcal/mol. Our results tested on S350 dataset were also compared with six other methods (**Table 3**). BayeStab's performance also exceeded prior methods when dealing with the imbalance problem.

Table 3 Comparison of different methods tested on the S350 dataset.

	Overall		Stabilizing mutations		Destabilizing mutations	
Method	σ	r	σ	r	σ	r
BayeStab	**1.09**	**0.75**	**1.29**	**0.66**	**1.37**	**0.62**
DUET	1.31	0.67	1.00	0.65	2.23	0.28
DynaMut2	1.37	0.66	1.16	0.63	2.01	0.38
SDM	1.80	0.52	1.43	0.42	3.12	0.15
mCSM	1.08	0.66	1.01	0.63	2.48	0.31
MAESTRO	1.79	0.55	1.52	0.43	1.37	0.61
I-mutant	1.75	0.53	1.42	0.42	2.89	0.25

The Myoglobin dataset does not overlap with the training data, indicating that it is appropriate for estimating overfitting. Our tested results on this dataset were r = 0.47, σ = 1.07 kcal/mol on direct mutations, r = 0.47, σ = 1.07 kcal/mol on reverse mutations, and r = -0.97, δ = -0.01 kcal/mol on the direct-reverse predictions (**Fig 4(g)-(i)**).

The latest curated dataset, S669, is also highly convincing for performance evaluation, since it is not included in the widely available training datasets. On the S669 dataset, BayeStab also achieved superior symmetry, showing r = -0.97, δ = -0.01 kcal/mol for direct-reverse prediction. Its performance on direct mutations reached r = 0.54, σ = 1.60 kcal/mol, and MAE = 1.07 kcal/mol. The reverse mutations showed r = 0.53, σ = 1.62 kcal/mol, and MAE = 1.07 kcal/mol (**Fig 4(j)-(l)**). Fifteen recently shown methods were also listed for comparison with BayeStab (**Table 4**). Our method's performance is highly competitive to be the state-of-the-art approach, showing highest linear correlation and improved symmetry.

Table 4 BayeStab compared with 15 recent methods tested on the S669 dataset. The data is adopted from[40]**.**

	Direct			Reverse			Dir-rev	
Method	r	σ	MAE	r	σ	MAE	r_{d-r}	δ
BayeStab	**0.54**	**1.60**	**1.07**	**0.53**	**1.62**	**1.07**	**-0.97**	**-0.01**
ACDC-NN	0.46	1.49	1.05	0.45	1.50	1.06	-0.98	-0.02
DDGun3D	0.43	1.60	1.11	0.41	1.62	1.14	-0.97	-0.05
PremPS	0.41	1.50	1.08	0.42	1.49	1.05	-0.85	0.09
ThermoNet	0.39	1.62	1.17	0.38	1.66	1.23	-0.85	-0.05
Rosetta	0.39	2.70	2.08	0.40	2.68	2.02	-0.72	-0.61
Dynamut	0.41	1.6	1.19	0.34	1.69	1.24	-0.58	-0.06
INPS3D	0.43	1.5	1.07	0.33	1.77	1.31	-0.50	-0.06
SDM	0.41	1.67	1.26	0.13	2.16	1.64	-0.40	-0.40
PopMuSic	0.41	1.51	1.09	0.24	2.09	1.64	-0.32	-0.69
MAESTRO	0.50	1.44	1.06	0.20	2.10	1.65	0.22	-0.57
FoldX	0.22	2.30	1.56	0.22	2.48	1.50	-0.20	-0.34
DUET	0.41	1.52	1.10	0.23	2.14	1.68	-0.12	-0.67
I-Mutant3.0	0.36	1.52	1.12	0.15	2.32	1.87	-0.06	-0.81
mCSM	0.36	1.54	1.13	0.22	2.30	1.86	-0.05	-0.85
Dynamut2	0.34	1.58	1.15	0.17	2.16	1.69	0.03	-0.64

4.2 Uncertainty decomposition

We then decomposed the uncertainties obtained from BayeStab and compared the uncertainties with all, 1/2, and 1/4 of the training dataset. When we tested on the S^{sym} dataset, we found the aleatoric uncertainty remained almost unchanged, whereas the epistemic uncertainty increased as the amount of training data decreased. This effect can be explained as the model-induced uncertainty increased due to the less training data, while the uncertainty inherent in the experimental data remained the same.

Besides, we could estimate how much noise from the dataset contributed to the predicted error. For the past two decades, the performance of the machine learning-based method seemed to have an upper bound. The prediction error, σ, stagnated at around 1 kcal/mol, yet the inherent noise of the dataset was rarely explored.

With BNN's powerful uncertainty division, we may find the dataset's noise is dominant in the overall uncertainty, indicating the model has almost reached the upper bound performance with the data available. More experimental data with the current measurement accuracy may not lead to higher performance, as the epistemic uncertainty is already very small compared with the aleatoric uncertainty.

Table 5 BayStab estimated the epistemic and aleatoric uncertainties when trained using various amounts of the S2648 dataset and tested on the S^{sym} dataset.

Training Dataset	Epistemic	Aleatoric
S2648	0.03	0.25
S2648 / 2	0.08	0.24
S2648 / 4	0.13	0.25

4.3 Web Server

We built a freely available and user-friendly web server (http://www.bayestab.com) using Flask. The home page and the result page of the web server are shown in **Fig 5.**

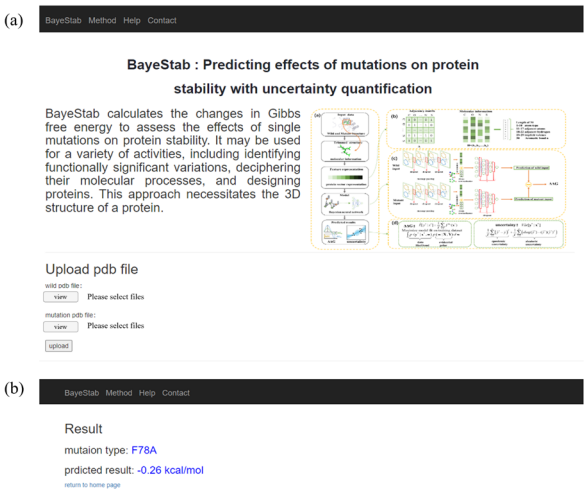

Fig 5. BayeStab web server. (a) The home page (b) and the result page of the web server.

The web server takes the structure information of the protein as the input. Users can upload PDB files of the wild type and mutant types to the server. The mutant type PDB files can be generated by Rosetta. Next, the user needs to fill in the mutation information. For example, L37S indicates that at the position of amino acid number 37, and leucine (L) becomes serine (S). Users also need to fill in the mutant protein chain information, such as A or B. Last, the user can get the predicted ΔΔG after submitting the task.

5 Conclusion

Here, we fuse the BNN and GNN-based methods to predict proteins' stability change upon mutations with quantified uncertainty. Our end-to-end deep learning model, BayeStab, can effectively learn molecular feature representations to predict the ΔΔG with significantly high performance.

The cross-validations on S2648 and Q3421 datasets show high linearity and low errors. Superior performance is also demonstrated when tested on four datasets. The predicted results are highly symmetric between direct and reverse mutations without bias towards predicting destabilization. The test results on the S669 are especially persuasive for proving BayeStab's improved generalization, as it has novel variants never encountered by the prior prediction tools. BayeStab achieved high Pearson correlation coefficients that outperformed state-of-the-art methods.

In addition, we propose to integrate concrete dropout in the GNN as our Bayesian approach to quantify the uncertainty, then we further decompose the uncertainty to model-induced and data noise-induced parts. To the best knowledge of the authors, this is a novel work to introduce uncertainty quantification into this field. Using the model trained on S2648 and tested on S^{sym}, we find the noise from the dataset is dominant in the prediction errors, indicating that the prediction upper bound is almost approaching. We also suspect that even if more experimental data is available, the improvement might still be subtle.

Last, BayeStab is also made accessible to wider users through a free web server. In the future, we hope BayeStab would benefit the research community to study protein dynamics and envision its contribution to deepen the understanding of mutations in diseases.

6 Acknowledgment

The author, Shuyu Wang, thanks for funding from the National Natural Science Foundation of China (No.62104034), the Natural Science Foundation of Hebei Province (No. F2020501033), and Fundamental Research Funds for the Central Universities(N2223032).

7 Conflict of Interest

The authors declare no conflict of interest.

References:

[1] V. Gapsys, S. Michielssens, D. Seeliger, B.L. de Groot, Accurate and Rigorous Prediction of the Changes in Protein Free Energies in a Large-Scale Mutation Scan, Angewandte Chemie International Edition 55(26) (2016) 7364-7368.

[2] S. Wan, D. Kumar, V. Ilyin, U. Al Homsi, G. Sher, A. Knuth, P.V. Coveney, The effect of protein mutations on drug binding suggests ensuing personalised drug selection, Sci Rep-Uk 11(1) (2021) 13452.

[3] G. Hao, G. Yang, C. Zhan, Structure-based methods for predicting target mutation-induced drug resistance and rational drug design to overcome the problem, Drug Discov Today 17(19) (2012) 1121-1126.

[4] D.E.V. Pires, D.B. Ascher, T.L. Blundell, DUET: A Server for Predicting Effects of Mutations on Protein Stability Using an Integrated Computational Approach, Nucleic Acids Res 42 (2014) 314-319.

[5] E. Capriotti, P. Fariselli, R. Casadio, I-Mutant2.0: predicting stability changes upon mutation from the protein sequence or structure., Nucleic Acids Res 33 (2005) 306-310.

[6] P. Fariselli, P.L. Martelli, C. Savojardo, R. Casadio, INPS: predicting the impact of non-synonymous variations on protein stability from sequence, Bioinformatics 31(17) (2015) 2816-2821.

[7] Y. Yang, X. Ding, G. Zhu, A. Niroula, Q. Lv, M. Vihinen, ProTstab – predictor for cellular protein stability, Bmc Genomics 20(1) (2019) 1-9.

[8] D.K. Witvliet, A. Strokach, A.F. Giraldo-Forero, J. Teyra, R. Colak, P.M. Kim, ELASPIC web-server: proteome-wide structure-based prediction of mutation effects on protein stability and binding affinity, Bioinformatics 32(10) (2016) 1589-1591.

[9] L. Quan, Q. Lv, Y. Zhang, STRUM: structure-based prediction of protein stability changes upon single-

point mutation, Bioinformatics 32(19) (2016) 2936-2946.
[10] Y. Dehouck, J.M. Kwasigroch, D. Gilis, M. Rooman, PoPMuSiC 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality, Bmc Bioinformatics 12(1) (2011) 151.
[11] E. Capriotti, P. Fariselli, R. Casadio, A neural-network-based method for predicting protein stability changes upon single point mutations, Intelligent Systems in Molecular Biology 20(1) (2004) 63-68.
[12] D.E.V. Pires, D.B. Ascher, T.L. Blundell, mCSM: predicting the effects of mutations in proteins using graph-based signatures, Bioinformatics 30(3) (2014) 335-342.
[13] J. Laimer, H. Hofer, M. Fritz, S. Wegenkittl, P. Lackner, MAESTRO - multi agent stability prediction upon point mutations, Bmc Bioinformatics 16(1) (2015) 116-116.
[14] C.H.M. Rodrigues, D.E.V. Pires, D.B. Ascher, DynaMut: predicting the impact of mutations on protein conformation, flexibility and stability, Nucleic Acids Res 46 (2018).
[15] A.P. Pandurangan, B. Ochoa-Montaño, D.B. Ascher, T.L. Blundell, SDM: a server for predicting effects of mutations on protein stability, Nucleic Acids Res 45(W1) (2017) W229-W235.
[16] M. Giollo, A.J. Martin, I. Walsh, C. Ferrari, S.C. Tosatto, NeEMO: a method using residue interaction networks to improve prediction of protein stability upon mutation, Bmc Genomics 15(S4) (2014) 1-11.
[17] C.H.M. Rodrigues, D.E.V. Pires, D.B. Ascher, DynaMut2: Assessing changes in stability and flexibility upon single and multiple point missense mutations, Protein Sci 30(1) (2021) 60-69.
[18] Z. Cang, G. Wei, Analysis and prediction of protein folding energy changes upon mutation by element specific persistent homology, Bioinformatics 33(22) (2017) 3549-3557.
[19] C. Chen, M. Lin, C. Liao, H. Chang, Y. Chu, iStable 2.0: Predicting protein thermal stability changes by integrating various characteristic modules, Comput Struct Biotec 18 (2020) 622-630.
[20] Y. Chen, H. Lu, N. Zhang, Z. Zhu, S. Wang, M. Li, PremPS: Predicting the impact of missense mutations on protein stability, Plos Comput Biol 16(12) (2020) e1008543.
[21] L. Montanucci, C. Savojardo, P.L. Martelli, R. Casadio, P. Fariselli, On the biases in predictions of protein stability changes upon variations: the INPS test case, Bioinformatics 35(14) (2019) 2525-2527.
[22] J. Fang, A critical review of five machine learning-based algorithms for predicting protein stability changes upon mutation, Brief Bioinform 21(4) (2020) 1285-1292.
[23] F. Pucci, M. Schwersensky, M. Rooman, Artificial intelligence challenges for predicting the impact of mutations on protein stability, Curr Opin Struc Biol 72 (2022) 161-168.
[24] B. Li, Y.T. Yang, J.A. Capra, M.B. Gerstein, Predicting changes in protein thermodynamic stability upon point mutation with deep 3D convolutional neural networks, Plos Comput Biol 16(11) (2020) e1008291.
[25] S. Benevenuta, C. Pancotti, P. Fariselli, G. Birolo, T. Sanavia, An antisymmetric neural network to predict free energy changes in protein variants, Journal of Physics D: Applied Physics 54(24) (2021) 245403.
[26] H. Cao, J. Wang, L. He, Y. Qi, J.Z. Zhang, DeepDDG: Predicting the Stability Change of Protein Point Mutations Using Neural Networks, J Chem Inf Model 59(4) (2019) 1508-1514.
[27] L. Montanucci, E. Capriotti, Y. Frank, N. Ben-Tal, P. Fariselli, DDGun: an untrained method for the prediction of protein stability changes upon single and multiple point variations, Bmc Bioinformatics 20(S14) (2019) 335-335.
[28] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521(7553) (2015) 436-444.
[29] J.S. Xavier, T. Nguyen, M. Karmarkar, S. Portelli, P.M. Rezende, J.P.L. Velloso, D.B. Ascher, D.E.V. Pires, ThermoMutDB: a thermodynamic database for missense mutations, Nucleic Acids Res 49(D1) (2021) D475-D479.
[30] A. Nisthal, C.Y. Wang, M.L. Ary, S.L. Mayo, Protein stability engineering insights revealed by domain-wide comprehensive mutagenesis, Proceedings of the National Academy of Sciences 116(33) (2019) 16367.
[31] M. Bronstein, J. Bruna, Y. Lecun, A. Szlam, P. Vandergheynst, Geometric Deep Learning: Going beyond Euclidean data, Ieee Signal Proc Mag 34 (2016).
[32] T. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, 2016.
[33] X. Jing, J. Xu, Fast and effective protein model refinement using deep graph neural networks, Nature Computational Science 1(7) (2021) 462-469.
[34] B. Lai, J. Xu, Accurate protein function prediction via graph attention networks with predicted structure information, Brief Bioinform (2021) bbab502.
[35] O. Caldararu, R. Mehra, T.L. Blundell, K.P. Kepp, Systematic Investigation of the Data Set Dependency of Protein Stability Predictors, J Chem Inf Model 60(10) (2020) 4772-4784.
[36] Z. Ghahramani, Probabilistic machine learning and artificial intelligence, Nature 521(7553) (2015) 452-9.
[37] Q. Kim, J. Ko, S. Kim, W. Jhe, Bayesian neural network with pretrained protein embedding enhances prediction accuracy of drug-protein interaction, Bioinformatics 37 (2021).
[38] L. Montanucci, P.L. Martelli, N. Ben-Tal, P. Fariselli, A natural upper bound to the accuracy of predicting

protein stability changes upon mutations, Bioinformatics (Oxford, England) 35 (2018).
[39] Y. Gal, J. Hron, A. Kendall, Concrete Dropout, 2017. arXiv preprint arXiv:1705.07832v1
[40] C. Pancotti, S. Benevenuta, G. Birolo, V. Alberini, V. Repetto, T. Sanavia, E. Capriotti, P. Fariselli, Predicting protein stability changes upon single-point mutation: a thorough comparison of the available tools on a new dataset, Brief Bioinform 23 (2022).
[41] G.A. Ordway, D.J. Garry, Myoglobin: an essential hemoprotein in striated muscle, J Exp Biol 207(20) (2004) 3441-3446.
[42] F. Pucci, K.V. Bernaerts, J.M. Kwasigroch, M. Rooman, Quantification of biases in predictions of protein stability changes upon mutations, Bioinformatics 34(21) (2018) 3659-3665.