Bioinformatics, YYYY, 0–0
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: DD Month YYYY
Original Paper

Data and text mining

ProS-GNN: Predicting effects of mutations on protein stability using graph neural networks

Shuyu Wang[1,*], Hongzhou Tang[1], Peng Shan[1], and Lei Zuo[2]

[1]Department of Control Engineering, Northeastern University, Qinhuangdao, Hebei, 066001, PR China, [2]Department of Electrical Engineering, Virginia Tech, Blacksburg, VA 24061, USA.

*To whom correspondence should be addressed. Hongzhou Tang and Shuyu Wang are co-first authors.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Predicting protein stability change upon variation through computational approach is a valuable tool to unveil the mechanisms of mutation-induced drug failure and help to develop immunotherapy strategies. However, some machine learning based methods tend to be overfitting on the training data or show anti-symmetric biases between direct and reverse mutations. Moreover, this field requires the methods to fully exploit the limited experimental data.
Results: Here we pioneered a deep graph neural network based method for predicting protein stability change upon mutation. After mutant part data extraction, the model encoded the molecular structure-property relationships using message passing and incorporated raw atom coordinates to enable spatial insights into the molecular systems. We trained the model using the S2648 and S3412 datasets, and tested on the S^{sym} and Myoglobin datasets. Compared to existing methods, our proposed method showed competitive high performance in data generalization and bias suppression with ultra-low time consumption. Furthermore, method was applied to predict the Pyrazinamide's Gibbs free energy change for a real case study.
Availability: https://github.com/shuyu-wang/ProS-GNN.
Contact: vincentwang622@126.com

1 Introduction

Proteins are composed of amino acids sequences, arranged into different groups. Changes to an amino acid due to DNA variation is called a missense mutation. Such mutations may result in changes of protein stability or protein misfolding[1]. One way to infer mutation induced protein stability change is to measure its ΔΔG. It refers to the change in folding energy between the mutant and wild state. The negative sign of ΔΔG indicates the variation decreases protein stability, and the positive sign means stability increases. Current study shows protein stability changes is one of the major underlying molecular mechanisms in multiple mutation-induced diseases[2]. Moreover, deeper insights into how specific mutations affect protein stability or interactions can identify possible drug resistance or sensitivity in patients[3], potentially leading to new precision medicines. This is especially important for genomic diseases such as cancers[4].

Recognizing the great potential of predicting protein stability changes upon mutation, researchers developed different computational tools, since they are low cost and high throughput. Prior attempts to predict protein stability were developed by molecular dynamics simulation[5]. They estimated free energy functions from protein structures using principles of statistical physics[6] or use knowledge-based terms of biophysical characteristics for regression fitting based on molecular mechanics[7, 8]. These methods showed advantages in characterizing the structural changes and the physical nature of the predicted folding free energy changes[9].

Data driven methods based on machine learning (ML)technologies are another promising branch to predict protein stability changes upon mutation. They are appealing due to efficient computation and high performance[10, 11]. Algorithms, such as support vector machines (SVM) [12-15], decision tree[16, 17], random forest (RF)[14, 18], gradient boosting[19, 20], and neural networks[21-23], or combinations of the above[24, 25] have been used for the purpose with preliminary successes. Before feeding the data to the ML pipeline, these methods need feature extraction. Some of the works only need sequence-based data and others might require structure-based data[26, 27]. Typically, methods using the 3D structures outperform the sequence-based methods[28]. Yet many prior works are prone to be biased on one direction variation or be overfitted[10]for practical usages. So the problem lies in how to fully exploit the limited experimental data and capture informative features.

End-to-end deep learning frameworks appeared to be a promising solution, which can enable useful features learning for various symbolic data. It can learn input features in the training process instead of fixing

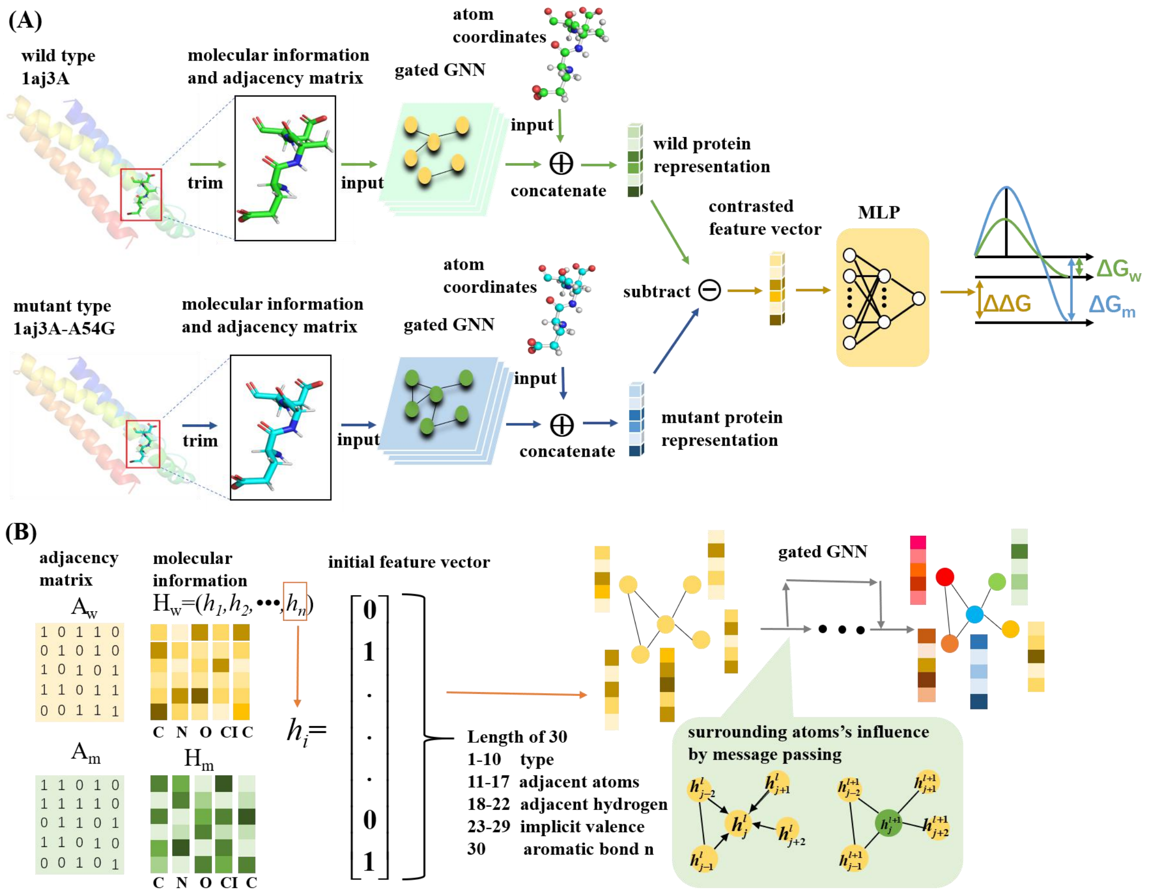


Fig. 1. (A) The architecture of ProS-GNN. (B) Illustration of the input molecular features, adjacency matrix, and message passing in the gated GNN.

them and obtain data-driven features by directly utilizing the training dataset[29]. 3D CNN has been used to predict protein stability change with high performance[30]. It treated protein structures as if they were 3D images with voxels parameterized using atom biophysical properties. However, 3D grid representation entails void space where no atoms reside, leading to inefficient computation[31].

From the point of view of a many-body system, proteins are graphs by nature, in which a vertex is an atom and edges are a chemical bonds[32]. The total free energy is a sum over all atomic energy contributions. The graph neural networks(GNN) offers a new viable solution to elucidate the structure-property relationship directly from protein structural data. It shows high accuracy with a relatively low computational cost due to less parameters. It can identify important atom features determining molecular properties by analyzing relations between neighboring atoms[33], since the message passing process extracts structural features and then relates them with the target properties[34]. Similar approach has been demonstrated for atomic energies prediction from a quantum-chemical view[35], which implies GNN's potential for protein related energy prediction. However, to the authors' best knowledge, protein stability change prediction using GNN remained unexplored.

In this regard, we propose a novel and agile approach to predict protein stability change upon mutation using a deep learning model. After trimming the non-mutant part of the protein, the model maps the 3D structural information and element compositions of the protein to a high-dimensional representation, and automatically captures the key factors leveraging a gated GNN. Our key conceptual advance is implementing the model to predict structure-property following the underlying biochemistry law. We then demonstrate the method's high performance by training with the S2648 and S3412 dataset, and tested on S^{sym} and Myoglobin dataset. Then we applied the method to predict the mutation effect on the drug for tuberculosis, Pyrazinamide (PZA), for drug resistance management.

2 Methods

2.1 Problem formulation

Our task is to predict the change of Gibbs free energy between mutant protein and wild protein. Our concept is derived from the many body Hamiltonian concept to embrace the principles of biochemistry, while maintaining the flexibility of a complex data-driven learning machine.

The input data is extracted from the PDB files, which contain the element composition and structural information. We formulate the protein feature vector $\boldsymbol{h}_i$ for the $\boldsymbol{i}$th atom, containing the vertices and edges information. This feature vector encodes the element information, the number of adjacent atoms, the number of adjacent hydrogen atoms, implicit valence, and aromatic bonds. Thee feature vectors are combined to form the input feature matrix $\boldsymbol{H} = [\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{n+r}]$ for a protein, and $\boldsymbol{H} \in \mathbb{R}^{(n+r)\times 30}$. $\boldsymbol{n}$ and $\boldsymbol{r}$ are the atomic numbers of mutant and non-mutant parts, respectively. The wild and mutant type's feature matrixes are denoted as $\boldsymbol{H}^w \in \mathbb{R}^{n\times 30}$ and $\boldsymbol{H}^m \in \mathbb{R}^{n\times 30}$. Similarly, the coordinate matrixes of the wild and mutant type atoms are $\boldsymbol{D}^w \in \mathbb{R}^{n\times 3}$ and $\boldsymbol{D}^m \in \mathbb{R}^{n\times 3}$, and the adjacency matrixes $\boldsymbol{A}^w \in \mathbb{R}^{n\times n}$ and $\boldsymbol{A}^m \in \mathbb{R}^{n\times n}$ denote the adjacent relationship between atoms in the two proteins. These matrices form the input features to predict $\boldsymbol{y}$, ΔΔG. So the overall problem can be elucidated simply as:

$$y = \boldsymbol{MLP}\left(\boldsymbol{GNN}\left(\boldsymbol{H}^{w}, \boldsymbol{A}^{w}, \boldsymbol{H}^{m}, \boldsymbol{A}^{m}\right), \boldsymbol{D}^{w}, \boldsymbol{D}^{m}\right) \quad (1).$$

Here ***MLP*** is short for multiple layer processing.

2.2 Non-mutant part trimming

Different from prior methods, which process the complete protein feature matrix ***H***, our model only processes the information of the mutant residue and its two adjacent ones, denoted as $\boldsymbol{H_e} \in \mathbb{R}^{n\times 30}$. The residual $\boldsymbol{H_r} \in \mathbb{R}^{r\times 30}$ is trimmed by an automated script ($\boldsymbol{H} = \left[\boldsymbol{H_e}\ \boldsymbol{H_r}\right]$).

2.3 Gated GNN

We expand the dimension of the feature vectors at the first layer $\boldsymbol{H}_e^{(1)}$ with a parameter matrix $\boldsymbol{W_e} \in \mathbb{R}^{30\times 1140}$ as $\boldsymbol{H}_e^{(1)} = \boldsymbol{H_e W_e}$ ($\boldsymbol{H}_e^{(1)} \in \mathbb{R}^{n\times 1140}$). In the GNN, the ***l***-th layer's atom feature $\boldsymbol{H}_e^{(l)}$ are processed by iterations of graph convolution to produce a set of updated atom features:

$$\boldsymbol{H}_e^{(l+1)} = \boldsymbol{Leaky\_relu}(\boldsymbol{WAH}_e^{(l)}) \quad (2)$$

where $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ is a learnable weight matrix and $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. ***Leaky_relu*** is the activation function.

This is a simplified the message passing process, where each atom gathers local information from its neighboring atoms and bonds, and then update information. Through information sharing between atoms, a global feature can be extracted based on this technique. In this way, the GNN implicitly learns the property to be predicted from the structure.

To improve the feature extraction performance, we integrated the gating mechanism into the network[31]. The gated graph layer is a linear combination of $\boldsymbol{H}_e^{(l)}$ and $\boldsymbol{H}_e^{(l+1)} \in \mathbb{R}^{n\times 1140}$:

$$\boldsymbol{H}_e^{gate} = \boldsymbol{GH}_e^{(l)} + (1 - \boldsymbol{G})\boldsymbol{H}_e^{(l+1)} \quad (3)$$

with

$$\boldsymbol{G} = \boldsymbol{f}(\boldsymbol{W}_{gate}\left[\boldsymbol{H}_e^{(l)}\ \boldsymbol{H}_e^{(l+1)}\right] + \boldsymbol{B}) \quad (4)$$

where $\boldsymbol{W_{gate}} \in \mathbb{R}^{n\times 1140}$ is a learnable matrix. $\boldsymbol{B} \in \mathbb{R}^{n\times 1140}$ is a bias matrix. ***f*** is the Sigmoid non-linear activation function. The feature matrix of the gated connection $\boldsymbol{H}_e^{gate}$, $\in \mathbb{R}^{n\times 1140}$ is then added to the first layer $\boldsymbol{H_e}^{(1)}$:

$$\boldsymbol{H}_e^{gnn} = \boldsymbol{H}_e^{(1)} + \boldsymbol{H}_e^{gate} \quad (5)$$

$\boldsymbol{H}_e^{gnn} \in \mathbb{R}^{n\times 1140}$ is the final output of GNN.

Then, we concatenate $\boldsymbol{H}_e^{gnn}$ with the coordinate matrix $\boldsymbol{D} \in \mathbb{R}^{n\times 3}$ to generate a feature matrix $\boldsymbol{H}_e^{d} = [\ \boldsymbol{H}_e^{gnn}\ \ \boldsymbol{D}] = [\ \boldsymbol{h}_1^{d}, \cdots, \boldsymbol{h}_n^{d}\ ] \in \mathbb{R}^{n\times 1143}$. Here we assume $\boldsymbol{h}_i^{d} \in \mathbb{R}^{1143}$ represents the energy contribution from the ***i***th atomic vector, so the sum of them corresponds to the total molecular energy:

$$z_{out} = \sum_{1}^{n} \boldsymbol{h}_i^{d} \quad (6)$$

Finally, the feature vectors of the mutant type is subtracted from the wild type to get a contrasted feature vector. This feature vector is used for ΔΔG prediction after MLP. The process mentioned above can be expressed as follows:

$$y = \boldsymbol{MLP}\left(\boldsymbol{Leaky\_relu}(\boldsymbol{W}_{out}(z_{out}^{m} - z_{out}^{w}) + \boldsymbol{b}_{out})\right) \quad (7)$$

where $\boldsymbol{W_{out}} \in \mathbb{R}^{1024\times 1024}$ is a weight matrix and $\boldsymbol{b_{out}} \in \mathbb{R}^{1204}$ is a bias vector.

2.4 Training

$$\ell = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (8)$$

where $\boldsymbol{y_i}$ and $\hat{y}_i$ stand for the predicted ΔΔG and the experimental ΔΔG for the ***i***-th sample, respectively. Given all ΔΔG labels in the training data set when mutations occur, the training goal is to minimize the mean squared error (MSE) loss.

Table 1. Comparison of different methods on the S^{sym} dataset

method	σ_{dir}	r_{dir}	σ_{rev}	r_{dir}	$r_{dir-rev}$	δ
ProS-GNN	**1.23**	**0.61**	**1.30**	**0.56**	**-0.94**	**0.04**
ThermoNet	1.56	0.47	1.55	0.47	-0.96	-0.01
POPMuSiC^{sym}	1.58	0.48	1.62	0.48	-0.77	0.03
DDGun	1.47	0.48	1.50	0.48	-0.99	-0.01
MAESTRO	1.36	0.52	2.09	0.32	-0.34	-0.58
FoldX	1.56	0.63	2.13	0.39	-0.38	-0.47
DUET	1.20	0.63	2.38	0.13	-0.21	-0.84
mCSM	1.23	0.61	2.43	0.14	-0.26	-0.91
SDM	1.74	0.51	2.28	0.32	-0.75	-0.32
I-Mutant 3.0	1.23	0.62	2.32	-0.04	0.02	-0.68
CAPSAT	1.71	0.39	2.88	0.05	-0.54	-0.72
iSTABLE	1.10	0.72	2.28	-0.08	-0.05	-0.60
NeEMO	1.08	0.72	2.35	0.02	0.09	-0.60
Rosetta	2.31	0.69	2.61	0.43	-0.41	-0.69
STRUM	1.05	0.75	2.51	-0.15	0.34	-0.87
INPS	1.42	0.51	1.44	0.50	-0.99	-0.04

3 Experiments

3.1 Dataset

To train and test our model, we use several data sets listed below.

S3421 contains 3421 experimentally determined mutations from 150 proteins.

S2648 includes 2648 single-point mutation in 131 different globular proteins.

S^{sym} contains 684 variations, and half of them are reverse variations.

Myoglobin is consisted of 134 mutations scattered throughout the protein chains. Myoglobin is a cytoplasmic globular protein that regulates cellular oxygen concentration.

3.2 Implement and evaluation

This work used a Nvidia Geforce GTX 3070 GPU for computing. We implemented the model using Pytorch and tuned the parameters by grid search. The dimension of the vertices vector is 1120 and the dimension of the full connected (FC) layers is 1024. The GNN and FC both have four layers. The model shows improved generalization, when the weight decay is 5×10^{-5} , drop-out rate is 0.5 and the batch size is 16. In addition, the model is optimized by an Adam optimizer.

The following measures are adopted to evaluate the performance of ProS-GNN. The primary measures for evaluating prediction accuracy are the Pearson correlation coefficient (***r***) and the root-mean-squared error (RMSE) (***σ***) of the experimental and predicted ΔΔGs. $\boldsymbol{r_{dir\text{-}rev}}$ and $\boldsymbol{\delta}\ _{dir\text{-}rev}$ are used to evaluate the prediction bias between the direct and reverse mutation prediction.

4 Results and Discussion

4.1 Trained using S2648 dataset

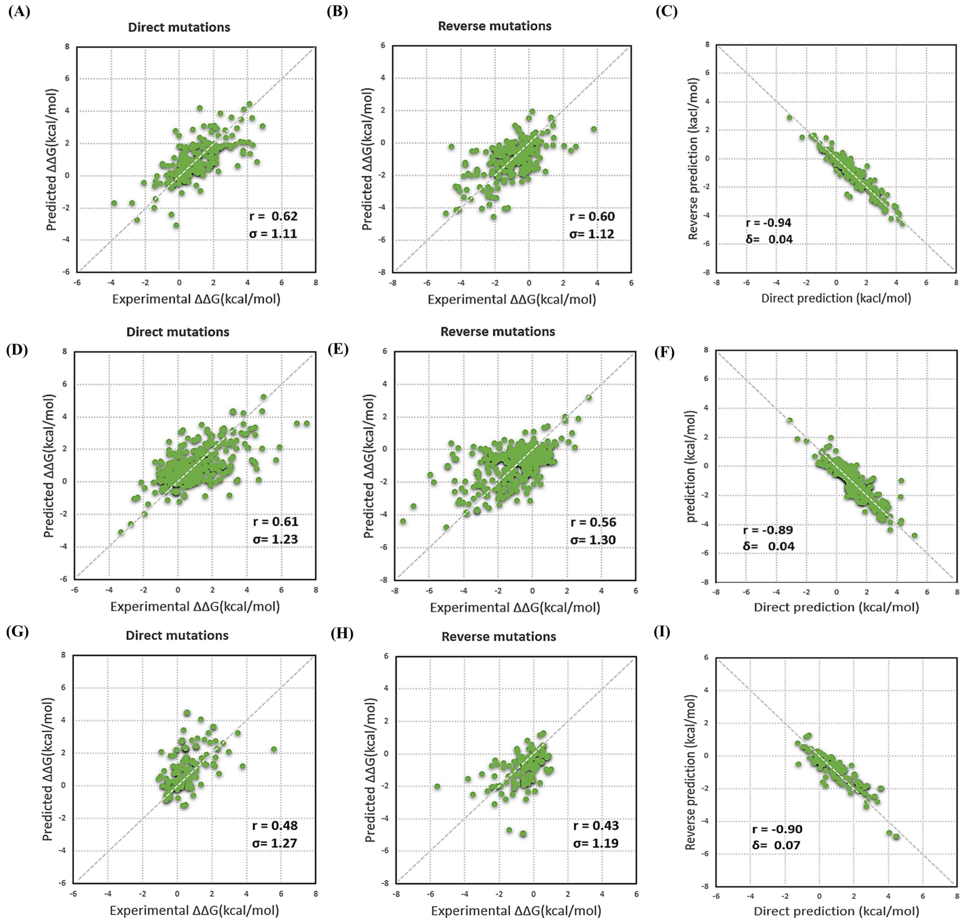


Fig. 2. ProS-GNN trained using S2648 dataset and tested on the three datasets. (A) Predicting ΔΔG for direct mutations in S2648. (B) reverse mutations in S2648. (C) Direct versus reverse ΔΔG values of all the mutations in the S2648 . (D) Predicting ΔΔG for direct mutations in S^{sym} (E) reverse mutations in S^{sym}. (F) Direct versus reverse ΔΔG values of all the mutations in the S^{sym} . (G)Predicting ΔΔG for direct mutations in Myoglobin. (H) reverse mutations in myoglobin. (I) Direct versus reverse ΔΔG values of all the mutations in the Myoglobin .

We first trained ProS-GNN using the S2648 dataset, and then tested with S2648, S^{sym}, and Myoglobin datasets, respectively. When tested with the S2648 dataset, the proposed method achieved $\boldsymbol{r}$ = 0.62， $\boldsymbol{\sigma}$ = 1.11 on direct mutations (Fig. 2 (A)), $\boldsymbol{r}$ = 0.60, $\boldsymbol{\sigma}$ = 1.12 on reverse mutations (Fig. 2 (B)), and $\boldsymbol{r}$ = -0.94, $\boldsymbol{\delta}$ = 0.04 on direct-reverse prediction(Fig. 2 (C)). The performance on the S^{sym} dataset achieved $\boldsymbol{r}$ = 0.61， $\boldsymbol{\sigma}$ = 1.23 on the direct mutations, $\boldsymbol{r}$ = 0.56， $\boldsymbol{\sigma}$ = 1.30 on the reverse mutations and $\boldsymbol{r}$ = -0.94, $\boldsymbol{\delta}$ = 0.04 on direct-reverse prediction (Fig. 2 (D)-(F)). Then, we compare our results with fifteen methods on the S^{sym} dataset and list them in Table1. It clearly showed our model outperformed other prior methods in prediction accuracy with little bias. On the Myoglobin dataset, the ProS-GNN achieved $\boldsymbol{r}$ = 0.48， $\boldsymbol{\sigma}$ = 1.27 on the direct mutations and achieved $\boldsymbol{r}$ = 0.43， $\boldsymbol{\sigma}$ = 1.19 on the reverse mutations and $\boldsymbol{r}$ = -0.90, $\boldsymbol{\delta}$ = 0.07 on direct-reverse prediction (Fig.2(G)-(I)), which potentially suggested generalization in real-life applications. These results showed that our method could effectively learn feature representations with high performance.

4.2 Trained using S3421 dataset

Similarly, we also trained ProS-GNN using S3421 dataset and tested with S3421, S^{sym}, and Myoglobin dataset, respectively. The testing results on the S3421 dataset showed $\boldsymbol{r}$ = 0.69， $\boldsymbol{\sigma}$ = 1.73 on direct mutations (Fig. 3 (A)), $\boldsymbol{r}$ = 0.71， $\boldsymbol{\sigma}$ = 1.69 on reverse mutations (Fig. 3 (B)), and $\boldsymbol{r}$ = -0.95, $\boldsymbol{\delta}$ = -0.21 on direct-reverse prediction(Fig. 3 (C)). When switched to the S^{sym} dataset, the ProS-GNN achieved $\boldsymbol{r}$ = 0.51， $\boldsymbol{\sigma}$ = 1.47 on the direct mutations, $\boldsymbol{r}$ = 0.51， $\boldsymbol{\sigma}$ = 1.43 on the reverse mutations, and $\boldsymbol{r}$ = -0.95, $\boldsymbol{\delta}$ = 0.21 on direct-reverse prediction (Fig. 3(D)-(F)). We found the performance was moderately inferior to the ones trained using S2648, which might be explained as S3214 shared no homology with S^{sym}. Last, the tested performance on Myoglobin dataset was also competitive, as it achieved $\boldsymbol{r}$ = 0.51， $\boldsymbol{\sigma}$ = 1.20 on the direct mutations prediction, $\boldsymbol{r}$ = 0.45， $\boldsymbol{\sigma}$ = 1.13 on the reverse mutations prediction and $\boldsymbol{r}$ = -0.88, $\boldsymbol{\delta}$ = -0.18 on direct-reverse prediction (Fig. 3 (G)-(I)).

4.3 Case studies: ribosomal protein S1(RpsA)

In the real case study, we used two single-point mutations, D343N and I351F, in Pyrazinamide (PZA). PZA is one of the first-line drugs, effective against latent Mycobacterium tuberculosis isolates. Resistance to this drug emerges due to mutations in pncA and rpsA genes, encoding pyrazinamidase (PZase) and ribosomal protein S1(RpsA), respectively.

We fetched the structure of RpsA (PDB ID 4NNI) from RCSB PDB, and generated the molecule structures of D343N and I351F with

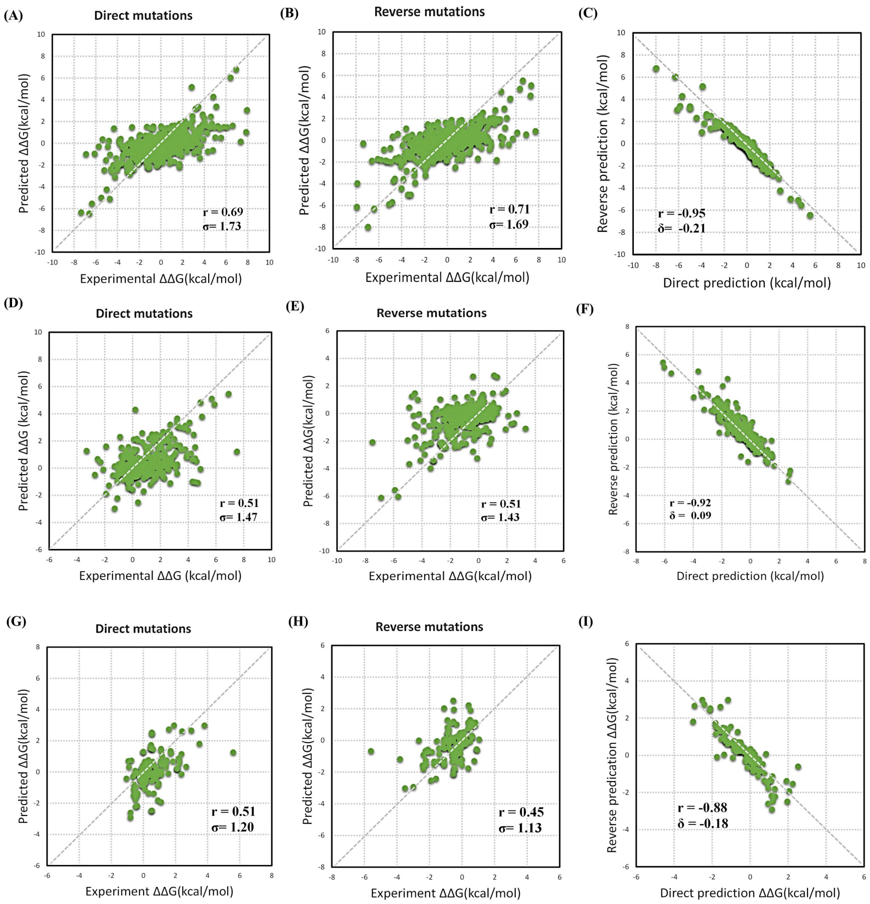


Fig. 3. ProS-GNN trained using S3421 dataset and tested on the three datasets. (A) Predicting ΔΔG for direct mutations in S3421 (B) the reverse mutations in S3421. (C) Direct versus reverse ΔΔG values of all the mutations in the S3421 . (D) Predicting ΔΔG for direct mutations in S^{sym} (E) reverse mutations in S^{sym}. (F) Direct versus reverse ΔΔG values of all the mutations in the S^{sym} . (G) Predicting ΔΔG for direct mutations in Myoglobin . (H) reverse mutations in myoglobin . (I) Direct versus reverse ΔΔG values of all the mutations in the Myoglobin .

PYMOL(Fig. 4). The literature[36] indicates the ΔΔG of 4nniA-D343N mutation and 4nniA-I351F are 3.2 kcal/mol and 2.9 kcal/mol. Our proposed method predicted 1.6 kcal/mol and 2.6 kcal/mol, respectively. While the one of the results showed discrepancies between predicted and experimental value, the differences were within 1.6 kcal/mol, which was still normal considering the method's RMSE. Therefore, it demonstrated the potential of ProS-GNN as a rapid estimator of ΔΔG upon protein mutations in a medical environment.

4.4 Time consumption study

Since we only extracted the information from the mutant residue and its two neighbors, the strategy substantially reduces the training and testing time by one or two orders of magnitude.

For example, the training time for the S2648 data set is down to 6-10 seconds per epoch after mutation part extraction(Table 2). Even if trained for 400 epoches, it only takes around one hour. Plus, the testing last for only 1 second. This highly efficient manner clearly caters the high throughput requirement in the pharmaceutical industry, where high volume data needs to be tested. It is noteworthy that removing the redundant data also boosts the overall prediction accuracy as the irrelevant information has been eliminated.

Table 2. Time consumption comparison between with and without mutant part data extraction

	data extraction	w/o data extraction
training time per epoch	6~10 seconds	900~1000 seconds
testing time	1 seconds	15~20 seconds

5 Conclusion

Here we pioneered into predicting protein stability change upon mutation with an end-to-end GNN. To exclude irrelevant features and speed up the training, we first extracted the mutant part data. Subsequently, the model leveraged a gated GNN to capture the molecular features by message passing. In addition, it incorporated the raw molecular coordinates into the framework to predict ΔΔG. Rigorous experimental evaluations show that our model performed highly competitively on the S2648, S3214, S^{sym}, and Myoglobin datasets. Furthermore, the method led to reasonable estimation for clinical drug resistance prediction. The substantial success over the task suggests a new strategy for swift protein stability change prediction and enlightens future GNN based method for improvement.

Funding

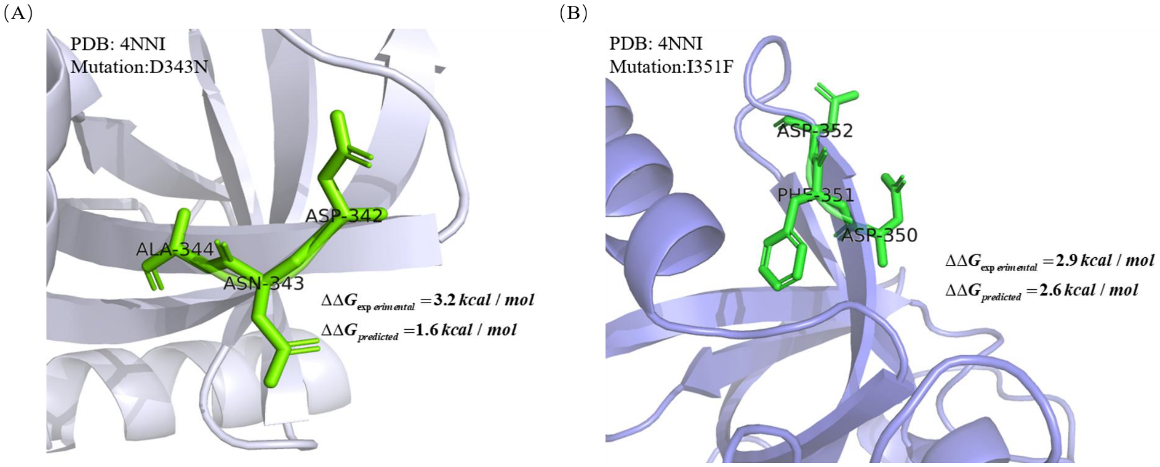

(A)
PDB: 4NNI
Mutation:D343N
ASP-342
ALA-344
ASN-343
$\Delta\Delta G_{exp\,erimental} = \mathbf{3.2}\ kcal\ /\ mol$
$\Delta\Delta G_{predicted} = \mathbf{1.6}\ kcal\ /\ mol$
(B)
PDB: 4NNI
Mutation:I351F
ASP-352
PHE-351
ASP-350
$\Delta\Delta G_{exp\,erimental} = \mathbf{2.9}\ kcal\ /\ mol$
$\Delta\Delta G_{predicted} = \mathbf{2.6}\ kcal\ /\ mol$

Fig. 4. RpsA protein's Gibbs free energy change prediction upon mutation and their residue local environment. (A) D343N mutation (B) I351F mutation

This work has been supported by the Natural Science Foundation of China(No.62104034), the Fundamental Research Fund from Central University(No. 2023012) and Natural Science Foundation of Hebei Province (No. F2020501033).

Conflict of Interest: We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript.

References

[1].Casadio, R., et al., Correlating disease-related mutations to their effect on protein stability: A large-scale analysis of the human proteome. Human Mutation, 2011. 32(10): p. 1161-1170.

[2].Hartl, F.U., Protein Misfolding Diseases. Annual Review of Biochemistry, 2017. 86(1): p. 21-26.

[3].Stefl, S., et al., Molecular Mechanisms of Disease-Causing Missense Mutations. Journal of Molecular Biology, 2013. 425(21): p. 3919-3936.

[4].Li, M., et al., Balancing Protein Stability and Activity in Cancer: A New Approach for Identifying Driver Mutations Affecting CBL Ubiquitin Ligase Activation. Cancer Research, 2016. 76(3): p. 561-571.

[5].Guerois, R., J.E. Nielsen and L. Serrano, Predicting Changes in the Stability of Proteins and Protein Complexes: A Study of More Than 1000 Mutations. Journal of Molecular Biology, 2002. 320(2): p. 369-387.

[6].Kollman, P.A., et al., Calculating Structures and Free Energies of Complex Molecules: Combining Molecular Mechanics and Continuum Models. Accounts of Chemical Research, 2000. 33(12): p. 889-897.

[7].Getov, I., M. Petukh and E. Alexov, SAAFEC: Predicting the Effect of Single Point Mutations on Protein Folding Free Energy Using a Knowledge-Modified MM/PBSA Approach. International Journal of Molecular Sciences, 2016. 17(4): p. 512.

[8].Petukh, M., M. Li and E. Alexov, Predicting Binding Free Energy Change Caused by Point Mutations with Knowledge-Modified MM/PBSA Method. PLOS Computational Biology, 2015. 11(7): p. e1004276.

[9].Sanavia, T., et al., Limitations and challenges in protein stability prediction upon genome variations: towards future applications in precision medicine. Computational and Structural Biotechnology Journal, 2020. 18: p. 1968-1979.

[10].Fang, J., A critical review of five machine learning-based algorithms for predicting protein stability changes upon mutation. Briefings in Bioinformatics, 2020. 21(4): p. 1285-1292.

[11].Montanucci, L., et al., DDGun: an untrained method for the prediction of protein stability changes upon single and multiple point variations. BMC Bioinformatics, 2019. 20(S14): p. 335-335.

[12].Capriotti, E., P. Fariselli and R. Casadio, I-Mutant2.0: predicting stability changes upon mutation from the protein sequence or structure. nucleic acids research, 2005. 33: p. 306-310.

[13].Cheng, J., A. Randall and P. Baldi, Prediction of protein stability changes for single-site mutations using support vector machines. proteins, 2005. 62(4): p. 1125-1132.

[14].Pires, D.E.V., D.B. Ascher and T.L. Blundell, DUET: A Server for Predicting Effects of Mutations on Protein Stability Using an Integrated Computational Approach. nucleic acids research, 2014. 42: p. 314-319.

[15].Fariselli, P., et al., INPS: predicting the impact of non-synonymous variations on protein stability from sequence. Bioinformatics, 2015. 31(17): p. 2816-2821.

[16].Huang, L., M.M. Gromiha and S. Ho, iPTREE-STAB. bioinformatics, 2007. 23(10): p. 1292-1293.

[17].Witvliet, D.K., et al., ELASPIC web-server: proteome-wide structure-based prediction of mutation effects on protein stability and binding affinity. Bioinformatics, 2016. 32(10): p. 1589-1591.

[18].Wainreb, G., et al., Protein stability: A single recorded mutation aids in predicting the effects of other mutations in the same amino acid site. Bioinformatics (Oxford, England), 2011. 27: p. 3286-92.

[19].Quan, L., Q. Lv and Y. Zhang, STRUM: structure-based prediction of protein stability changes upon single-point mutation. Bioinformatics, 2016. 32(19): p. 2936-2946.

[20].Yang, Y., et al., ProTstab–predictor for cellular protein stability. BMC Genomics, 2019. 20(1): p. 1-9.

[21].Dehouck, Y., et al., PoPMuSiC 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality. BMC Bioinformatics, 2011. 12(1): p. 151.

[22].Capriotti, E., P. Fariselli and R. Casadio, A neural-network-based method for predicting protein stability changes upon single point mutations, in Intelligent Systems in Molecular Biology. 2004. p. 63-68.

[23].Giollo, M., et al., NeEMO: a method using residue interaction networks to improve prediction of protein stability upon mutation. BMC Genomics, 2014. 15(S4): p. 1-11.

[24].Tian, J., et al., Predicting changes in protein thermostability brought about by single- or multi-site mutations. BMC Bioinformatics, 2010. 11(1): p. 370-370.

[25].Laimer, J., et al., MAESTRO - multi agent stability prediction upon point mutations. BMC Bioinformatics, 2015. 16(1): p. 116-116.

[26].Chen, C.W., J. Lin and Y.W. Chu, iStable: off-the-shelf predictor integration for predicting protein stability changes. BMC Bioinformatics, 2013. 14 Suppl 2(2): p. S5.

[27].Worth, C.L., R. Preissner and T.L. Blundell, SDM—a server for predicting effects of mutations on protein stability and malfunction. nucleic acids research, 2011. 39: p. 215-222.

[28].Parthiban, V., M.M. Gromiha and D. Schomburg, CUPSAT: prediction of protein stability upon point mutations. nucleic acids research, 2006. 34: p. 239-242.

[29].Cao, H., et al., DeepDDG: Predicting the Stability Change of Protein Point Mutations Using Neural Networks. Journal of Chemical Information and Modeling, 2019. 59(4): p. 1508-1514.

[30].Li, B., et al., Predicting changes in protein thermodynamic stability upon point mutation with deep 3D convolutional neural networks. PLOS Computational Biology, 2020. 16(11): p. e1008291.

[31].Lim, J., et al., Predicting Drug–Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation. Journal of Chemical Information and Modeling, 2019. 59(9): p. 3981-3988.

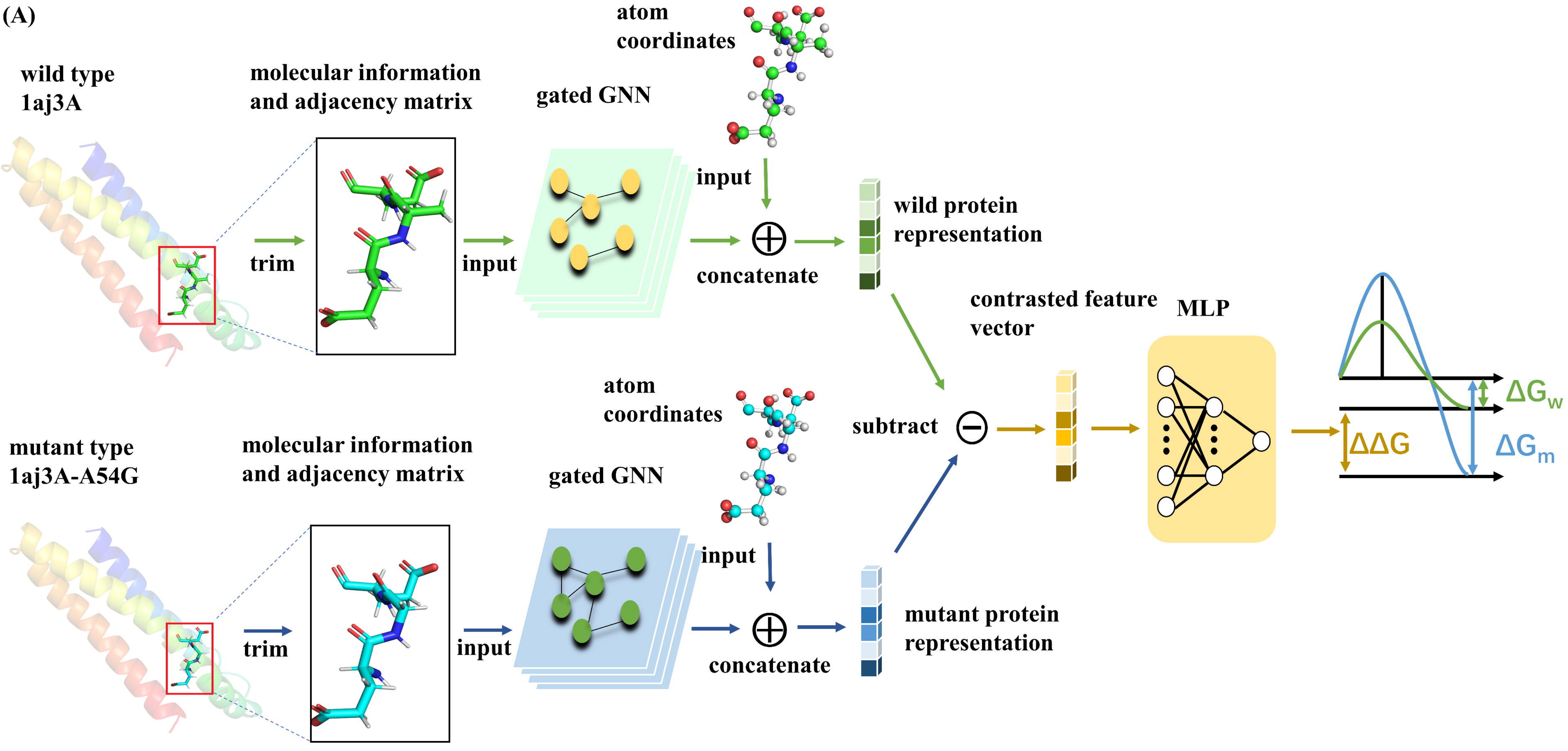
[32].Tsubaki, M., K. Tomii and J. Sese, Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. Bioinformatics, 2019. 35(2): p. 309-318.

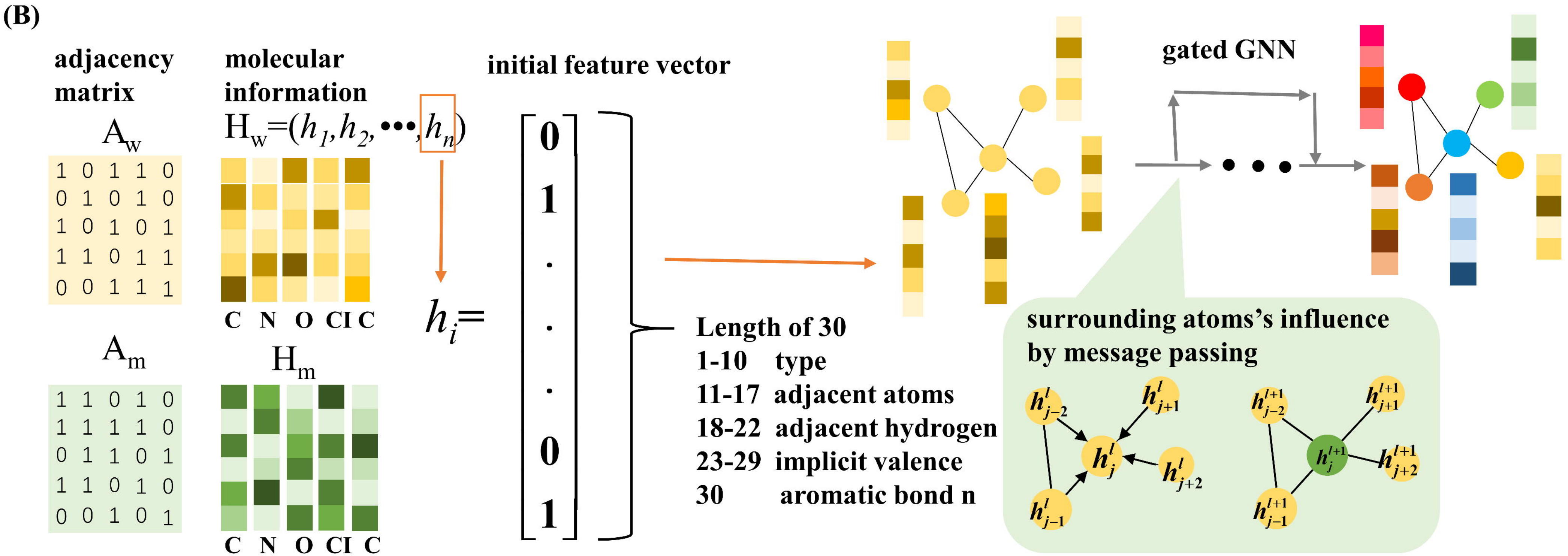
[33].Ryu, S., J. Lim and W.Y. Kim, Deeply learning molecular structure-property relationships using graph attention neural network.. 2018.

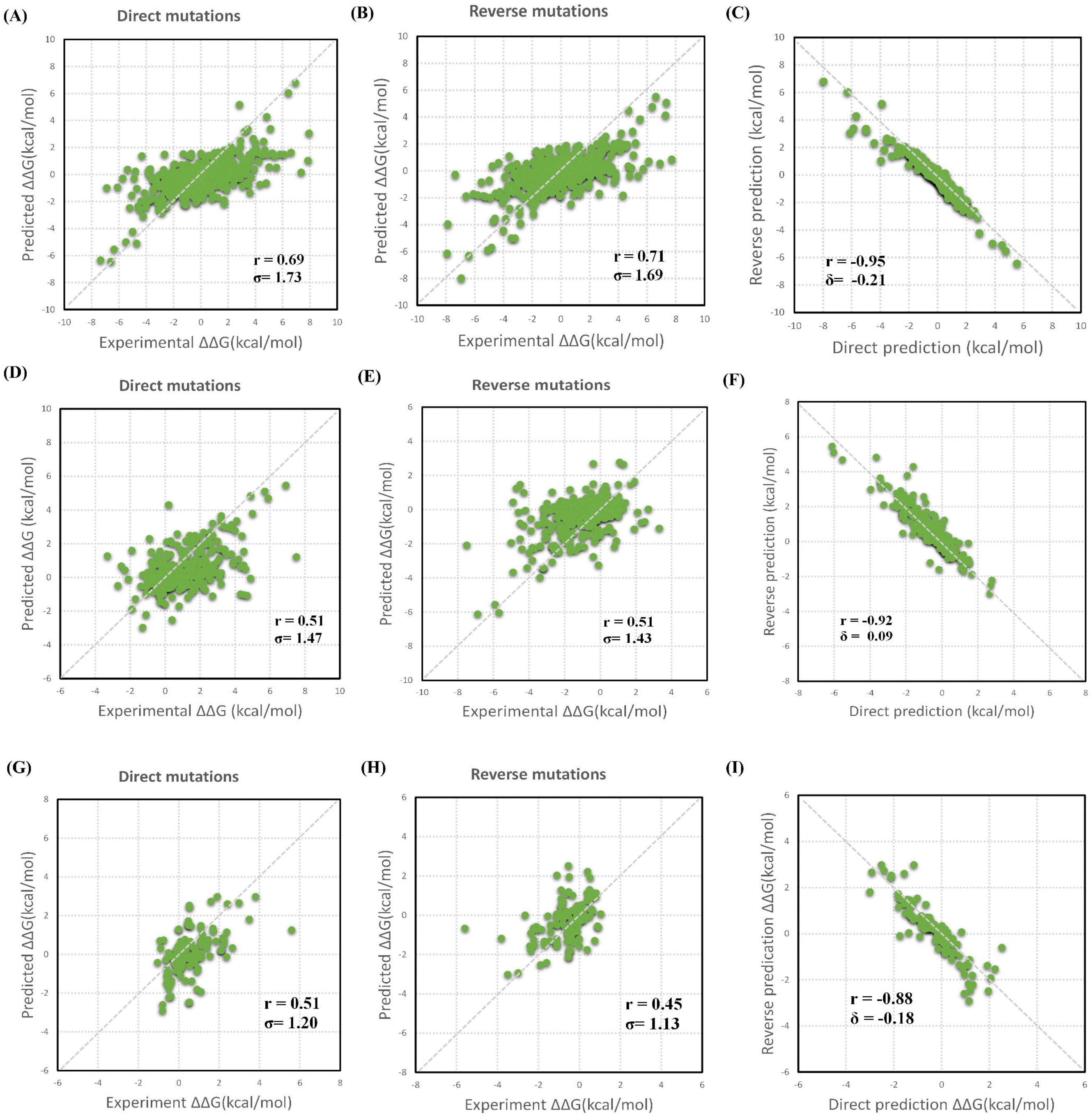
[34].Wieder, O., et al., A compact review of molecular property prediction with graph neural networks. Drug discovery today. Technologies, 2020.

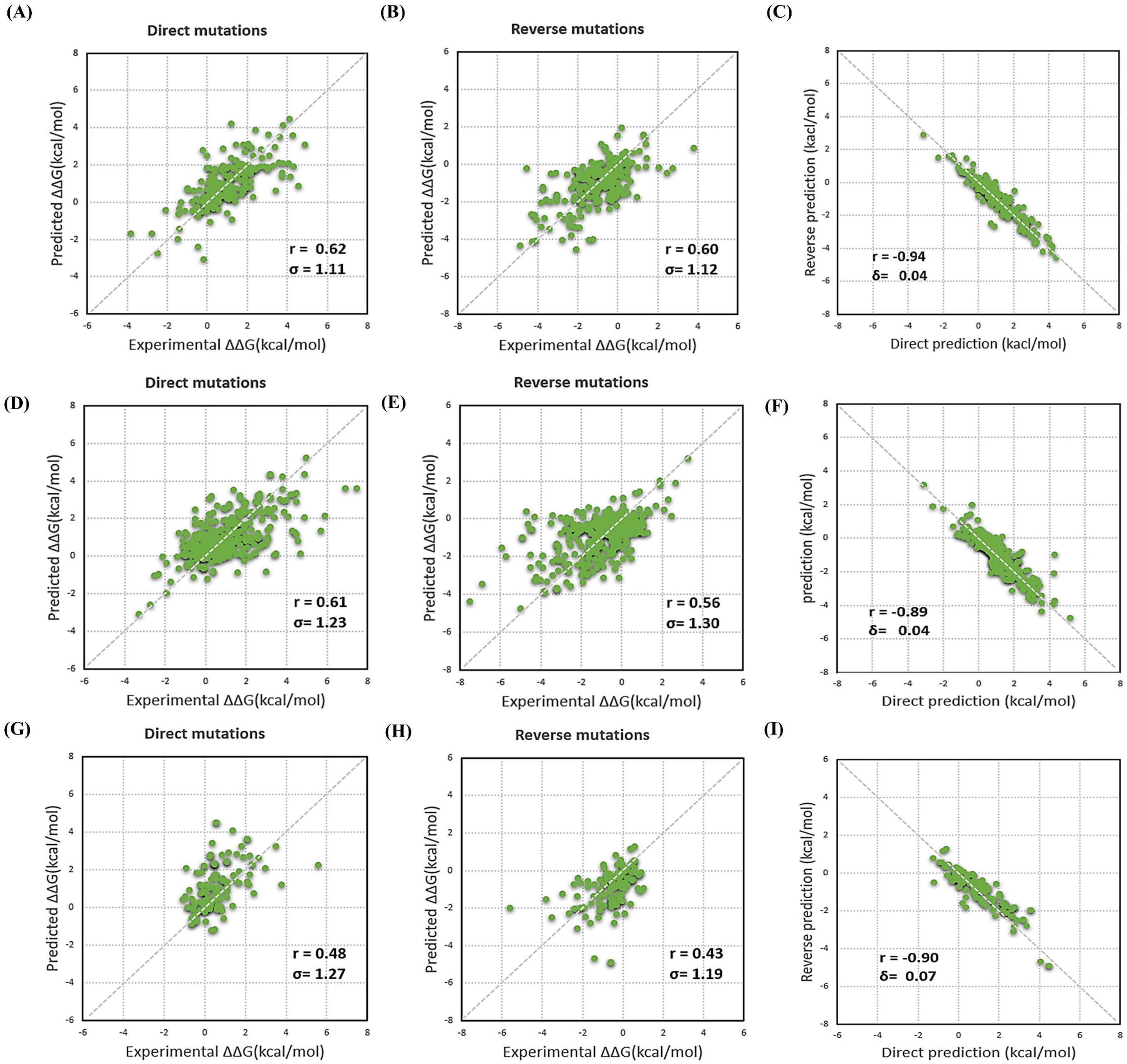
[35].Schütt, K.T., et al., Quantum-chemical insights from deep tensor neural networks. Nature Communications, 2017. 8(1): p. 13890-13890

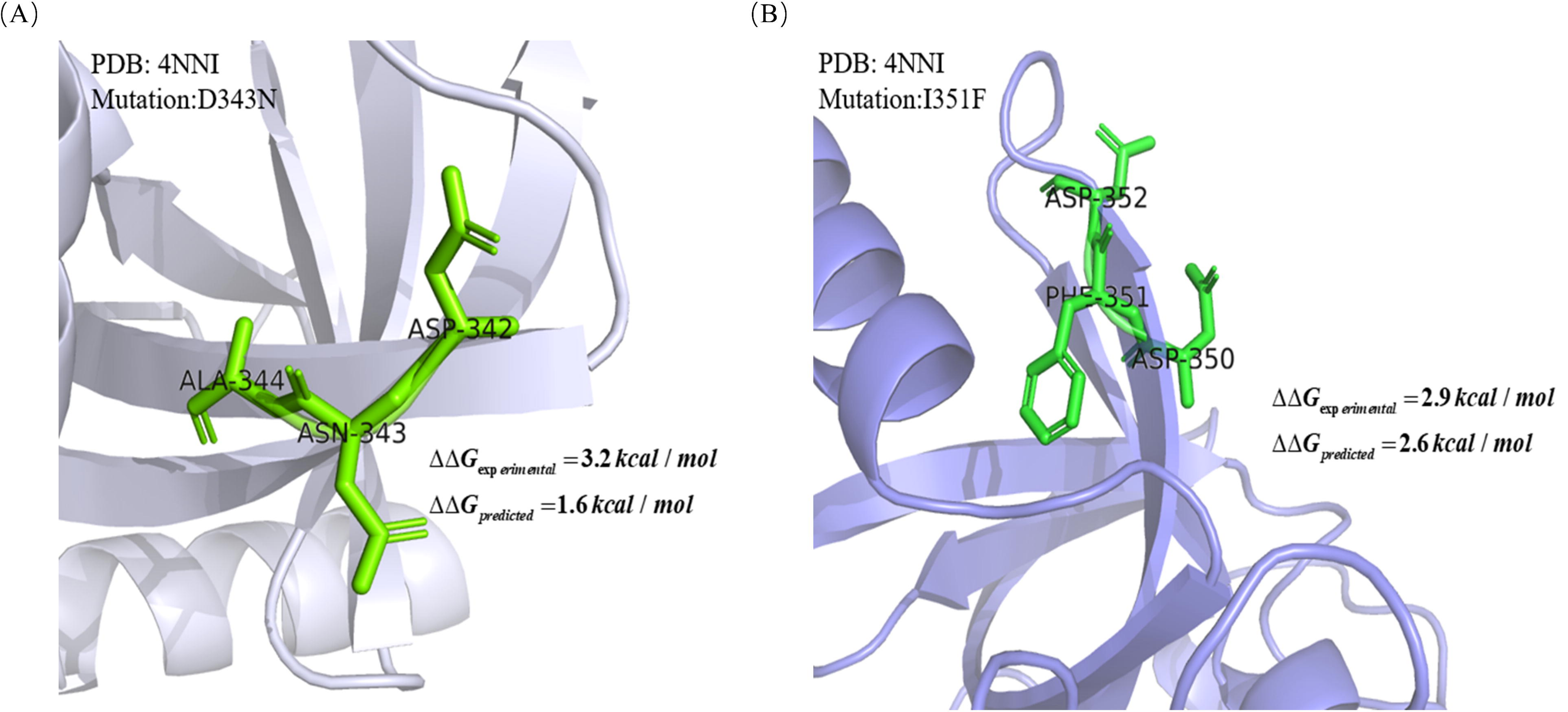
[36].Khan, M.T., et al., Structural and free energy landscape of novel mutations in ribosomal protein S1 (rpsA) associated with pyrazinamide resistance. Scientific Reports, 2019. 9(1): p. 7482.


(A)
wild type
1aj3A
trim
molecular information
and adjacency matrix
input
gated GNN
atom
coordinates
input
concatenate
wild protein
representation
mutant type
1aj3A-A54G
trim
molecular information
and adjacency matrix
input
gated GNN
atom
coordinates
input
concatenate
mutant protein
representation
subtract
contrasted feature
vector
MLP
ΔG$_w$
ΔΔG
ΔG$_m$


(B)
adjacency matrix
A_w
1 0 1 1 0
0 1 0 1 0
1 0 1 0 1
1 1 0 1 1
0 0 1 1 1
A_m
1 1 0 1 0
1 1 1 1 0
0 1 1 0 1
1 1 0 1 0
0 0 1 0 1
molecular information
$H_w=(h_1,h_2,\bullet\bullet\bullet,h_n)$
C N O Cl C
H_m
C N O Cl C
initial feature vector
$h_i=\begin{bmatrix}0\\1\\ \cdot\\ \cdot\\ \cdot\\0\\1\end{bmatrix}$
Length of 30
1-10 type
11-17 adjacent atoms
18-22 adjacent hydrogen
23-29 implicit valence
30 aromatic bond n
gated GNN
surrounding atoms's influence by message passing
h^l_{j-2}
h^l_{j+1}
h^l_j
h^l_{j+2}
h^l_{j-1}
h^{l+1}_{j-2}
h^{l+1}_{j+1}
h^{l+1}_j
h^{l+1}_{j+2}
h^{l+1}_{j-1}


(A)
Direct mutations
Predicted ΔΔG(kcal/mol)
Experimental ΔΔG(kcal/mol)
r = 0.69
σ= 1.73
(B)
Reverse mutations
Predicted ΔΔG(kcal/mol)
Experimental ΔΔG(kcal/mol)
r = 0.71
σ= 1.69
(C)
Reverse prediction (kcal/mol)
Direct prediction (kcal/mol)
r = -0.95
δ= -0.21
(D)
Direct mutations
Predicted ΔΔG (kcal/mol)
Experimental ΔΔG (kcal/mol)
r = 0.51
σ= 1.47
(E)
Reverse mutations
Predicted ΔΔG(kcal/mol)
Experimental ΔΔG(kcal/mol)
r = 0.51
σ= 1.43
(F)
Reverse prediction (kcal/mol)
Direct prediction (kcal/mol)
r = -0.92
δ = 0.09
(G)
Direct mutations
Predicted ΔΔG(kcal/mol)
Experiment ΔΔG(kcal/mol)
r = 0.51
σ= 1.20
(H)
Reverse mutations
Predicted ΔΔG(kcal/mol)
Experiment ΔΔG(kcal/mol)
r = 0.45
σ= 1.13
(I)
Reverse predication ΔΔG(kcal/mol)
Direct prediction ΔΔG(kcal/mol)
r = -0.88
δ = -0.18


(A)
Direct mutations
Predicted ΔΔG(kcal/mol)
Experimental ΔΔG(kcal/mol)
r = 0.62
σ = 1.11
(B)
Reverse mutations
Predicted ΔΔG(kcal/mol)
Experimental ΔΔG(kcal/mol)
r = 0.60
σ= 1.12
(C)
Reverse prediction (kacl/mol)
Direct prediction (kacl/mol)
r = -0.94
δ= 0.04
(D)
Direct mutations
Predicted ΔΔG(kcal/mol)
Experimental ΔΔG(kcal/mol)
r = 0.61
σ= 1.23
(E)
Reverse mutations
Predicted ΔΔG(kcal/mol)
Experimental ΔΔG(kcal/mol)
r = 0.56
σ= 1.30
(F)
prediction (kcal/mol)
Direct prediction (kcal/mol)
r = -0.89
δ= 0.04
(G)
Direct mutations
Predicted ΔΔG(kcal/mol)
Experimental ΔΔG(kcal/mol)
r = 0.48
σ= 1.27
(H)
Reverse mutations
Predicted ΔΔG(kcal/mol)
Experimental ΔΔG(kcal/mol)
r = 0.43
σ= 1.19
(I)
Reverse prediction (kcal/mol)
Direct prediction (kcal/mol)
r = -0.90
δ= 0.07


(A)
PDB: 4NNI
Mutation:D343N
ASP-342
ALA-344
ASN-343
$\Delta\Delta G_{\exp erimental} = 3.2\, kcal / mol$
$\Delta\Delta G_{predicted} = 1.6\, kcal / mol$
(B)
PDB: 4NNI
Mutation:I351F
ASP-352
PHE-351
ASP-350
$\Delta\Delta G_{\exp erimental} = 2.9\, kcal / mol$
$\Delta\Delta G_{predicted} = 2.6\, kcal / mol$