

Sequence analysis

DDGemb: predicting protein stability change upon single- and multi-point variations with embeddings and deep learning

Castrense Savojardo ^{1,‡,*}, Matteo Manfredi ^{1,‡}, Pier Luigi Martelli ^{1,*}, Rita Casadio ^{1,2}

¹Biocomputing Group, Department of Pharmacy and Biotechnology, University of Bologna, Via San Giacomo 9/2, Bologna, 40126, Italy

²The Alma Climate Institute, Interdepartmental Center, University of Bologna, Bologna, 40100, Italy

*Corresponding authors. Castrense Savojardo, Biocomputing Group, Department of Pharmacy and Biotechnology, University of Bologna, Via San Giacomo 9/2, Bologna, 40126, Italy. E-mail: castrense.savojardo2@unibo.it; Pier Luigi Martelli, Biocomputing Group, Department of Pharmacy and Biotechnology, University of Bologna, Via San Giacomo 9/2, Bologna, 40126, Italy. E-mail: pierluigi.martelli@unibo.it.

[‡]Equal contribution.

Associate Editor: Jianlin Cheng

Abstract

Motivation: The knowledge of protein stability upon residue variation is an important step for functional protein design and for understanding how protein variants can promote disease onset. Computational methods are important to complement experimental approaches and allow a fast screening of large datasets of variations.

Results: In this work, we present DDGemb, a novel method combining protein language model embeddings and transformer architectures to predict protein $\Delta\Delta G$ upon both single- and multi-point variations. DDGemb has been trained on a high-quality dataset derived from literature and tested on available benchmark datasets of single- and multi-point variations. DDGemb performs at the state of the art in both single- and multi-point variations.

Availability and implementation: DDGemb is available as web server at <https://ddgemb.biocomp.unibo.it>. Datasets used in this study are available at <https://ddgemb.biocomp.unibo.it/datasets>.

1 Introduction

Computational methods for predicting the effect of variations on protein thermodynamic stability play a fundamental role in computational protein design (Notin *et al.* 2024), in functional characterization of protein variants (Vihinen 2021) and their relation to disease onset (Puglisi 2022, Pandey *et al.* 2023). In the last years, several methods have been presented for the prediction of protein stability change upon variation ($\Delta\Delta G$).

Tools available can be roughly classified according to the type of information they rely on (protein structure and/or sequence) and on the type of method which carries out the prediction. Structure-based methods rely on the availability of the protein structure as an input. Different structure-based predictive approaches have been presented, including methods based on force fields and energy functions (Schymkowitz *et al.* 2005, Kellogg *et al.* 2011, Worth *et al.* 2011), conventional machine-learning methods (Capriotti *et al.* 2005, Dehouck *et al.* 2011, Pires *et al.* 2014b, Laimer *et al.* 2015, Savojardo *et al.* 2016, Montanucci *et al.* 2019, Chen *et al.* 2020), deep-learning approaches (Li *et al.* 2020, Benevenuto *et al.* 2021), and consensus methods (Pires *et al.* 2014a, Rodrigues *et al.* 2018, 2021).

Sequence-based methods only use features that can be extracted from the protein sequence. So far, the vast majority

of methods available are based on canonical features such as evolutionary information and physicochemical properties, processed by conventional machine-learning methods (Capriotti *et al.* 2005, Cheng *et al.* 2006, Fariselli *et al.* 2015, Montanucci *et al.* 2019, Li *et al.* 2021). ACDC-NN-Seq introduced deep-learning methods (convolutional networks) to process sequence profiles extracted from multiple-sequence alignments (Pancotti *et al.* 2021). Recently, PROSTATA (Umerenkov *et al.* 2023) adopted protein language models for encoding the protein wild-type and mutated sequences. The protein language model input is then processed in PROSTATA using a simple neural network with a single hidden layer. The sequence-based THPLM adopts pretrained protein language models and a simple convolutional neural network (Gong *et al.* 2023). Finally, ThermoMPNN (Dieckhaus *et al.* 2024) also adopts a pretrained pLM called ProteinMPNN (Dauparas *et al.* 2022) in combination with a deep network to predict $\Delta\Delta G$ upon single-point variations.

One of the major challenges in the field of protein stability prediction is the ability to predict $\Delta\Delta G$ upon multi-point variations, i.e. how protein stability is affected when variations occur at multiple residue positions. So far, only a few methods support multi-point variations as an input: four structure-based methods [FoldX (Schymkowitz *et al.* 2005), MAESTRO (Laimer *et al.* 2015, 2016), DDGun3D

(Montanucci *et al.* 2019) and Dynamut2 (Rodrigues *et al.* 2021)], and one sequence-based method, DDGunSeq (Montanucci *et al.* 2019). Overall, the performance of methods for predicting $\Delta\Delta G$ upon multi-point variations is generally lower than that obtained for single-point variations.

In this work, we present a novel method called DDGemb for the prediction of protein $\Delta\Delta G$ upon both single- and multi-point variations. DDGemb exploits the power of ESM2 protein language model (Lin *et al.* 2023) for protein and variant representation in combination with a deep-learning architecture based on a Transformer encoder (Vaswani *et al.* 2017) to predict the $\Delta\Delta G$.

We train DDGemb using full-length protein sequences and single-point variations from the S2648 dataset (Dehouck *et al.* 2011), previously adopted to train different state-of-the-art approaches (Dehouck *et al.* 2011, Fariselli *et al.* 2015, Savojardo *et al.* 2016).

The performance of DDGemb is evaluated on $\Delta\Delta G$ prediction upon both single- and multi-point variations. For single-point variations, we adopted the S669 dataset recently presented in literature and already adopted for benchmarking a large set of tools (Pancotti *et al.* 2022). For multi-point variations, we adopted a dataset derived from the PTmul dataset (Montanucci *et al.* 2019). In both benchmarks, DDGemb reports state-of-the-art performance, overpassing both sequence- and structure-based methods.

2 Materials and methods

2.1 Datasets

2.1.1 The S669 blind test set

For a fair and comprehensive evaluation of DDGemb performance and for comparing with other state-of-the-art approaches, we take advantage of an independent dataset adopted in literature to score a large set of available tools for predicting protein stability change upon variation (Pancotti *et al.* 2022).

The dataset, named S669, comprises 1338 direct and reverse single-site variations occurring in 95 protein chains. $\Delta\Delta G$ values were retrieved from ThermoMutDB (Xavier *et al.* 2021) and manually checked by authors. In this paper, we adopt the convention by which negative $\Delta\Delta G$ values indicate destabilizing variations. Interestingly, the dataset has been built to be nonredundant at 25% sequence identity with respect to datasets routinely used for training tools available in literature, including the S2648 (Dehouck *et al.* 2011) and the VariBench dataset (Nair and Vihinen 2013). This enables a fair comparison with most state-of-the-art tools. Variations included in S669 are provided in relation to PDB chains. In this work, since DDGemb adopts protein language models for input encoding, we mapped all variations on full-length UniProt (<https://www.uniprot.org/>) sequences using SIFTS (Dana *et al.* 2019).

2.1.2 Training set: the S2450 dataset

To build our training set, we started from the well-known and widely adopted S2648 dataset (Dehouck *et al.* 2011), containing 2648 single-point variations on 131 different proteins. Associated experimental $\Delta\Delta G$ values are retrieved from the ProTherm database (Bava *et al.* 2004) and were manually checked and corrected to avoid inconsistencies. Differently from previous works adopting the same dataset (Dehouck *et al.* 2011, Fariselli *et al.* 2015), in which variations are

directly mapped on PDB chain sequences, in this work, we adopted full-length protein sequences from UniProt. To this aim, we used SIFTS (Dana *et al.* 2019) to map PDB chains and variant positions on corresponding UniProt sequences.

Homology reduction of the S669 dataset against S2648 was originally performed in (Pancotti *et al.* 2022) considering only PDB-covered portions of the sequences. This procedure does not guarantee to detect all sequence similarity on full-length sequences. For this reason, in this work, we compared UniProt sequences in S669 and S2648, removing from the training set those having >25% sequence identity with any sequence in the test set (S669). Overall, 18 sequences were removed from S2648, accounting for 198 single-point variations. This reduced dataset is then referred to as S2450 throughout the entire paper.

The S2450 dataset was adopted here to perform 5-fold cross-validation. To this aim, we implemented the stringent data split procedure described in (Fariselli *et al.* 2015), by which all variations occurring on the same protein are put in the same cross-validation subset and proteins are divided among subsets taking into consideration pairwise sequence identity (setting a threshold to 25%). In this way, during cross-validation, no redundancy is present between protein sequences included in training and validation sets.

By construction, the S2450 dataset is unbalanced toward destabilizing variations (i.e. negative $\Delta\Delta G$ values). To balance the dataset, to reduce the bias toward destabilizing $\Delta\Delta G$ values, and to improve the model capability of predicting stabilizing variations, we exploited thermodynamic reversibility of variations, by which $\Delta\Delta G(A \rightarrow B) = -\Delta\Delta G(B \rightarrow A)$ (Capriotti *et al.* 2008). Using the reversibility property, the set of variations can be artificially doubled to include reverse variations, switching the sign of experimental $\Delta\Delta G$ values.

2.1.3 Multiple variations: the reduced PTmul dataset

We also adopted a dataset for testing the DDGemb on the prediction of $\Delta\Delta G$ upon multi-point variations. The dataset, referred to as PTmul, has been introduced in (Montanucci *et al.* 2019): it comprises 914 multi-point variations on 91 proteins. However, the original PTmul dataset share a high level of sequence similarity when compared to our S2450 training dataset. In order to perform a fair evaluation of the performance, we excluded from PTmul all proteins that are similar to any protein in our S2450 training set. After this reduction step, we retained 82 multi-point variants occurring in 14 proteins. Although the number of variants is significantly reduced if compared to the original dataset, using this homology reduction procedure ensures a fair evaluation of the different methods. The reduced dataset is referred to as PTmul-NR.

2.2 The DDGemb method

An overview of the DDGemb deep-learning model is shown in Fig. 1. The architecture comprises two components: (i) the input encoding and (ii) the $\Delta\Delta G$ prediction model. In the next section we describe in detail both components.

2.2.1 Input encoding

For encoding a single residue variation, we start from the wild-type and the variant protein sequences. The latter is derived from the former upon either single-point or multi-point variations. In the first step, the two sequences, both of length L , are encoded using the ESM2 protein language model

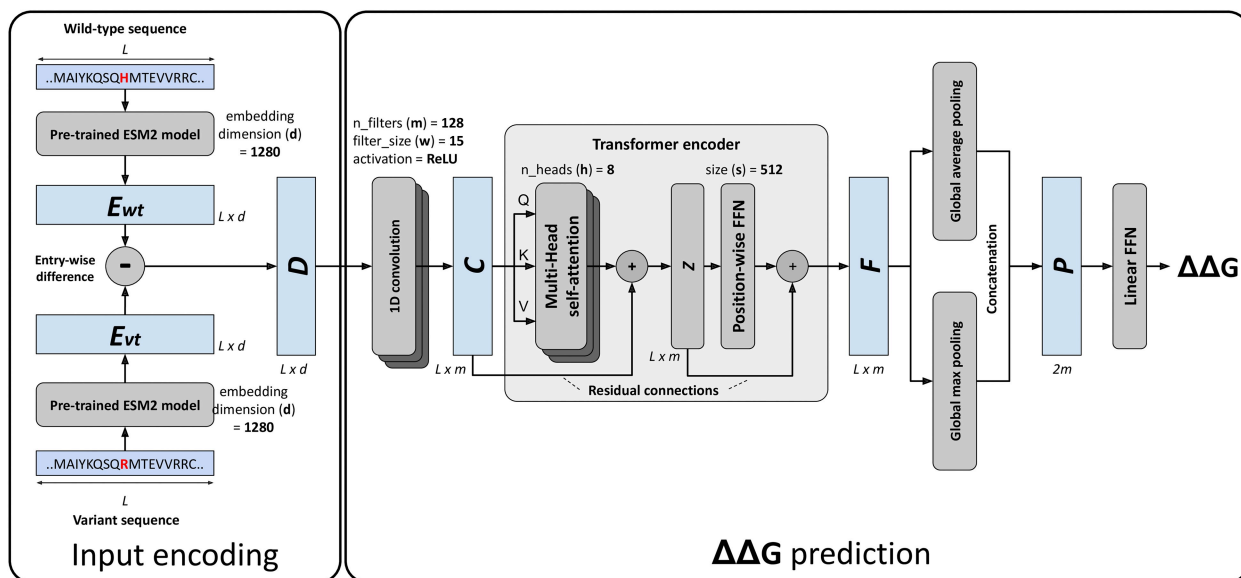


Figure 1. The DDGEmb model architecture. For input encoding details refer to Section 2.2.1; for $\Delta\Delta G$ prediction see Sections 2.2.2 and 2.2.3.

(pLM) (Rives *et al.* 2021, Lin *et al.* 2023). Among the different models available and after input encoding optimization (see Section 3), here we adopted the medium-size 33-layers model with 650M parameters and trained on the UniRef50 database. This model provides residue-level embeddings of dimension 1280 and represents a good trade-off between representation expressivity and computational requirements.

For generating embeddings, we adopted the ESM2 package available at <https://github.com/facebookresearch/esm>.

The application of the ESM2 pLM provides two $L \times d$ matrices, named E_{wt} and E_{vt} , representing the residue-level embeddings of the wild-type and variant sequences (derived either from a single- or multi-point variation), respectively. A single $L \times d$ matrix D encoding the variation is then generated computing the element-wise difference of E_{wt} and E_{vt} :

$$D = E_{wt} - E_{vt} \quad (1)$$

The matrix D is used as input for the downstream $\Delta\Delta G$ prediction architecture.

2.2.2 The Transformer based $\Delta\Delta G$ prediction network

The remaining part of the DDGEmb architecture is devised to predict a $\Delta\Delta G$ value starting from the input matrix D encoding the protein variant. The hyperparameters of the final model were optimized in cross-validation, according to the different configurations reported in Table 1 (see Section 3). After optimization, the final selected model is Model4 (Table 1), described in the following.

The input matrix D is firstly processed by a 1D convolution layer comprising $m = 128$ filters of size $w = 15$, with ReLU activation functions. The 1D-convolution layer provides a way of projecting the higher-dimensional input data into a lower-dimensional space of size m , extracting local contextual information through a series of sliding filters of width w . The output of the 1D-convolution is a matrix C of dimension $L \times m$.

The matrix C is then passed through a Transformer encoder layer (Vaswani *et al.* 2017), consisting of a cascading architecture including a multi-head attention layer with eight attention heads (h), residual connections, and a position-wise

Table 1. Five-fold cross-validation results of different $\Delta\Delta G$ prediction architectures.

Model	Model configuration	\overline{PCC}	\overline{RMSE}	\overline{MAE}
Model0	$m = 32, h = 2, s = 128$	0.68 ± 0.01	1.27 ± 0.10	0.97 ± 0.09
Model1	$m = 64, h = 2, s = 256$	0.70 ± 0.01	1.23 ± 0.11	0.94 ± 0.09
Model2	$m = 64, h = 4, s = 256$	0.70 ± 0.01	1.24 ± 0.11	0.96 ± 0.09
Model3	$m = 128, h = 4, s = 512$	0.70 ± 0.01	1.24 ± 0.11	0.96 ± 0.09
Model4	$m = 128, h = 8, s = 512$	0.71 ± 0.01	1.23 ± 0.10	0.94 ± 0.08
Model5	$m = 256, h = 8, s = 1024$	0.71 ± 0.02	1.23 ± 0.12	0.95 ± 0.10

feedforward network (FFN). The Transformer encoder is responsible for computing self-attention across the input sequence, producing in output a representation of the input taking into consideration the relations among the different positions of the input sequence.

The architecture adopted here is directly derived from the original Transformer definition (Vaswani *et al.* 2017). Formally, given the input sequence C of dimension $L \times m$, each head i of the multi-head attention layer adopts three matrices of learnable weights, called A_Q^i , A_K^i , and A_V^i each having dimension $m \times r$, where $r = m/h$ (r is equal to 16 in our case) and h is the number of attention heads (here set to 8). The input matrix C is firstly projected using the A_Q^i , A_K^i , and A_V^i as follows:

$$Q^i = C \cdot A_Q^i \quad (2)$$

$$K^i = C \cdot A_K^i \quad (3)$$

$$V^i = C \cdot A_V^i \quad (4)$$

where \cdot denotes the matrix product operator.

Then, for each head i , an attention output Z_i of dimension $L \times r$ is computed as follows:

$$Z^i = \text{softmax}\left(\frac{Q^i \cdot K^i}{\sqrt{r}}\right) \cdot V^i \quad (5)$$

The different Z^i from the different attention heads are then concatenated, multiplied with an output weight matrix A_O of

dimension $m \times m$, and the result added to the input C matrix by residual connection:

$$Z = [Z^1, Z^2, \dots, Z^b] \cdot A_O + C \quad (6)$$

where $[\]$ denotes the concatenation operator (by rows) and the Z output matrix has dimension $L \times m$ (as the input matrix C).

The final Transformer encoder output F of dimension $L \times m$ is computed by independently applying a position-wise Feed-Forward Network (FFN) to each position $1 \leq j \leq L$ of Z , and residual connection addition. In other words, each row f_j of F is computed as follows:

$$f_j = \text{FFN}(z_j) + z_j = \text{ReLU}(z_j \cdot W_1 + b_1) \cdot W_2 + b_2 + z_j \quad (7)$$

where ReLU is the activation function defined as $g(x) = \max(0, x)$, and W_1, b_1, W_2, b_2 are position-independent weight parameters and biases of the FFN, having, respectively, dimensions $m \times s, 1 \times s, s \times m$, and $1 \times m$. Here, we set the dimensionality s of the hidden layer of the position-wise FFN to 512.

The output matrix F of the Transformer encoder is then collapsed to two unidimensional vectors by means of Global Average and Max Pooling layers, denoted as p_{ave} and p_{max} , respectively, acting on the first dimension L of the matrix F :

$$p_{\text{ave}}(F) = \left(1/L \sum_{j=1}^L f_{j1}, \dots, 1/L \sum_{j=1}^L f_{jm} \right) \quad (8)$$

$$p_{\text{max}}(F) = (\max_j f_{j1}, \dots, \max_j f_{jm}) \quad (9)$$

The pooled vectors are then concatenated into a single vector P of size $2m$:

$$P = [p_{\text{ave}}(F), p_{\text{max}}(F)] \quad (10)$$

where $[\]$ denotes the concatenation operator.

P is finally processed by a linear FFN parametrized by a weight vector w^O and bias b^O , producing in output the predicted $\Delta\Delta G$ value \hat{y} :

$$\hat{y} = P \cdot w^O + b^O \quad (11)$$

2.2.3 Model training and implementation

Given a dataset of N protein single residue variations $\mathcal{D} = \{D^1, D^2, \dots, D^N\}$, corresponding target $\Delta\Delta G$ values $Y = \{y^1, y^2, \dots, y^N\}$, and model predictions $\hat{Y} = \{\hat{y}^1, \hat{y}^2, \dots, \hat{y}^N\}$ training is carried out minimizing the Mean Squared Error function on training data:

$$\text{MSE}(Y, \hat{Y}) = \frac{1}{N} \sum_{i=1}^N (y^i - \hat{y}^i)^2 \quad (12)$$

The optimization is carried out with gradient descent and using the Adam optimizer (Kingma and Ba 2017). The training data are split into mini batches of size 128. Training is performed for 500 epochs and stopped when error starts

decreasing on a subset of the training set used as validation data (early stopping).

The training procedure and the model itself were implemented using the PyTorch Python package (<https://pytorch.org>). All experiments were carried out on a single machine equipped with two AMD EPYC 7413 CPUs with 48/96 CPU cores/threads and 768 GB RAM.

2.2.4 DDGemb web server

We release DDGemb as a web server at <https://ddgemb.bio.comp.unibo.it>. The server provides a user-friendly web interface, providing both interactive and batch submission modes.

In the interactive mode, the user can predict $\Delta\Delta G$ for up to 100 variations occurring on a single protein sequence as an input. Results of interactive jobs can be directly visualized on the DDGemb website and downloaded in JSON or TSV formats. Both single- and multi-point variations are supported.

Batch submission mode is dedicated to larger prediction jobs. The user can submit up to 2000 variations occurring on at most 500 proteins per job. Results of batch jobs can be downloaded in JSON and/or TSV formats.

In both cases, user jobs are maintained for a month after completion. The user can retrieve job results using the job assigned upon submission.

The web application is implemented using Django (version 4.0.4), Bootstrap (version 5.3.0), JQuery (version 3.6.0), and neXtProt FeatureViewer (version 1.3.0-beta6) for graphical visualization of predicted variants and their $\Delta\Delta G$ along the sequence.

2.3 Scoring performance

To score the performance of the different approaches, we use the following well-established scoring indexes. In what follows, e and p are experimental and predicted $\Delta\Delta G$ values, respectively, while p^{dir} and p^{inv} are predicted $\Delta\Delta G$ for direct and corresponding reverse variations, respectively.

The Pearson's correlation coefficient (PCC) between e and p is defined as:

$$PCC(e, p) = \frac{\sum_{i=1}^N (e_i - \bar{e})(p_i - \bar{p})}{\sqrt{\sum_{i=1}^N (e_i - \bar{e})^2} \sqrt{\sum_{i=1}^N (p_i - \bar{p})^2}} \quad (13)$$

where \bar{e} and \bar{p} are average experimental and predicted $\Delta\Delta G$ values, respectively.

The Root Mean Square Error ($RMSE$) between e and p is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (e_i - p_i)^2}{N}} \quad (14)$$

The Mean Absolute Error (MAE) between e and p is defined as:

$$MAE = \frac{\sum_{i=1}^N |e_i - p_i|}{N} \quad (15)$$

To score anti-symmetry properties of the different tools we adopted two additional measures defined in literature (Pucci *et al.* 2018).

The Pearson's correlation between p^{dir} and p^{inv} , referred to as r_{d-r} , defined as:

$$r_{d-r} = r(p^{dir}, p^{inv}) \quad (16)$$

The anti-symmetry bias $\langle\delta\rangle$ defined as:

$$\langle\delta\rangle = \frac{\sum_{i=1}^N (p_i^{dir} - p_i^{inv})}{2N} \quad (17)$$

3 Results

3.1 Cross-validation results on the S2450 dataset

In a first experiment, we performed 5-fold cross-validation on the S2450 dataset. To this aim, we adopted the most stringent data split procedure proposed in Fariselli *et al.* (2015), which consists in retaining all variations occurring in the same protein within the same cross-validation subset and in confining proteins with >25% sequence identity in the same subset. Sequence comparison was performed using full-length UniProt sequences.

Considering both average *PCC*, *RMSE*, and *MAE* values and the corresponding standard deviations, the highest performance is obtained using the architecture Model4, including 128 1D-convolutional filters, 8 Transformer encoder attention heads and 512 hidden units in the Transformed encoder FFN output (Table 1). This configuration has been then chosen as the final model.

3.2 Prediction of single-point variations on the S669 dataset

We compared DDGEmb with several state-of-the-art methods introduced in the past years using the common benchmark dataset S669.

Results for 21 different methods were taken from (Pancotti *et al.* 2022), except DDGEmb, presented in this work, PROSTATA (Umerenkov *et al.* 2023), THPLM (Gong *et al.* 2023), and ThermoMPNN (Dieckhaus *et al.* 2024) whose results were extracted from the respective papers. Scored methods include nine sequence-based predictors, namely INPS (Fariselli *et al.* 2015), ACDC-NN-Seq (Pancotti *et al.* 2022), DDGun (Montanucci *et al.* 2019), I-Mutant3-Seq (Capriotti *et al.* 2005), SAAFEC-SEQ (Li *et al.* 2021), MUPro (Cheng *et al.* 2006), PROSTATA (Umerenkov *et al.* 2023), THPLM (Gong *et al.* 2023) and ThermoMPNN (Dieckhaus *et al.* 2024), and fifteen structure-based methods, ACDC-NN (Benevenuta *et al.* 2021), PremPS (Chen *et al.* 2020), DDGun3D (Montanucci *et al.* 2019), INPS-3D (Savojardo *et al.* 2016), ThermoNet (Li *et al.* 2020), MAESTRO (Laimer *et al.* 2015, 2016), Dynamut (Rodrigues *et al.* 2014a), PoPMuSiC (Dehouck *et al.* 2011), DUET (Rodrigues *et al.* 2014a), SDM (Worth *et al.* 2011), mCSM (Pires *et al.* 2014b), Dynamut2 (Rodrigues *et al.* 2021), I-Mutant3-3D (Capriotti *et al.* 2005), Rosetta (Kellogg *et al.* 2011), and FoldX (Schymkowitz *et al.* 2005). Results are listed in Table 2.

For each method we report *PCC*, *RMSE* and *MAE* computed considering (i) all variations (both direct and reverse) in the dataset (columns under “Total”), (ii) only direct variations (columns under “Direct”), and (iii) only reverse variations (columns under “Reverse”). In addition, we computed PCC_{d-r} and $\langle\delta\rangle$.

On the S669 dataset, DDGEmb reports the highest *PCC*, *RMSE*, and *MAE* values (Total, Direct and Reverse). Our DDGEmb overall scores as the top-performing tool in this

Table 2. Comparative benchmark of different sequence- and structure-based methods on the S669 independent test set of single-point variations.

Method	Input	Total			Direct			Reverse			Symmetry	
		<i>PCC</i>	<i>RMSE</i>	<i>MAE</i>	<i>PCC</i>	<i>RMSE</i>	<i>MAE</i>	<i>PCC</i>	<i>RMSE</i>	<i>MAE</i>	r_{d-r}	$\langle\delta\rangle$
DDGEmb	SEQ	0.68	1.40	0.99	0.53	1.40	0.99	0.52	1.40	0.99	-0.97	0.01
PROSTATA	SEQ	0.65	1.45	1.00	0.49	1.45	1.00	0.49	1.45	0.99	-0.99	-0.01
ACDC-NN	3D	0.61	1.5	1.05	0.46	1.49	1.05	0.45	1.5	1.06	-0.98	0.02
INPS-Seq	SEQ	0.61	1.52	1.1	0.43	1.52	1.09	0.43	1.53	1.1	-1.00	0.00
PremPS	3D	0.62	1.49	1.07	0.41	1.5	1.08	0.42	1.49	1.05	-0.85	0.09
ACDC-NN-Seq	SEQ	0.59	1.53	1.08	0.42	1.53	1.08	0.42	1.53	1.08	-1.00	0.00
DDGun3D	3D	0.57	1.61	1.13	0.43	1.6	1.11	0.41	1.62	1.14	-0.97	0.05
INPS3D	3D	0.55	1.64	1.19	0.43	1.5	1.07	0.33	1.77	1.31	-0.5	0.38
THPLM	SEQ	0.53	1.63		0.39	1.60		0.35	1.66		-0.96	-0.01
ThermoNet	3D	0.51	1.64	1.2	0.39	1.62	1.17	0.38	1.66	1.23	-0.85	0.05
DDGun	SEQ	0.57	1.74	1.25	0.41	1.72	1.25	0.38	1.75	1.25	-0.96	0.05
MAESTRO	3D	0.44	1.8	1.3	0.5	1.44	1.06	0.2	2.1	1.66	-0.22	0.57
ThermoMPNN	SEQ	0.43	1.52									
Dynamut	3D	0.5	1.65	1.21	0.41	1.6	1.19	0.34	1.69	1.24	-0.58	0.06
PoPMuSiC	3D	0.46	1.82	1.37	0.41	1.51	1.09	0.24	2.09	1.64	-0.32	0.69
DUET	3D	0.41	1.86	1.39	0.41	1.52	1.1	0.23	2.14	1.68	-0.12	0.67
I-Mutant3.0-Seq	SEQ	0.37	1.91	1.47	0.34	1.54	1.15	0.22	2.22	1.79	-0.48	0.76
SDM	3D	0.32	1.93	1.45	0.41	1.67	1.26	0.13	2.16	1.64	-0.4	0.4
mCSM	3D	0.37	1.96	1.49	0.36	1.54	1.13	0.22	2.3	1.86	-0.05	0.85
Dynamut2	3D	0.36	1.9	1.42	0.34	1.58	1.15	0.17	2.16	1.69	0.03	0.64
I-Mutant3.0	3D	0.32	1.96	1.49	0.36	1.52	1.12	0.15	2.32	1.87	-0.06	0.81
Rosetta	3D	0.47	2.69	2.05	0.39	2.7	2.08	0.4	2.68	2.02	-0.72	0.61
FoldX	3D	0.31	2.39	1.53	0.22	2.3	1.56	0.22	2.48	1.5	-0.2	0.34
SAAFEC-SEQ	SEQ	0.26	2.02	1.54	0.36	1.54	1.13	-0.01	2.4	1.94	-0.03	0.83
MUPro	SEQ	0.32	2.03	1.58	0.25	1.61	1.21	0.20	2.38	1.96	-0.32	0.95

Results for all methods except DDGEmb, THPLM, ThermoMPNN, and PROSTATA were taken from (Pancotti *et al.* 2022). For PROSTATA and THPLM direct and reverse *PCC*, *RMSE*, and *MAE* were taken from the reference papers (Gong *et al.* 2023, Umerenkov *et al.* 2023). ThermoMPNN results were taken from Dieckhaus *et al.* (2024) PROSTATA total *PCC*, Total *RMSE*, Total *MAE*, PCC_{d-r} , and $\langle\delta\rangle$ were computed using the predictions available at the method GitHub repository. Bold values highlight the top-performing method(s) on the respective metric.

Table 3. Comparative benchmark of different methods on multi-point variations from the PTmul-NR dataset.

Method	PCC	RMSE	MAE
DDGemb	0.59	2.16	1.59
FoldX	0.36	5.51	3.66
MAESTRO	0.28	2.55	1.88
DDGun	0.23	2.55	2.10
DDGun3D	0.17	2.57	2.08

benchmark, significantly surpassing both structure- and sequence-based methods.

3.3 Prediction of multi-point variations

We finally tested DDGemb in the prediction of multi-point variations using the PTmul-NR dataset. This allowed us to directly compare with other methods such as DDGun/DDGun3D (Montanucci *et al.* 2019), MAESTRO (Laimer *et al.* 2015), and FoldX (Schymkowitz *et al.* 2005). Results are listed in Table 3.

On the PTmul-NR dataset, DDGemb significantly outperforms DDGun, DDGun3D, FoldX, and MAESTRO on the prediction of multi-point variations, achieving the highest correlation coefficient of 0.59, and the lowest RMSE and MAE values of 2.16 and 1.59, respectively.

These results suggest that DDGemb can be effectively used for assessing with high accuracy the impact of multi-point variations on protein stability. Remarkably, our method has been trained using only single-point variants, suggesting the ability of the proposed approach to generalize on multi-point variants as well.

4 Conclusion

In this work, we present DDGemb, a novel method based on protein language models and Transformers to predict protein stability change ($\Delta\Delta G$) upon single- and multi-point variations. Our method has been trained on a high-quality dataset derived from literature and tested using recently introduced benchmark datasets of thermodynamic data for single- and multi-point variations. In all the benchmarks, DDGemb reports performances that are superior to the state of art, outperforming both sequence- and structure-based methods and achieving an overall PCC of 0.68 on single-point variations. Moreover, on multi-point variations, our method reports a PCC of 0.59, which is significantly higher than the one achieved by the second top-performing approach, FoldX, reporting PCC equal to 0.36. Our study suggests the relevance of a transformer architecture specifically fine-tuned to predict the $\Delta\Delta G$ upon variation, in combination with numerical representations provided by protein language models.

Conflict of interest: None declared.

Funding

The work was supported by the European Union NextGenerationEU through the Italian Ministry of University and Research under the projects “Consolidation of the Italian Infrastructure for Omics Data and Bioinformatics” (ElixirxNext-GenIT) (Investment PNRRM4C2-I3.1, Project IR_0000010, CUP B53C22001800006), “HEAL ITALIA” (Investment PNRR-M4C2-I1.3, Project PE_00000019, CUP J33C22002920006), “National Centre for HPC, Big Data

and Quantum Computing” (Investment PNRR-M4C2-I1.4, Project CN_00000013).

Data availability

The data underlying this article are available in the article and in its online supplementary material.

References

- Bava KA, Gromiha MM, Uedaira H *et al.* ProTherm, version 4.0: thermodynamic database for proteins and mutants. *Nucleic Acids Res* 2004;32:D120–1.
- Benevuta S, Pancotti C, Fariselli P *et al.* An antisymmetric neural network to predict free energy changes in protein variants. *J Phys D Appl Phys* 2021;54:245403.
- Capriotti E, Fariselli P, Rossi I *et al.* A three-state prediction of single point mutations on protein stability changes. *BMC Bioinformatics* 2008;9:S6.
- Capriotti E, Fariselli P, Casadio R *et al.* I-Mutant2.0: predicting stability changes upon mutation from the protein sequence or structure. *Nucleic Acids Res* 2005;33:W306–10.
- Chen Y, Lu H, Zhang N *et al.* PremPS: predicting the impact of missense mutations on protein stability. *PLoS Comput Biol* 2020;16:e1008543.
- Cheng J, Randall A, Baldi P *et al.* Prediction of protein stability changes for single-site mutations using support vector machines. *Proteins* 2006;62:1125–32.
- Dana JM, Gutmanas A, Tyagi N *et al.* SIFTS: updated structure integration with function, taxonomy and sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Res* 2019;47:D482–9.
- Dauparas J, Anishchenko I, Bennett N *et al.* Robust deep learning-based protein sequence design using ProteinMPNN. *Science* 2022;378:49–56.
- Dehouck Y, Kwasigroch JM, Gilis D *et al.* PoPMuSiC 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality. *BMC Bioinformatics* 2011;12:151.
- Dieckhaus H, Brocchiacono M, Randolph NZ *et al.* Transfer learning to leverage larger datasets for improved prediction of protein stability changes. *Proc Natl Acad Sci USA* 2024;121:e2314853121.
- Fariselli P, Martelli PL, Savojarado C *et al.* INPS: predicting the impact of non-synonymous variations on protein stability from sequence. *Bioinformatics* 2015;31:2816–21.
- Gong J, Wang J, Zong X *et al.* Prediction of protein stability changes upon single-point variant using 3D structure profile. *Comput Struct Biotechnol J* 2023;21:354–64. <https://doi.org/10.1016/j.csbj.2022.12.008>
- Kellogg EH, Leaver-Fay A, Baker D *et al.* Role of conformational sampling in computing mutation-induced changes in protein structure and stability. *Proteins* 2011;79:830–8.
- Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv, arXiv:1412.6980v9, 2017, preprint: not peer reviewed.
- Laimer J, Hofer H, Fritz M *et al.* MAESTRO—multi agent stability prediction upon point mutations. *BMC Bioinformatics* 2015;16:116.
- Laimer J, Hiebl-Flach J, Lengauer D *et al.* MAESTROweb: a web server for structure-based protein stability prediction. *Bioinformatics* 2016;32:1414–6.
- Li B, Yang YT, Capra JA *et al.* Predicting changes in protein thermodynamic stability upon point mutation with deep 3D convolutional neural networks. *PLoS Comput Biol* 2020;16:e1008291.
- Li G, Panday SK, Alexov E *et al.* SAAFEC-SEQ: a sequence-based method for predicting the effect of single point mutations on protein thermodynamic stability. *Int J Mol Sci* 2021;22:606.
- Lin Z, Akin H, Rao R *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* 2023;379:1123–30.

- Montanucci L, Capriotti E, Frank Y *et al.* DDGun: an untrained method for the prediction of protein stability changes upon single and multiple point variations. *BMC Bioinformatics* 2019;20:335.
- Nair PS, Vihinen M. VariBench: a benchmark database for variations. *Hum Mutat* 2013;34:42–9.
- Notin P, Rollins N, Gal Y *et al.* Machine learning for functional protein design. *Nat Biotechnol* 2024;42:216–28.
- Pancotti C, Benevenuta S, Repetto V *et al.* A deep-learning sequence-based method to predict protein stability changes upon genetic variations. *Genes (Basel)* 2021;12:911.
- Pancotti C, Benevenuta S, Birolo G *et al.* Predicting protein stability changes upon single-point mutation: a thorough comparison of the available tools on a new dataset. *Brief Bioinform* 2022;23:bbab555.
- Pandey P, Ghimire S, Wu B *et al.* On the linkage of thermodynamics and pathogenicity. *Curr Opin Struct Biol* 2023;80:102572.
- Pires DEV, Ascher DB, Blundell TL *et al.* DUET: a server for predicting effects of mutations on protein stability using an integrated computational approach. *Nucleic Acids Res* 2014a;42:W314–9.
- Pires DEV, Ascher DB, Blundell TL *et al.* mCSM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics* 2014b;30:335–42.
- Pucci F, Bernaerts KV, Kwasigroch JM *et al.* Quantification of biases in predictions of protein stability changes upon mutations. *Bioinformatics* 2018;34:3659–65.
- Puglisi R. Protein mutations and stability, a link with disease: the case study of frataxin. *Biomedicines* 2022;10:425.
- Rives A, Meier J, Sercu T *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc Natl Acad Sci USA* 2021;118:e2016239118.
- Rodrigues CH, Pires DE, Ascher DB *et al.* DynaMut: predicting the impact of mutations on protein conformation, flexibility and stability. *Nucleic Acids Res* 2018;46:W350–5.
- Rodrigues CHM, Pires DEV, Ascher DB *et al.* DynaMut2: assessing changes in stability and flexibility upon single and multiple point missense mutations. *Protein Sci* 2021;30:60–9.
- Savojardo C, Fariselli P, Martelli PL *et al.* INPS-MD: a web server to predict stability of protein variants from sequence and structure. *Bioinformatics* 2016;32:2542–4.
- Schymkowitz J, Borg J, Stricher F *et al.* The FoldX web server: an online force field. *Nucleic Acids Res* 2005;33:W382–8.
- Umerenkov D, Nikolaev F, Shashkova TI *et al.* PROSTATA: a framework for protein stability assessment using transformers. *Bioinformatics* 2023;39:btad671.
- Vaswani A, Shazeer N, Parmar N *et al.* Attention is all you need. arXiv, arXiv:1706.03762v7, 2017, preprint: not peer reviewed.
- Vihinen M. Functional effects of protein variants. *Biochimie* 2021;180:104–20.
- Worth CL, Preissner R, Blundell TL *et al.* SDM—a server for predicting effects of mutations on protein stability and malfunction. *Nucleic Acids Res* 2011;39:W215–22.
- Xavier JS, Nguyen T-B, Karmarkar M *et al.* ThermoMutDB: A thermodynamic database for missense mutations. *Nucleic Acids Research* 2021;49:D475–9. <https://doi.org/10.1093/nar/gkaa925>