An antisymmetric neural network to predict free energy changes in protein variants

An antisymmetric neural network to predict free energy changes in protein variants

S Benevenuta[1], C Pancotti[1], P Fariselli[*], G Birolo[1] and T Sanavia[1]

[1]Department of Medical Sciences, University of Torino, Via Santena

E-mail: piero.fariselli@unito.it

[1]These two authors equally contributed to the work.
[*]Author to whom any correspondence should be addressed.

Abstract

The prediction of free energy changes upon protein residue variations is an important application in biophysics and biomedicine. Several methods have been developed to address this problem so far, including physical-based and machine learning models. However, most of the current computational tools, especially data-driven approaches, fail to incorporate the antisymmetric basic thermodynamic principle: a variation from wild-type to a mutated form of the protein structure (XW → XM) and its reverse process (XM → XW) must have opposite values of the free energy difference: ΔΔGWM = −ΔΔGMW. Here, we build a deep neural network system that, by construction, satisfies the antisymmetric properties. We show that the new method (ACDC-NN) achieved comparable or better performance with respect to other state-of-the-art approaches on both direct and reverse variations, making this method suitable for scoring new protein variants preserving the antisymmetry. The code is available at: https://github.com/compbiomed-unito/acdc-nn.

Keywords: deep learning, protein stability, free energy changes, antisymmetry, ACDC

1. Introduction

Disruptions of protein stability and its normal functions are linked to several diseases [1]. Protein stability perturbation can also contribute to the loss of function in haploinsufficient genes [2]. Usually, normal protein activity is accomplished by switching among a few folding structures representing local minima of the unfolding free energy. These minima might be affected by one or more non-synonymous DNA variations able to alter the amino acid sequence and, therefore, to increase or decrease the protein stability [3]. The main experimental measure for studying the effects of non-synonymous variants on protein stability is the difference of the Gibbs free energy of unfolding (ΔΔG) between the mutated structure of the protein and its wild-type form ($\Delta\Delta G = \Delta G_M - \Delta G_W$). The ΔΔG prediction is of great interest in protein engineering and in the interpretation of non-synonymous DNA variations.

The accurate prediction of ΔΔG is still an open challenge and several computational tools have been developed to tackle this problem so far [4–18]. However, as recently pointed out [3, 16, 19–22], most of the current methods suffer from biases due to the lack of antisymmetric predictions. The ΔΔG antisymmetry can be summarized as follows: given the wild-type (W) and mutated (M) protein structures, differing by one residue in position X, the quantity $\Delta\Delta G_{WM} = \Delta G_W - \Delta G_M$ represents the change in the protein stability caused by the amino acid substitution $X_W \rightarrow X_M$. Similarly, given the symmetry between the two molecular systems M and W, for the reverse variation $X_M \rightarrow X_W$ the corresponding change in the Gibbs free energy has the opposite sign:

$$\Delta\Delta G_{WM} = -\Delta\Delta G_{MW}. \qquad (1)$$

This is due to the thermodynamic rule linking unfolding (U) and folding (F) states, which can be described at the equilibrium by the relationship:

$$\Delta G_{UF} = -RTlog\left(\frac{[U]}{[F]}\right). \qquad (2)$$

For the reverse transformation from F to U we have:

$$\Delta G_{FU} = -RTlog\left(\frac{[F]}{[U]}\right) = -\Delta G_{UF}. \qquad (3)$$

If we then consider the two proteins W and M, which differ by a residue along the sequence, we can easily find the previous equation:

$$\Delta\Delta G_{WM} = RTlog\left(\frac{[U]}{[F]}\right)_W - RTlog\left(\frac{[F]}{[U]}\right)_M = -\Delta\Delta G_{MW}. \qquad (4)$$

Antisymmetric predictions are not easy to achieve; Usmanova *et al* [20] showed that, in order to fulfill perfect antisymmetry, structure-based methods should be able to generate the low- est energy structure of the variant starting from the native one and vice-versa. This is a challenging task due to the intrinsic features of the tools. Therefore, biases δ_{MW} and δ_{WM} in both directions should be considered in the equation:

$$\Delta\Delta G_{WM} = -\Delta\Delta G_{MW} + \delta_{MW} + \delta_{WM}. \qquad (5)$$

In this paper, we present ACDC-NN, a new method that scores protein stability changes in protein variants while pre- serving the physical property of antisymmetry that character- izes $\Delta\Delta G$ measures. The main points provided by this new tool and investigated in this study are:

- The lack of experimental data currently available [3] and the related strategies to address the problem.
- The development of an antisymmetric method through a specifically designed loss function and the implementation of physical constraints in the network structure.
- The construction of a novel deep neural network architec- ture able to encode pairwise contact potentials and evolu- tionary information in convolutional filters.
- The proper evaluation of the methods taking into account the sequence similarity between the protein sequences, an issue seldom addressed in the assessment of most of the cur- rent approaches [3].

ACDC-NN outperforms the state-of-the-art methods when applied to an unbiased dataset for sequence similarity, accounting for the reverse variants. The source code of ACDC-NN is publicly available at the github repository https://github.com/compbiomed-unito/acdc-nn.

2. Results

2.1 Learning antisymmetry using DDGun3D predictions

To address the deficiency of experimental data, we used the transfer learning technique (see section 3). The pre-training phase was performed on a subset of artificially created variants starting from the Ivankov dataset [20], that we named IvankovDDGun. The ΔΔG values for this dataset were calculated using DDGun3D [23], an untrained predictor mainly based on statistical potentials. DDGun3D uses sequence and structure-based features and is able to achieve comparable performance with the other state-of-the-art predictors while maintaining antisymmetric properties. The ability to reconstruct the predictions is measured using the Pearson correlation coefficient r and root mean squared error (RMSE); the ΔΔG antisymmetry is measured with rd − i (equation (6)) and the bias ⟨δ⟩. On the IvankovDDGun test set, ACDC-NN reproduced the DDGun3D predictions (r = 0.97 − 0.98) while achieving almost perfect antisymmetry with rd − i close to −1 (table 1). The obtained results showed that using only sequence and structure information, ACDC-NN could encode the information that DDGun3D calculates using the statistical potentials by Bastoll et al [24] and Skolnick et al [25], the BLOSUM62 substitution matrix [26], the Kyte–Doolittle hydrophobicity [27] and the residue solvent accessibility.

2.2. Learning to predict experimental ΔΔG values

After the pre-training phase on simulated data, ACDC-NN was trained on the experimentally derived ΔΔG values from S2648 through 10-fold cross-validation, achieving r = 0.55 and RMSE = 1.33. As expected, the performance on real data was significantly lower than those obtained from the simulation. However, the results are in line with the current state-ofthe-art methods on the same dataset [3]. To evaluate and compare the antisymmetry property and the prediction performance on the experimental data, we used Ssym dataset [21], which was specifically built to assess the ability of predicting equally well both the direct and reverse variants of free energy. In particular, we predicted each variant from Ssym using in the training sets the proteins showing less than 25% identity with respect to those reported in Ssym (see section 3). The comparative performance of the different methods on the prediction of direct and related reverse variants, as well as the correlation between them on the Ssym dataset, were reported in table 2 in terms of Pearson correlation coefficient (r) and root mean square error (RMSE) between the predicted values and the experimental ΔΔGs. The antisymmetry was assessed through rd − i and the bias ⟨δ⟩. RMSE and ⟨δ⟩ are expressed in kcal mol−1.

The results reported in table 2 showed that, considering the correlation of the reverse variants, ACDC-NN outperformed all the other predictors, with similar performance obtained by DDGun3D, the method used for the pre-training. Other predictors with higher performance on direct variants performed worse when considering the reverse, indicating the missing thermodynamic antisymmetry, as previously highlighted in Pucci et al [21, 22]. It is worth highlighting that, among the methods reported in table 2, only Inps-NoSeqId, ThermoNet, ACDC-NN and ACDC-NN∗ were trained in cross-validation removing the sequence identity (i.e. sequence similarity y <25%). In addition, ACDC-NN showed excellent results in antisymmetry (−0.98) when only one structure was available (ACDC-NN) and it achieved perfect antisymmetry (−1.00) with both structures (ACDC-NN∗). Both versions of the model outperformed all the other methods on both direct and reverse variants of the Ssym dataset.

2.3. Evaluation of ACDC-NN on p53 and myoglobin variants

We also evaluated ACDC-NN on two other experimental datasets consisting of single-variants in two clinically relevant proteins: the p53 transcription factor and the myoglobin. The p53 dataset was created by Pires et al [14], and consists of 42 variants within the p53 DNA binding domain. The myoglobin dataset consists of 134 variants (113 variants with some multiple experiments) collected by Kepp [30]. Since both myoglobin and p53 are present in the S2648 training dataset, we predicted the variants using ACDCNN in cross-validation (see section 3), removing sequence similarity with the two proteins. In order to predict the reverse variants we used MODELLER [31] to generate the

mutated structures, which are available at our on-line repository https://github.com/compbiomed-unito/acdc-nn. ACDC-NN predictions were compared with the experimentally-determined ΔΔG values as reported in figure 1. The performance obtained in both datasets is in agreement with the results observed on Ssym dataset (table 2). The other available deep-learning approach, Thermonet, performed slightly worse than ACDC-NN, as reported in tables 3 and 4.

2.4. Analysis of the convolutional filters

ACDC-NN consists of two independent convolutional layers, one spanning the 3D residue contacts (KT) and the other one spanning the sequence nearest neighbors (KS). Figures 2 and 3 report heatmaps of the two filters highlighting the original residue positions (from alanine (A) to tyrosine (Y)). Interestingly, KS filters span an input containing the central residue (the one that undergoes the residue substitution) while the KT filters span an input containing only the 3D neighbors. This difference leads to the generation of a main diagonal appearing in the KS filters only. It is worth highlighting that the filters generate two 20 × 20 matrix and that those matrices can be correlated with both pairwise contact potentials and evolutionary substitution matrices. The filter matrix KS showed statistically significant correlation with both evolutionary-based substitution matrices and pairwise contact potential, specifically −0.75 with BLOSUM62 [26] and 0.54 with Simons statistical potential [32], respectively. On the other hand, the filter matrix KT significantly correlates only with contact potentials, particularly with the BastollaVendruscolo statistical potential [24], achieving a Pearson correlation equal to 0.51. These findings indicate that not only the network has learnt both the physical-based and the evolutionary information embedded in the DDGun3D score, but also it is able to generalize by blending the information to other statistical potentials (as in [32]), which showed that the network can also extract consistent physical relationships from the data.

3. Materials and methods

3.1. Datasets and cross-validation

Three previously collected datasets and an artificial dataset were used:

• S2648, which contains 2,648 manually curated variants with experimentally measured ΔΔG values [7].

• Ssym, which provides variations on proteins whose wildtype and mutated 3D structures are solved by x-ray crystallography. It contains 684 variations, and half of them are reverse variations [21].

• Ivankov2000, which contains 2,000 single-point variants with available structures, 1,000 in a given direction and 1,000 in the opposite one [20]. This dataset does not report measured ΔΔG values because it has been specifically designed to score the anti-symmetric properties of the available predictors.

• IvankovDDGun, an artificial dataset derived from Ivankov2000 by generating all the possible direct and revers variations in every sequence position and assigning the corresponding DDGun3D predictions as ΔΔG values. This dataset was used only to pre-train the neural network.

Since these datasets also contain variants in the same proteins or in proteins with remarkable sequence similarity, to properly assess the predictive performance of a method avoiding overfitting issues, it is mandatory to consider the sequence identity between the proteins whose variants are used to fit the model and those used to test it [3]. Therefore, we generated cross-validation folds by generating validation and test sets characterized by proteins sufficiently different from those available in the training sets. To this aim, blastclust algorithm [33] was used to generate clusters of proteins with sequence identity lower than 25% (command blasclust -i infile.fasta -o out.custers -p T -L 0.5 -b F -S 25).

3.1.1. Sequence profiles.

For each protein containing at least one variation, the sequence was derived from the ATOM field of the PDB coordinate file. Multiple sequence alignments were applied against the Uniprot database (release 2016) using hhblits tool [34] with default parameters. From each alignment, a profile was derived, i.e. a matrix N × 20 where N is the length of the protein while the 20 columns represent the 20 amino acid types. Each profile entry Prof(i, a) corresponds to the frequency of the residue type a in the protein-sequence position i.

3.1.2. Measure of performance

To evaluate the performance, Pearson correlation (indicated by r) and RMSE were estimated between the predicted and observed ΔΔG values. To assess the antisymmetric property of ΔΔG predictors, we adopted two scoring indices: rd − i and ⟨δ⟩. rd − i is the Pearson correlation coefficient between the direct and the corresponding reverse variations:

$$r_{d-i} = \frac{Cov(\Delta\Delta G^{dir}, \Delta\Delta G^{inv})}{\sigma_{dir}\sigma_{inv}}, \qquad (6)$$

where Cov is the covariance and σ is the standard deviation. ⟨δ⟩ is the average bias quantifying the prediction shift:

$$\langle\delta\rangle = \frac{\sum_{i=1}^{N}(\Delta\Delta G_i^{dir} + \Delta\Delta G_i^{inv})}{2N}. \qquad (7)$$

A perfectly antisymmetric method should have rd − i equal to −1 and ⟨δ⟩ equal to 0.

3.2. Building ACDC-NN

To address the antisymmetry property, we built an intrinsically antisymmetric neural network architecture, specifically an Antisymmetric Convolutional Differential Concatenated Neural Network (ACDC-NN). The ACDC-NN is a convolutional network that takes two separated inputs for direct and reverse variations, processes these variations through convolutional operations and then uses the extracted features as input for two siamese neural networks with shared weights. Since a perfectly antisymmetric predictor should hold the equality $\Delta\Delta G_{WM} = -\Delta\Delta G_{MW}$, we designed a specific loss function that constrains the Neural Network to learn this property by considering the following equations:

$$\Delta\Delta G_{WM} = \frac{1}{2}(\Delta\Delta G_{WM} + \Delta\Delta G_{WM}) = \frac{1}{2}(\Delta\Delta G_{WM} - \Delta\Delta G_{MW})$$
$$\Delta\Delta G_{WM} = -\Delta\Delta G_{MW} \Rightarrow \Delta\Delta G_{WM} + \Delta\Delta G_{MW} = 0. \qquad (8)$$

We implemented these constraints through the following customized loss function:

$$J = \log(\cosh(D - y)) + abs(S), \quad \text{with D} = (O_D - O_I)/2,\ S = (O_D + O_I)/2, \tag{9}$$

where y is the experimental ΔΔG value, while OD and OI are the outputs of the direct and reverse modules of the Differential Siamese Network, respectively. log(cosh(x)) was used since it is less sensitive to outliers than the classical mean square error loss. Each of the two inputs consists of 620 elements to code variation, sequence and structure information:

• Variation (V): 20 features (one for each amino acid) coding for the variation by setting all the entries to 0 with the exception of the wild-type and the variant residue positions set to −1 and 1, respectively. This input corresponds to a one-dimensional matrix V ∈ R 20×1 .

• Sequence (S or 1D-input): 100 features representing protein profile information of the sequence neighborhood. Considering i as the variant position in the sequence, we used a window of 2 residues, i.e. [i − 2,i − 1,i,i + 1,i + 2], so to obtain 20 × 5 elements, with the profile information of these 5 positions. This input corresponds to a matrix S ∈ R 5×20 .

• Structure (T or 3D-input): 500 features representing protein profile information of the 3D-structure neighborhood. We considered residues up to 5 Å from the variation, taking a maximum of 25 residues sorted according to their distance in Å from the amino acid of interest. This input corresponds to a matrix T ∈ R 25×20 .

On both S and T, a 2D convolutional layer was applied with 20 filters, i.e. a kernel equal to (1, 20) and stride (1, 1), reported as Keras-style parameters. This layer generates two 20 × 20 filter matrices KS and KT shown in figures 2 and 3.

$$\begin{aligned} &\text{2D-Conv}(S, K_S) = S' \quad \text{where } S' \in \mathbb{R}^{5\times 20} \\ &\text{2D-Conv}(T, K_T) = T' \quad \text{where } T'' \in \mathbb{R}^{25\times 20}. \end{aligned} \tag{10}$$

The filter matrices KS and KT were learned during the training phase. To make the model invariant to the order of the 3D neighbors, 2D global average pooling operation (2D-GAP) was applied on T ′ (which encodes the structure information):

$$\text{2D-GAP}(T') = T'' \quad \text{where } T'' \in \mathbb{R}^{1\times 20}. \tag{11}$$

Then the Dot product was calculated between the variant vector V and both T ′ ′ and S ′ :

$$\begin{aligned} D &= T'' \cdot V, \quad \text{where } D \in \mathbb{R}^{1\times 1} \\ E &= S' \cdot V, \quad \text{where } E \in \mathbb{R}^{5\times 1}. \end{aligned} \tag{12}$$

After all these operations, we obtained six processed features, computed for both the direct variation

and its reverse. These features were then concatenated with the variant vector V and used as input to the Differential Siamese Networks, which have shared weights, and their outputs were combined in two final Lambda layers of 'difference/2' and average. Thanks to the loss function described above (9), the average output is minimized and the difference is as close as possible to the true ΔΔG values. The ACDC-NN architecture is displayed in figures 4 and 5.

3.2.1. Pre-training

Pre-training was performed on a subset of artificial ΔΔG values extracted from IvankovDDGun dataset. For all the direct and reverse variations, each sequence position was generated and the ΔΔG values predicted with DDGun3D were used as output. ACDC-NN was trained on 400,000 simulated variations split in half for direct and reverse variations. A validation set (100,000), built without intersections or sequence identity with the training dataset, was generated to search for the optimal parameters. The final model was then tested on a blind test set built in the same way (100,000) (see section 2). We found that the optimal configuration for the differential net consisted of two hidden layers with 32 and 16 units, respectively. For a full overview of all the optimal parameters, see table 5.

3.2.2. Transfer learning using the experimental data

Starting from the original network, ACDC-NN was re-trained on the S2648 dataset of experimental ΔΔG values [7], splitting the data and removing the sequence similarity between training, validation and test sets. We fixed the weights of the convolutional part that learned a compact representation of the structure and sequence features, while the differential siamese network part was re-trained. To increase the size of the training set we also added the experimentally-determined structures from Ivankov2000 dataset and assigned the DDGun3D predictions as outputs.

3.2.3. Predictions

The ACDC-NN model was built so that it can be used to make predictions in two different cases:

• When both the wild-type and variant structures are available, these are respectively used as direct and reverse inputs so that the network can provide a prediction that, by construction, is perfectly antisymmetric with respect to the network itself.

• When only the wild-type structure is available, as usual, the reverse input is created starting from the direct one by inverting the variation encoding, but preserving the same structure. The prediction in this case is not perfectly antisymmetric, but still very close, as shown in section 2.

4. Discussion and conclusion

Very few methods in the literature satisfy the principle of antisymmetry imposed by thermodynamics, either due to the skewed datasets on which they are trained or the intrinsic characteristics used. The main ACDC-NN novelty is the introduction of the physical property of antisymmetry in the learning process, exploiting a specifically designed loss function which minimizes the absolute difference of the direct and reverse predictions. Thanks to its architecture, ACDC-NN achieves perfect antisymmetry when both 3D structures (reverse and direct) are available. When only one structure is known, ACDC-NN creates an artificial reverse, preserving remarkable predictions and nearly perfect antisymmetry (table 2). This may lead to a mild break of the antisymmetry as shown in table 2 by the results of ACDC-NN∗ (when both structures are provided) vs ACDC-NN (only one structure at a time is provided and the reverse is simulated). Only recently, another deep learning approach (ThermoNet) has been introduced [18], which showed to have learnt the antisymmetry of the problem (table 2). Although ThermoNet and ACDC-NN are both based on deep learning architectures, they adopt very different input data types [18]. Another relevant point is that ACDC-NN has been evaluated using

variants in proteins which are not similar to those used for the training. When testing the performance of any method on a dataset, the correct procedure (considered only by a few methods so far) is to remove sequence similarities between the training and test data to avoid overfitting. Following this procedure, we therefore divided the Ssym dataset into ten subsets as described in the Method section. As shown in table 2, on this dataset ACDC-NN consistently outperformed all the existing methods. In conclusion, ACDC-NN is a user-friendly tool that shows comparable or better performance with the state-of-the-art methods while preserving perfect thermodynamic antisymmetry. In future studies, this method can be used for further improvements in order to predict multiple site variations and to build a robust sequencebased method when the structure information is not available. To reproduce the results presented in this work and to test new protein variants, the code is publicly available at https://github.com/compbiomed-unito/acdc-nn.

References

[1] Hartl F U 2017 Protein misfolding diseases Annu. Rev. Biochem. 86 21–6

[2] Birolo G, Benevenuta S, Fariselli P, Capriotti E, Giorgio E and Sanavia T 2021 Protein stability perturbation contributes to the loss of function in haploinsufficient genes Front. Mol. Biosci. 8 10

[3] Sanavia T, Birolo G, Montanucci L, Turina P, Capriotti E and Fariselli P 2020 Limitations and challenges in protein stability prediction upon genome variations: towards future applications in precision medicine Comput. Struct. Biotechnol. J. 18 1968–79

[4] Capriotti E, Fariselli P and Casadio R 2005 I-Mutant2.0: predicting stability changes upon mutation from the protein sequence or structure Nucleic Acids Res. 33 W306–10

[5] Chen C W, Lin J and Chu Y W 2013 iStable: off-the-shelf predictor integration for predicting protein stability changes BMC Bioinform. 14 S5

[6] Cheng J, Randall A and Baldi P 2006 Prediction of protein stability changes for single-site mutations using support vector machines Proteins 62 1125–32

[7] Dehouck Y, Kwasigroch J M, Gilis D and Rooman M 2011 PoPMuSiC 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality BMC Bioinform. 12 1–12

[8] Fariselli P, Martelli P L, Savojardo C and Casadio R 2015 INPS: predicting the impact of non-synonymous variations on protein stability from sequence Bioinformatics 31 2816–21

[9] Giollo M, Martin A J, Walsh I, Ferrari C and Tosatto S C 2014 NeEMO: a method using residue interaction networks to improve prediction of protein stability upon mutation BMC Genom. 15 1–11

[10] Laimer J, Hiebl-Flach J, Lengauer D and Lackner P 2016 MAESTROweb: a web server for structure-based protein stability prediction Bioinformatics 32 1414–16

[11] Masso M and Vaisman I I 2010 AUTO-MUTE: web-based tools for predicting stability changes in proteins due to single amino acid replacements Protein Eng. Des. Sel. 23 683–7

[12] Parthiban V, Gromiha M M and Schomburg D 2006 CUPSAT: prediction of protein stability upon point mutations Nucleic Acids Res. 34 W239–42

[13] Pires D E, Ascher D B and Blundell T L 2014 DUET: a server for predicting effects of mutations on protein stability using an integrated computational approach Nucleic Acids Res. 42 W314–19

[14] Pires D E, Ascher D B and Blundell T L 2014 mCSM: predicting the effects of mutations in proteins using graph-based signatures Bioinformatics 30 335–42

[15] Quan L, Lv Q and Zhang Y 2016 STRUM: structure-based prediction of protein stability changes upon single-point mutation Bioinformatics 32 2936–46

[16] Savojardo C, Fariselli P, Martelli P L and Casadio R 2016 INPS-MD: a web server to predict stability of protein variants from sequence and structure Bioinformatics 32 2542–4

[17] Schymkowitz J, Borg J, Stricher F, Nys R, Rousseau F and Serrano L 2005 The FoldX web server: an online force field Nucleic Acids Res. 33 W382–8

[18] Li B, Yang Y T, Capra J A and Gerstein M B 2020 Predicting changes in protein thermodynamic stability upon point mutation with deep 3D convolutional neural networks PLoS Comput. Biol. 16 e1008291

[19] Fang J 2020 A critical review of five machine learning-based algorithms for predicting protein stability changes upon mutation Brief. Bioinform. 21 1285–92

[20] Usmanova D R et al 2018 Self-consistency test reveals systematic bias in programs for prediction change of stability upon mutation Bioinformatics 34 3653–8

[21] Pucci F, Bernaerts K V, Kwasigroch J M and Rooman M 2018 Quantification of biases in predictions of protein stability changes upon mutations Bioinformatics 34 3659–65

[22] Montanucci L, Savojardo C, Martelli P L, Casadio R and Fariselli P 2019 On the biases in predictions of protein stability changes upon variations: the INPS test case Bioinformatics 35 2525–7

[23] Montanucci L, Capriotti E, Frank Y, Ben-Tal N and Fariselli P 2019 DDGun: an untrained method for the prediction of protein stability changes upon single and multiple point variations BMC Bioinform. 20 335

[24] Bastolla U, Farwer J, Knapp E W and Vendruscolo M 2001 How to guarantee optimal stability for most representative structures in the protein data bank Proteins 44 79–96

[25] Skolnick J, Godzik A, Jaroszewski L and Kolinski A 1997 Derivation and testing of pair potentials for protein folding. When is the quasichemical approximation correct? Protein Sci. 6 676–88

[26] Henikoff S and Henikoff J G 1992 Amino acid substitution matrices from protein blocks Proc. Natl Acad. Sci. 89 10915–19

[27] Kyte J and Doolittle R F 1982 A simple method for displaying the hydropathic character of a protein J. Mol. Biol. 157 105–32

[28] Worth C L, Preissner R and Blundell T L 2011 SDM—a server for predicting effects of mutations on protein stability and malfunction Nucleic Acids Res. 39 W215–22

[29] Kellogg E H, Leaver-Fay A and Baker D 2011 Role of conformational sampling in computing mutation-induced changes in protein structure and stability Proteins 79 830–8

[30] Kepp K P 2015 Towards a ‘Golden Standard’ for computing globin stability: stability and structure sensitivity of myoglobin mutants Biochim. Biophys. Acta 1854 1239–48

[31] Sali A and Blundell T L 1993 Comparative protein modelling by satisfaction of spatial restraints J. Mol. Biol. 234 779–815

[32] Simons K T, Ruczinski I, Kooperberg C, Fox B A, Bystroff C and Baker D 1999 Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins Proteins 34 82–95

[33] BlastClust 2000 (available at: https://ftp.ncbi.nih. gov/blast/documents/blastclust.html)

[34] Zimmermann L et al 2018 A completely reimplemented MPI bioinformatics toolkit with a new HHpred server at its core J. Mol. Biol. 430 2237–43

Table 1. Results on the IvankovDDGun test set and on Ivankov2000: the performance of ACDC-NN in learning DDGun3D was measured in terms of Pearson correlation coefficient (r) and root mean square error (RMSE). The antisymmetry property was assessed in terms of Pearson correlation coefficient (rd − i) and the bias (⟨δ⟩) between the predicted values. ⟨δ⟩ are expressed in kcal mol−1. IvankovDDGun (Test) is the test set extracted from the IvankovDDGun artificial dataset (never seen during training). Ivankov2000 consists of the original dataset whose ΔΔG values were derived from the predictions of DDGun3D and both direct and reverse 3D structures were evaluated.

	Performance	Antisymmetry	
Dataset	(r/RMSE)	r_{d-i}	$\langle\delta\rangle$
IvankovDDGun (test)	0.98/0.26	−1.0	0.01
Ivankov2000	0.97/0.27	−0.98	0.007

Table 2. Results on Ssym: the performance on direct and reverse variants was measured in terms of Pearson correlation coefficient (r) and root mean square error (RMSE). The antisymmetry was assessed using the correlation coefficient rd − i (6) and the bias ⟨δ⟩ (7). RMSE and ⟨δ⟩ are expressed in kcal mol−1 . All the predictions of the methods were taken from Pucci et al [21], except for INPS-NoSeqId, INPS, INPS3D, DDGun3D and ThermoNet. The results of DDGun3D, INPS and ThermoNet were taken from Montanucci et al [23], Fariselli et al [8] and from [18], respectively. ACDC-NN* reports the cross-validation performance using in input both structures, wild-type and mutated (as experimentally defined). Only Inps-NoSeqId, ThermoNet, ACDC-NN and ACDC-NN* were trained in cross-validation addressing the sequence identity issue (sequence similarity <25%).

	Pearson/RMSE		Antisymmetry	
Method	Direct	Reverse	r_{d-i}	$\langle\delta\rangle$
ACDC-NN*	**0.57/1.45**	**0.57/1.45**	**−1.00**	**0.00**
ACDC-NN	**0.58/1.42**	**0.55/1.47**	−0.99	**−0.01**
INPS-NoSeqId [22]	0.48/1.42	0.47/1.45	−0.99	−0.06
ThermoNet [18]	0.47/1.56	0.47/1.55	−0.96	−0.01
DDGun3D [23]	0.56/1.42	0.53/1.46	−0.99	−0.02
INPS [8]	0.51/1.42	0.50/1.44	−0.99	−0.04
INPS3D [16]	0.59/1.29	0.44/1.64	−0.86	−0.55
PopMusicSym [21]	0.48/1.58	0.48/1.62	−0.77	0.03
SDM [28]	0.51/1.74	0.32/2.28	−0.75	−0.32
CUPSAT [12]	0.39/1.71	0.05/2.88	−0.54	−0.72
Rosetta [29]	0.69/2.31	0.43/2.61	−0.41	−0.69
FoldX [17]	0.63/1.56	0.39/2.13	−0.38	−0.47
Maestro [10]	0.52/1.36	0.32/2.09	−0.34	−0.58
PoPMuSiC [7] v2.1	0.63/1.21	0.25/2.18	−0.29	−0.71
mCSM [14]	0.61/1.23	0.14/2.43	−0.26	−0.91
DUET [13]	0.63/1.20	0.13/2.38	−0.21	−0.84
AUTOMUTE [11]	0.73/1.07	−0.01/2.61	−0.06	−0.99
iStable [5]	0.72/1.10	−0.08/2.28	−0.05	−0.60
MUPRO [6]	0.79/0.94	0.07/2.51	−0.02	−0.97
I-Mutant [4] v3.0	0.62/1.23	−0.04/2.32	0.02	−0.68
NeEMO [9]	0.72/1.08	0.02/2.35	0.09	−0.60
STRUM [15]	0.75/1.05	−0.15/2.51	0.34	−0.87

Table 3. Results on p53: comparison between ACDC-NN and Thermonet, the other available deep-learning approach, on p53. ACDC-NN∗ values were obtained when both direct and reverse structures were used as input. The results of ThermoNet were taken from [18].

	Pearson/RMSE		Antisymmetry	
Method	Direct	Reverse	r_{d-i}	$\langle\delta\rangle$
ACDC-NN*	0.61/1.69	0.61/1.69	−1.00	0.00
ACDC-NN	0.62/1.67	0.61/1.72	−0.99	−0.01
ThermoNet	0.45/2.01	0.56/1.92	−0.93	−0.04

Table 4. Results on myoglobin: comparison between ACDC-NN and Thermonet, the other available deep-learning approach, on myoglobin. ACDC-NN∗ values were obtained when both direct and reverse structures were used as input. The results of ThermoNet were taken from [18].

	Pearson/RMSE		Antisymmetry	
Method	Direct	Reverse	r_{d-i}	$\langle\delta\rangle$
ACDC-NN*	0.58/0.89	0.58/0.89	−1.00	0.00
ACDC-NN	0.58/0.89	0.57/0.89	−0.99	−0.01
ThermoNet	0.38/1.16	0.37/1.18	−0.97	−0.02

Table 5. The optimal architectures of the differential net before and after transfer learning. The optimal parameters were selected on a validation set without intersections or homologies with the training set.

NN parameters	Before transfer-learning	After transfer-learning
Hidden units	32, 16	32, 16
Dropout	0.2	0.2
Epochs	40	150
Batch-size	100	150
Optimizer	Adam	Adam
Loss	logcosh + abs	logcosh + abs

Figure 1. (A) Performance of ACDC-NN in predicting ΔΔG for the direct mutations in p53 (r = 0.62, RMSE = 1.67 kcal mol−1). (B) Performance of ACDC-NN in predicting ΔΔG for the reverse mutations in p53 (r = 0.61, RMSE = 1.72 kcal mol−1). (C) Direct versus reverse ΔΔG values of all p53 mutations predicted by ACDC-NN (rd−i = −0.99 and ⟨δ⟩ = −0.01 kcal mol−1). (D) Performance of ACDC-NN in predicting ΔΔG for the direct mutations in myoglobin (r = 0.58, RMSE = 0.89 kcal mol−1). (E) Performance of ACDC-NN in predicting ΔΔG for the reverse mutations in myoglobin (r = 0.57, RMSE = 0.89 kcal mol−1). (F) Direct versus reverse ΔΔG values of all myoglobin mutations predicted by ACDC-NN, with a Pearson correlation of rd−i = −0.99 and ⟨δ⟩ = −0.01 kcal mol−1 .

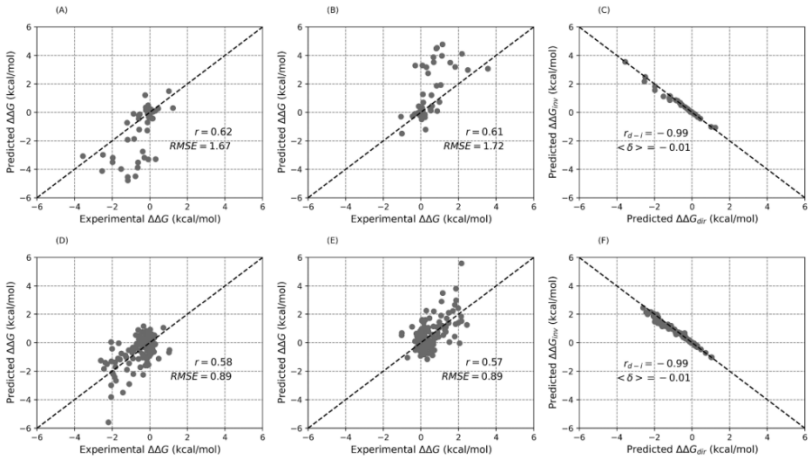

Figure 2. Heatmap of the filter matrix KS spanning over the sequence nearest neighbour of the variant residue. The rows indicate the filters while the columns represent the residue position in the input matrix. This matrix is significantly correlated with both substitution matrices and statistical potentials.

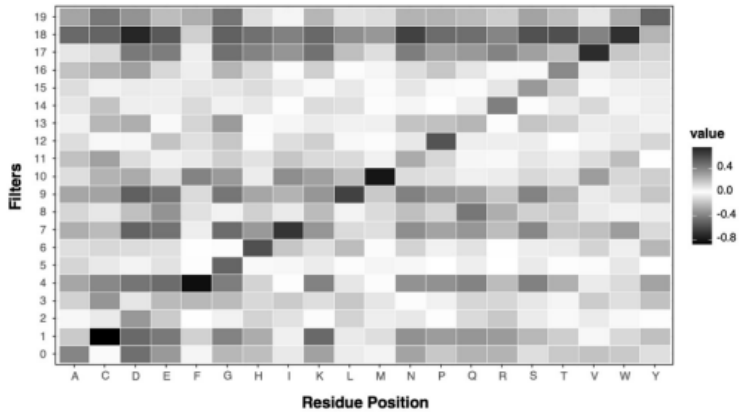

Figure 3. Heatmap of the filter matrix KT spanning over the three-dimensional neighbors of the variant residue. The rows indicate the filters while the columns represent the residue position in the input matrix. This matrix is significantly correlated with statistical contact potentials.

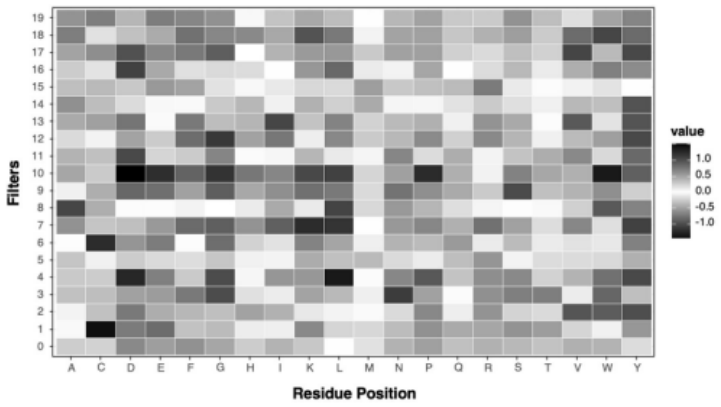

Figure 4. Constituent module of ACDC-NN. The module consists in a 2D-convolutional operation applied to both T (3D inputs) and S (1D inputs) using 20 filters with kernel (1, 20) and stride (1, 1); 3D information underwent 2D-global average pooling (GAP). Dot product was then applied to both T″ and S ′ processed information with the variation encoding vector V. Finally, all the 26 features were concatenated and used as input in a dense network. The notation used in the figure is the same as in the text.

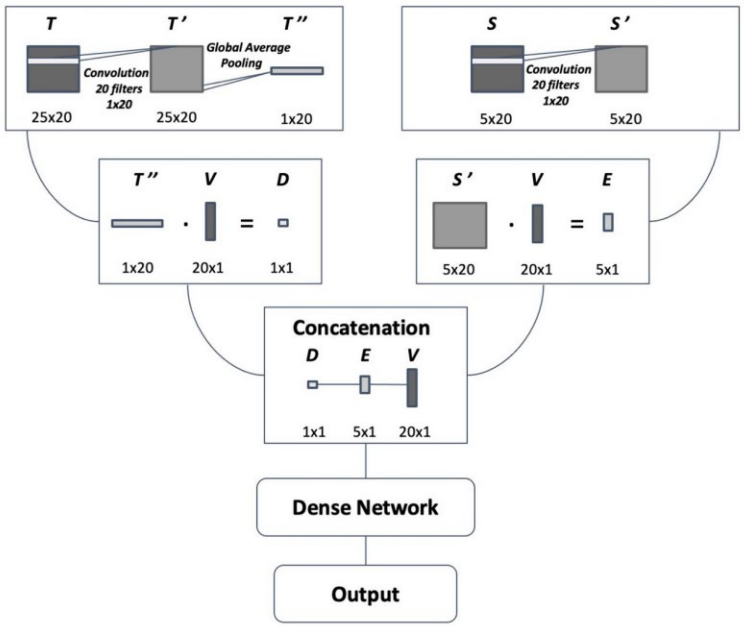

Figure 5. Complete ACDC-NN architecture. The module displayed in figure 4 was used for both direct and reverse variations. A final layer takes the average and the difference between the two outputs (representing the ΔΔG predictions for direct and reverse variations, respectively). The two Siamese networks have shared weights, both in the convolutional and dense parts of the network, represented by the dashed lines.