# Brief Communication

**ProSPr: Democratized Implementation of Alphafold Protein Distance Prediction Network**

Wendy M Billings[1], Bryce Hedelius[1], Todd Millecam[1], David Wingate[2], Dennis Della Corte[1*]

[1]Department of Physics and Astronomy, Brigham Young University, Utah
[2]Department of Computer Science, Brigham Young University, Utah

**Abstract**

Deep neural networks have recently enabled spectacular progress in predicting protein structures, as demonstrated by DeepMind's winning entry with Alphafold at the latest Critical Assessment of Structure Prediction competition (CASP13). The best protein prediction pipeline leverages intermolecular distance predictions to assemble a final protein model, but this distance prediction network has not been published. Here, we make a trained implementation of this network available to the broader scientific community. We also benchmark its predictive power in the related task of contact prediction against the CASP13 contact prediction winner TripletRes. Access to ProSPr will enable other labs to build on best in class protein distance predictions and to engineer superior protein reconstruction methods.

**Introduction**

Recently, a variety of powerful protein structure prediction methods, based on machine learning algorithms, have been reported.[1] Although direct prediction of structure from sequence has been attempted,[2] reproducible success is currently based on two-stage protocols.[3] The first stage is the training of a deep convolutional neural network (CNN) that predicts some macromolecular structure restraints like residue to residue distances, residue

contacts, dihedral angles or secondary structure assignments.[4] In a second stage, these restraints are used to construct a folded three-dimensional structure of the target protein. In the recent Critical Assessment of Structure Prediction (CASP13) a two stage folding protocol developed by DeepMind outperformed all established academic groups and predicted 25 of 43 protein structures with highest quality.[5] Unfortunately, DeepMind has not expressed a plan to publish the source code of their Alphafold protocol.

**Results & Discussion**

Here, we report the re-implementation of the first part of the Alphafold pipeline, an intramolecular distance prediction CNN, made freely available as source code (https://github.com/dellacortelab/prospr) and a Docker[6] container (see Methods). The CNN is in agreement with architectural details revealed by DeepMind at the December 2018 CASP13 conference (http://predictioncenter.org/casp13/doc/presentations/) and recently presented at a symposium at Washington University (https://www.youtube.com/watch?v=uQ1uVbrIv-Q); however, certain design decisions and hyperparameters were not shared in sufficient detail and required re-engineering. A graphical abstract of the CNN is given in Figure 1.

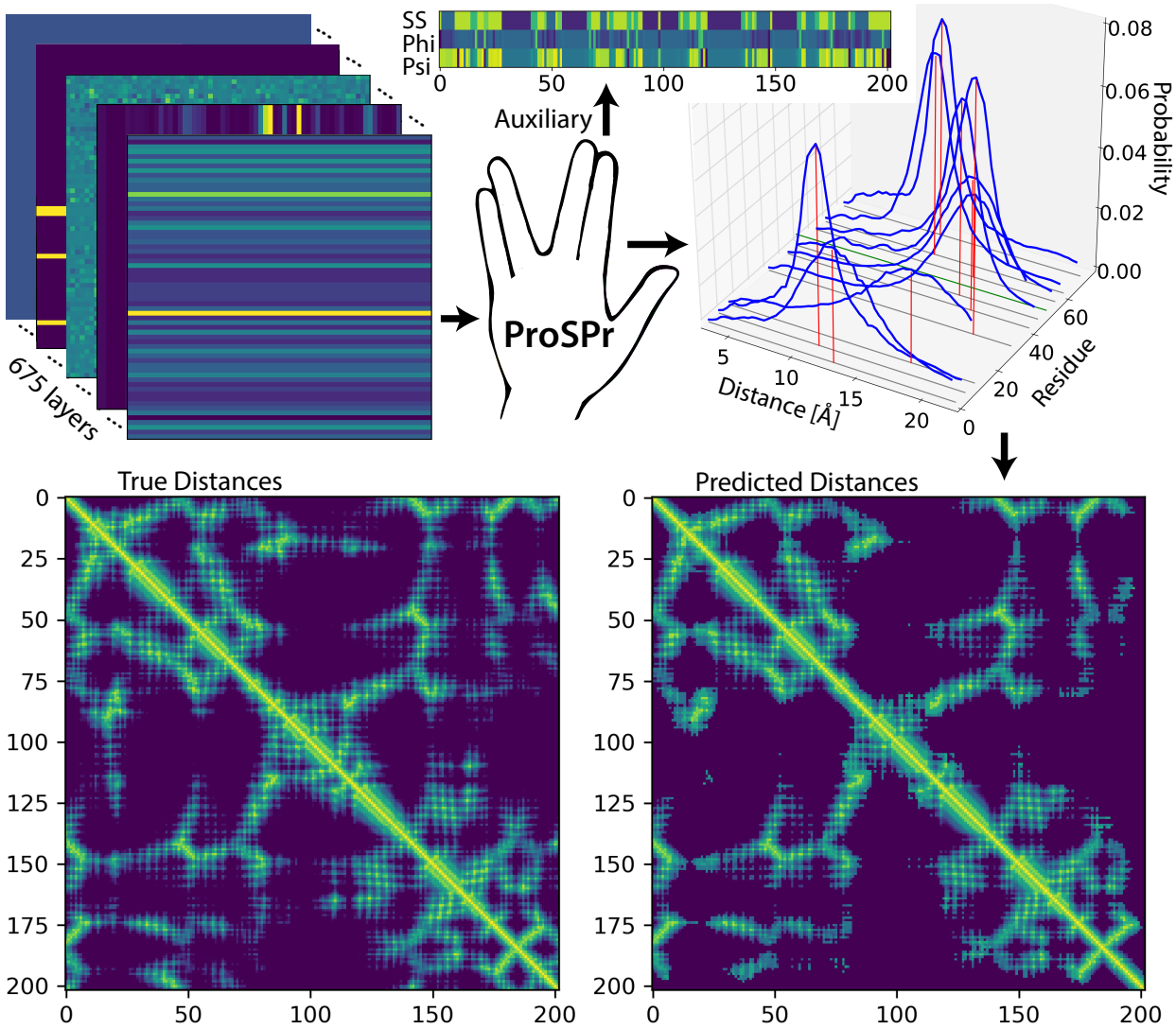


Figure 1 ProSPr distance prediction: Input sequence is converted into 675 layer input vector on top left. ProSPr CNN predicts as auxiliaries secondary structure elements for each residue from 9 DSSP classes (SS) and Phi/Psi Torsion angles between 0 and 360 degrees (top center). Further, it predicts distance distributions between each residue pair, shown as distance histogram for reside 50 of CASP Target T1016-D1 and selected residues (top right). The maxima of the distance distribution form a distance prediction map (heatmap, bottom right); left bottom, the real distances as measured in structure file of T1016-D1.

The CNN, named ProSPr (**Pro**tein **S**tructure **Pr**ediction), predicts the $C_\beta$-$C_\beta$ distance distributions between all amino acid residues ($C_\alpha$ for Glycine) in a given protein sequence. We trained three versions of ProSPr on sequences in the CATH S35 dataset[7] (Supplementary Note and Figure S1) with the same network architecture but different input vectors. ProSPr follows Alphafold

56 exactly and uses as input features the sequence information, the results of multiple sequence
57 alignments (MSA) computed with PSI-BLAST[8] and HHblits,[9] as well as a Potts model[10,11,12]
58 calculated from the MSA. ProSPr2 omits the Potts model, and ProSPr3 only uses the sequence
59 information as input.
60
61 The performance of these three models was tested on the CASP13 dataset for free and
62 template-based models. The predicted distance distributions were converted into contact
63 probabilities (distance between residues < 8 Å) and precision scores for three different classes
64 of contacts were calculated according to the CASP assessment protocol.[13] ProSPr precision
65 scores were directly compared to the performance of CASP13 winning CNN TripletRes[14] and are
66 shown in Figure 2 (Supplementary Table S1). Without being explicitly trained for this purpose,
67 ProSPr predicts contacts for 109 tested CASP13 domains with precision comparable to
68 TripletRes over all classes, as shown in Table 1. Table 1 shows precision scores for ProSPr
69 contacts with a maximum distance distribution < 8 Å, and for the full set of contacts
70 independent of distribution maxima. For high confidence predictions, with maximum <8 Å,
71 ProSPr is on average 2% better than TripletRes on the L/5 scores. The L/2 and L scores are not
72 directly comparable, because the absolute number of contacts ranked for ProSPr is
73 substantially lower if the maximum < 8 Å criterion is applied than the total number of possible
74 contacts ranked with TripletRes. For precision comparison the ranked probabilities of all
75 contacts, independent of maximum, are therefore also reported. Under these conditions, we
76 see that ProSPr is comparable to TripletRes, though on average slightly inferior. ProSPr2 results
77 are comparable to ProSPr short and medium length contact predictions but are inferior to

ProSPr long contact predictions. ProSPr3 is inferior to ProSPr in all categories. The performance of ProSPr2/3 was compared to TripletRes and is shown in Supplementary Figure S2. One issue with current precision reporting is that a smaller number of high confidence predictions leads to an inflation of L and L/2 scores, making model comparisons based on precision metric alone difficult to interpret. However, L/5 scores measure accurately the ability of a network to assign high confidence contacts and ProSPr outperforms TripleRes by an average of 2 %, which is in agreement with reports given by the Alphafold authors. (https://www.youtube.com/watch?v=uQ1uVbrIv-Q) Because ProSPr is trained to predict distances, the comparison against TripletRes only serves as a proof of concept. It would be a simple task to change the ProSPr network's final layers and to train it explicitly for contact predictions, which was not the scope of this work.

| | Short (\|i-j\| > 5 & \|i-j\| < 12) | | | Medium (\|i-j\| > 11 and \|i-j\| < 24) | | | Long (\|i-j\| > 23) | | |
|---|---|---|---|---|---|---|---|---|---|
| | L | L/2 | L/5 | L | L/2 | L/5 | L | L/2 | L/5 |
| TripletRes Average | 0.3001 | 0.4981 | 0.7276 | 0.3835 | 0.5787 | 0.7641 | 0.5308 | 0.6595 | 0.7627 |
| ProSPr Average | **0.6889** | **0.6938** | **0.7674** | **0.6680** | **0.6796** | **0.7657** | **0.6294** | **0.6972** | **0.7709** |
| ProSPr Full Average | 0.2980 | 0.4979 | 0.7428 | 0.3551 | 0.5511 | 0.7497 | 0.4969 | 0.6274 | 0.7436 |
| ProSPr2 Full Average | 0.2858 | 0.4639 | 0.6766 | 0.3378 | 0.4988 | 0.6746 | 0.3485 | 0.4459 | 0.5582 |
| ProSPr3 Full Average | 0.2064 | 0.2819 | 0.4018 | 0.2019 | 0.2587 | 0.3358 | 0.1287 | 0.1664 | 0.2176 |

Table 1: Average precision scores for TripletRes and different ProSPr models are compared. ProSPr Average only ranks contacts where the maximum of the distance probability distribution alls between 0-8 Å, all other ProSPr rows sort contacts by total probability to be between 0-8 Å.

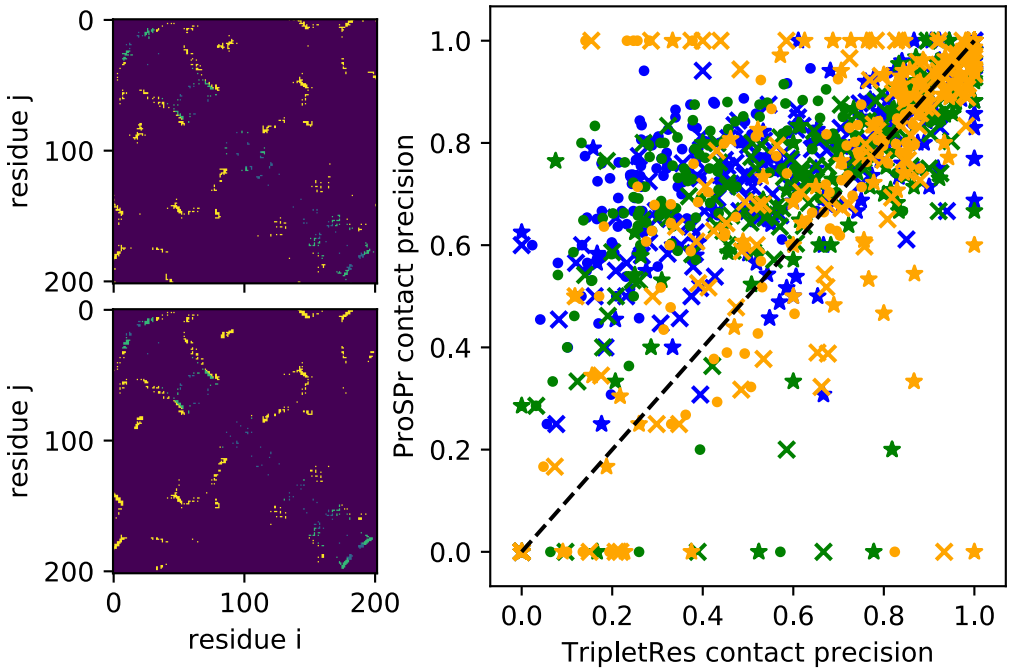


Figure 2: ProSPr distance predictions for 109 CASP13 domains were converted to contacts. Left panel shows an example contact label set for TR1016-D1 on top and the predicted contacts on the bottom. Right panel compares the precision for ProSPr contacts to those of TripletRes. Contacts are colored in blue, green, yellow for short, mid, and long. Markers circle, x, star correspond to L, L/2, L/5.

Next to the python-based source code a Docker[6] container of ProSPr is made available to enable rapid usage of the distance prediction protocol. The container includes input vectors for select CASP13 targets, three pretrained ProSPr models, and the distance prediction function to reproduce the results reported here. In addition, the distribution includes all dependencies necessary to produce a distance prediction for arbitrary sequences. Furthermore, the training set based on the CATH database, including the MSA and Potts models, is made available

107 (https://byu.box.com/v/ProteinStructurePrediction) to repeat the training outlined in the
108 methods section (approximately 2 TB of data). The GitHub repository contains a training
109 function that can be used to either improve a pretrained model, or to train a modified ProSPr
110 model for further optimization or ablation testing (full training on CATH dataset takes ~4 weeks
111 on single T100 GPU). The original Alphafold protocol ensembled distance predictions over 4
112 separately trained models and subtracted a reference network during CASP13. A pretrained
113 reference network is also provided that predicts distances only from sequence length and
114 whether each residue is glycine (Supplementary Note). With time, we will make additional
115 converged models of ProSPr and more comprehensive Docker containers available, to enable
116 model ensembling.
117
118 The field of protein structure prediction has to tackle the challenge of protein reconstruction
119 from geometric distance restraint distributions. During CASP13 it became apparent that
120 converting good distance predictions into chemically sound structures is still an unsolved
121 problem.[4] ProSPr lowers the entrance barrier for academic labs and enables the community to
122 quickly build on top of the internal coordinate predictions to develop improved protein
123 reconstruction protocols. Further, we anticipate applications of ProSPr to investigate validity of
124 evolutionary constraints as apparent from MSA, as ProSPr makes it possible to rapidly compare
125 the effects of many single mutations on protein distances. These insights might also enable
126 improved algorithms for in-silico drug discovery for mutated targets. In addition, we observed
127 that ProSPr can interpolate distances between missing residues (Supplementary Figure S3),

rendering it as a possible tool to support protein reconstruction from low resolution x-ray or cryo-EM data.[15]

In conclusion, we have demonstrated that ProSPr, a CNN based on the scarce details available for Alphafold, predicts residue-residue contacts with accuracy comparable to CASP13 winner TripletRes. ProSPr has the potential to propel protein structure prediction forward by democratizing the deep neural network and to empower directed evolution and protein reconstruction efforts.

## Methods

### Overview of ProSPr Architecture

Distance predictions within ProSPr can be made by calling distance prediction function, which consists of three steps as shown in Figure M1. Initially, a (L+32)x(L+32) profile is constructed for a sequence of length L using PSIBLAST, HHblits, a Potts model, and adding a frame of 32 bins as padding (Supplementary Note). Second, for a set 64x64 crops, defined by a stride parameter, of the profile an input vector with dimensions 675x64x64 is assembled. The input vector encodes the raw parameters, score, H parameters and Frobenius norm derived from the Potts model (total of 530 layers). Further, it contains two layers that hold the lists of residues for the crop, 42 layers for one-hot encoding of the sequence, 40 layers for a position specific substitution matrix (PSSM), 60 layers for the HHblits profile, and one layer for the sequence length. Third, the input layer is propagated through the CNN. After an initial batch norm, 1 dimensional convolution filters are applied to reshape the vector to a 128x64x64 matrix. This matrix is iterated 220 times through a residual network (RESNET) block that performs batch norming,

applies the exponential linear unit (ELU) activation function, projects down to 64x64x64 dimensions, applies again batch norming and ELU, and then cycles through 4 different dilation filters. The dilation filters have sizes 1,2,4, and 8 and are applied with a padding of the same size to retain dimensionality. After a final batch norm, the matrix is projected up to 128x64x64 and an identity addition is performed. After 220 iterations the final matrix is subject to two 1 dimensional convolutions that reshape it into the final distance and auxiliary predictions. The auxiliaries predict 8 classes of secondary structure as defined within the DSSP classifications, and the phi and psi dihedrals for each residue; the angles are binned with 10 degrees resolution between 0 and 360. Due to possible gaps in the sequence, an additional classification bin is introduced for each auxiliary prediction that represents unassignable information. The auxiliary predictions were only used for training but could yield additional insights in ProSPr applications.

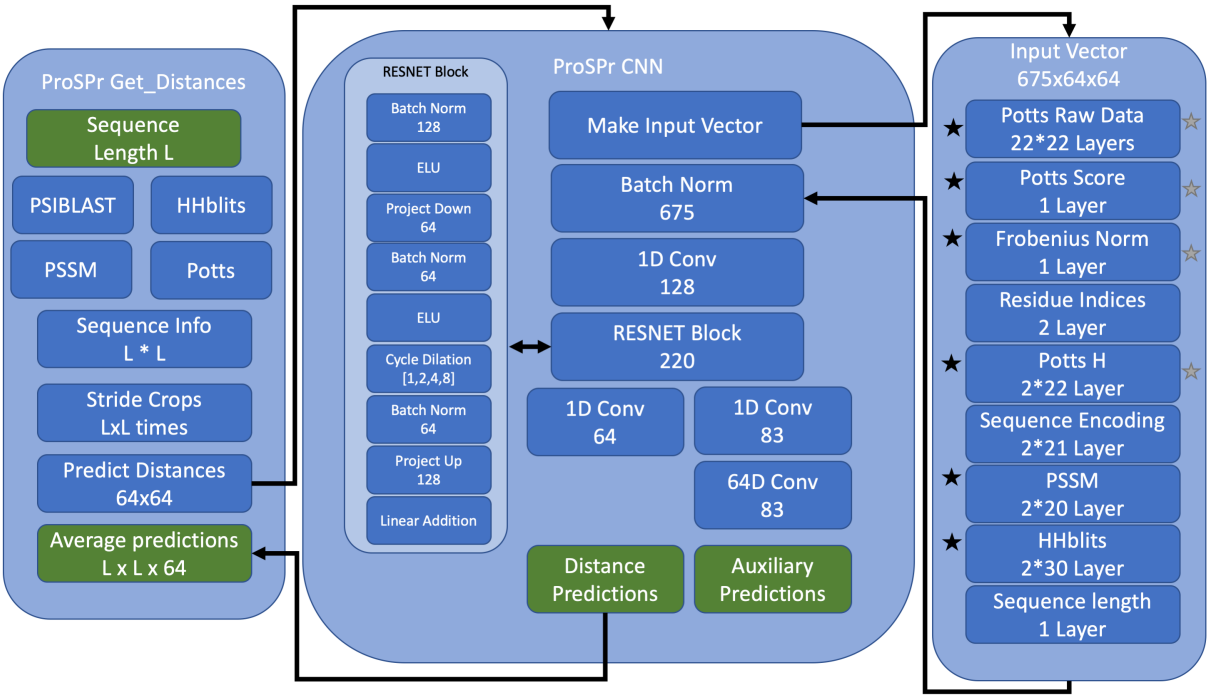


Figure M1: Overview of ProSPr core architectural components. On left the Get_Distances function, with inputs and outputs highlighted in green. In the center the ProSPr deep convolutional neural network with 220 RESNET blocks shown as an inlet. On the right a

169 breakdown of input features, which were all used for ProSPr, the layers marked with a grey or
170 black star were excluded in ProSPr2-3, respectively.

171

172 **Training of ProSPr**

173 ProSPr was trained on 64x64 crops extracted from the CATH S35 dataset[7] with 26393, 1000,
174 and 500 domains randomly selected as training, validation, and test sets, respectively
175 (Supplementary Note). Initial weights were assigned randomly with Pytorch, the loss was
176 calculated using cross entropy and an Adam optimizer with learning rate of 0.001 was used to
177 update the weights. Total loss was calculated as the weighted sum of ten times the distance
178 loss, the losses of two secondary structure assignments, and the losses of 4 torsion angles
179 assignments. Training loss and validation loss converged after 500,000 iterations with training
180 batch sizes of 8 (Supplementary Figure S1), which corresponds approximately to the number of
181 total crops necessary to visit each subdomain in the training set once. The training of ProSPr2
182 and ProSPr3 used the same setup, only the input vectors contained different amount of
183 information. For ProSPr2 all layers that contained Potts information were set to zero. For
184 ProSPr3 the PSSM and HHBlits layers were also set to zero. For these networks, the training loss
185 did not converge within 500,000 iterations (Supplementary Figure S1).

186

187 **Convert distances into contacts**

188 As a test, the distances for 109 CASP13 domains, which were not included in the training or
189 validation sets, were predicted and converted into contacts. Instead of using all possible 64x64
190 crops, a stride of 25 was used between the crops to speed up evaluation of large domains.
191 Average contact scores improved by 1% when a stride of 1 was used for the 44 shortest

domains. The 64x64x64 distance output encodes the probability of a residue i and j to have distances either not assignable (e.g. gap in sequence), in the range of 2.3 – 22 Å with .3 Å resolution between classes, or greater than 22 Å. If the maximum of the probability distribution fell between 2.3 – 8 Å (bins 1-19), we considered two residues in contact for the high confidence predictions. In all cases, contacts were ranked according to the sum probability of distances between 2.3 and 8 Å and the top L, L/2, L/5 (L is length of sequence) contacts were selected to calculate accuracy scores. The contacts were classified based on the sequence separation of residues *i* and *j* into: short-range ($6 \leq |i - j| \leq 11$), medium-range ($12 \leq |i - j| \leq 23$) and long-range ($|i - j| \geq 24$) contacts.

**Evaluation of contact accuracy**

According to CASP protocol, precision was calculated as follows:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositive}$$

The average in each category was calculated over 109 test domains from CASP13. For the comparison with TripletRes, the difference in average precision per category was again averaged.

## Installation instruction for Docker

To install ProSPr as a docker container and to see all currently available options enter in the command line (after installing docker):

*docker run prospr/prospr*

Yes, it is that easy!


## Acknowledgments

DDC expresses gratitude for computational resources offered by BYU Office of Research Computing.


## References


1 Xu, J. *Proc. Nat. Acad. Sc. U. S.* **116**, 16856-16865 (2019).
2 AlQuraishi, M. *Cell Syst.* **8**, 292-301. e293 (2019).
3 Greener, J. G., Kandathil, S. M. & Jones, D. T. *Nat. Commun.* **10**, 1-13 (2019).
4 Xu, J. & Wang, S. *bioRxiv*, 624460 (2019).
5 AlQuraishi, M. *Bioinformatics* (2019).
6 Boettiger, C. *Oper. Syst. Rev.* **49**, 71-79 (2015).
7 Knudsen, M. & Wiuf, C. *Hum. Genomics Proteomics* **4**, 207 (2010).
8 Altschul, S. F. *et al. Nucleic Acids Res.* **25**, 3389-3402 (1997).
9 Remmert, M., Biegert, A., Hauser, A. & Söding, J. *Nat. Methods.* **9**, 173 (2012).
10 Wu, F.-Y. *Rev. Mod. Phys.* **54**, 235 (1982).
11 Ekeberg, M., Hartonen, T. & Aurell, E. *J. Comput. Phys.* **276**, 341-356 (2014).
12 Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M. & Aurell, E. *Phys. Rev. E* **87**, 012707 (2013).
13 Schaarschmidt, J., Monastyrskyy, B., Kryshtafovych, A. & Bonvin, A. M. J. J. *Proteins* **86**, 51-66 (2018).
14 Li, Y., Zhang, C., Bell, E. W., Yu, D. J. & Zhang, Y. *Proteins* (2019).
15 Subramaniya, S. R. M. V., Terashi, G. & Kihara, D. *Nat. Methods.*, 1-7 (2019).