Protein folding quality assessment using 3D oriented convolutional neural networks: Supplementary Information

Guillaume Pagès[1,2,3], Benoit Charmettant[3], and Sergei Grudinin*[1,2,3]

[1]Univ. Grenoble Alpes, 38000 Grenoble, France
[2]CNRS, LJK, 38000 Grenoble, France
[3]Inria, 38000 Grenoble, France

1 Network topology

Table S1 shows the different layers used in the network. As the optimizer algorithm, we used the Adam optimizer [1], with a learning rate of 0.0001 and a beta parameter of 0.9.

2 Atom types

Table S2 lists all 167 atom types used in this work according to the PDB naming scheme.

3 Tests with other neural network configurations

In order to show the improvements brought by our neural network (NN) design, we made tests keeping the same architecture, modifying one element and retraining the network. Figure S1 and Tables S3-S4 summarize the results of different variants of Ornate. Here we list the nine tested architectures :

- The Base is the Ornate architecture as described in Table S1.
- The 4 Types architecture does not differentiate between the atoms of the same chemical elements in the input layer. There are thus only 4 types

*sergei.grudinin@inria.fr

of atoms : Carbon, Oxygen, Nitrogen and Sulfur. Therefore, the output dimensions in the input layer in Table S1 is 24 × 24 × 24 × 4.

- The 20 Types architecture uses the splitting of atoms in the initial layer into 20 groups adopted from protein docking studies [2, 3]. Therefore, the output dimensions in the input layer in Table S1 is 24 × 24 × 24 × 20.
- The CA Only architecture represents all the residues by their C_α atoms. We should note that the orientation of the local frames is kept according to the local topology, and all the C_α atoms have types that correspond to their amino acids. The output dimensions in the input layer in Table S1 is 24 × 24 × 24 × 20.
- The No Orientation architecture does not align the input meshes with the local topology of the backbone. Instead, they are aligned with x, y, z axes. The dimensionality of all the layers in Table S1 is kept unchanged.
- The + Secondary Structure architecture adds additional information about the secondary structure assignment of each residue at layer 15 of the network (see Table S1). The assignment was computed using the STRIDE algorithm [4].
- The + Surface Area architecture adds information about the solvent accessible surface area of each residue at layer 15 of the network (see Table S1). These areas were computed using the *nsc* method by Frank Eisenhaber [5, 6].
- The No Router architecture uses only one route for the fully connected part of the network.
- The Skip First Neighbor architecture does not map first residue neighbors of the considered residue on the mesh when constructing the input layer.

Figure S1 shows the progress of the training loss for all the tested architectures. This figure demonstrates a significant loss of training performance when using only C_α atoms, without using the local orientation of the input meshes, or without using the router architecture. Other modifications do not affect the training performance of the models significantly.

Tables S3 and S4 shed more light on the performance of the trained model on the test sets CASP 11 and CASP 12. We can clearly see that the model with only 4 atom types has a bad performance in both, the training loss, and the Pearson correlation tests. Also, we can see that adding information about secondary structure or surface area does not improve the training rate and neither helps the network to make better predictions. We believe that this information is redundant with the input and thus does not have impact on the performance of the model. Limiting the input to 20 predefined atom types affects the results only marginally, however they are slightly improved. This is clearly seen in Table S4, where the model with 20 predefined atom types has the smallest training loss on stage 2 test datasets. This suggests that the 20 chosen atom types are consistent with what the first retyper layer learns. We also believe that if we train the retyper layer for a longer time, we can always obtain better results than the predefined typization does. Finally, mapping only C_α atoms or skipping the first neighbors of the current residue has a negative effect on the training loss, as it is demonstrated for the test datasets in Table S4. Concerning the Pearson correlation between the predicted scores and the ground truth, Table S3 shows no significant difference in the performance of these two models compared to the base one, with slightly better results for the C_α model. It lets us envisage to use a coarse-grained representation of the input for a future version of this work. However, we will have to solve the problem of the slow training rate for such a model.

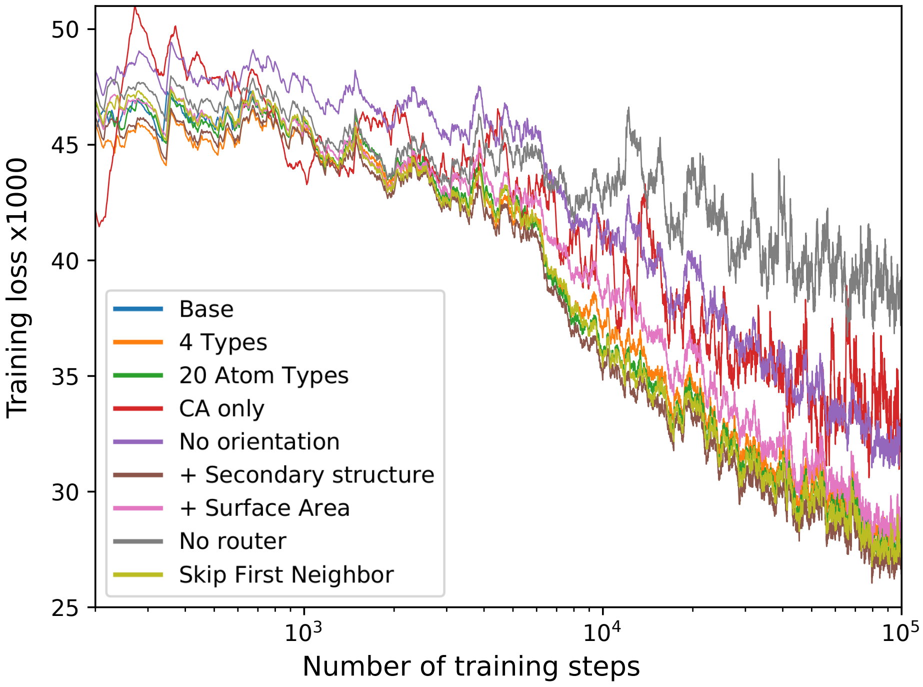

Figure S1: Variation of the training loss for multiple NN architectures during 100,000 training steps. Please see main text for more details.

References

[1] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[2] Sheng-You Huang and Xiaoqin Zou. An iterative knowledge-based scoring function for protein–protein recognition. *Proteins: Structure, Function, and Bioinformatics*, 72(2):557–579, 2008.

[3] Petr Popov and Sergei Grudinin. Knowledge of native protein–protein interfaces is sufficient to construct predictive models for the selection of binding candidates. *Journal of chemical information and modeling*, 55(10):2242–2255, 2015.

[4] Dmitrij Frishman and Patrick Argos. Knowledge-based protein secondary structure assignment. *Proteins: Structure, Function, and Bioinformatics*, 23(4):566–579, 1995.

[5] Frank Eisenhaber, Philip Lijnzaad, Patrick Argos, Chris Sander, and Michael Scharf. The double cubic lattice method: efficient approaches to numerical integration of surface area and volume and to dot surface contouring of molecular assemblies. *Journal of Computational Chemistry*, 16(3):273–284, 1995.

[6] Frank Eisenhaber and Patrick Argos. Improved strategy in analytic surface calculation for molecular systems: handling of singularities and computational efficiency. *Journal of Computational Chemistry*, 14(11):1272–1280, 1993.

Layer	Type	Input	Output dimensions	Settings	Parameters
0	Input	-	$24 \times 24 \times 24 \times 167$	-	-
1	3D Convolution (Retyper)	0	$24 \times 24 \times 24 \times 15$	Filter size $1 \times 1 \times 1$, stride 1	$15 \times 167 = 2505$
2	3D Convolution	1	$22 \times 22 \times 22 \times 20$	Filter size $3 \times 3 \times 3$, stride 1	$15 \times 20 \times 3 \times 3 \times 3 = 8100$
3	Add bias	2	$22 \times 22 \times 22 \times 20$	-	20
4	Batch Norm.	3	$22 \times 22 \times 22 \times 20$	-	-
5	Activation	4	$22 \times 22 \times 22 \times 20$	ELU	-
6	3D Convolution	5	$19 \times 19 \times 19 \times 30$	Filter size $4 \times 4 \times 4$, stride 1	$20 \times 30 \times 4 \times 4 \times 4 = 38400$
7	Add bias	6	$19 \times 19 \times 19 \times 30$	-	30
8	Batch Norm.	7	$19 \times 19 \times 19 \times 30$	-	-
9	Activation	8	$19 \times 19 \times 19 \times 30$	ELU	-
10	3D Convolution	9	$16 \times 16 \times 16 \times 20$	Filter size $4 \times 4 \times 4$ stride 1	$30 \times 20 \times 4 \times 4 \times 4 = 38400$
11	Add bias	10	$16 \times 16 \times 16 \times 20$	-	20
12	Batch Norm.	11	$16 \times 16 \times 16 \times 20$	-	-
13	Activation	12	$16 \times 16 \times 16 \times 20$	ELU	-
14	Average Pool	13	$4 \times 4 \times 4 \times 20$	Filter size $4 \times 4 \times 4$, stride 4	-
15	Reshape	14	1280	-	-
16	Linear	15	640	-	$1280 \times 640 = 819200$
17	Add bias	16	640	-	640
18	Activation	17	640	ELU	-
19	Linear	18	20	-	$640 \times 20 = 12800$
20	Add bias	19	20	-	20
21	Activation	20	20	Softmax	-
22	Linear	14	20×64	-	$1280 \times 20 \times 64 = 1638400$
23	Add bias	22	20×64	-	$20 \times 64 = 1280$
24	Activation	23	20×64	ELU	-
25	Linear	24	20×1	-	$20 \times 64 = 1280$
26	Multiply	24,21	20	-	-
27	Sum	26	1		–
28	Activation	27	1	$1/2 + \tanh$	-

Table S1: Detailed topology of the scoring neural network. The network contains a total of 2,595,565 adjustable parameters.

Residue	Atoms
ALA	C N O CA CB
ARG	C N O CA CB CG CD NE CZ NH1 NH2
ASN	C N O CA CB CG ND2 OD1
ASP	C N O CA CB CG OD1 OD2
CYS	C N O CA CB SG
GLN	C N O CA CB CG CD NE2 OE1
GLU	C N O CA CB CG CD OE1 OE2
GLY	C N O CA
HIS	C N O CA CB CG CD2 ND1 CE1 NE2
ILE	C N O CA CB CG1 CG2 CD1
LEU	C N O CA CB CG CD1 CD2
LYS	C N O CA CB CG CD CE NZ
MET	C N O CA CB CG SD CE
PHE	C N O CA CB CG CD1 CD2 CE1 CE2 CZ
PRO	C N O CA CB CG CD
SER	C N O CA CB OG
THR	C N O CA CB CG2 OG1
TRP	C N O CA CB CG CD1 CD2 CE2 CE3 NE1 CZ2 CZ3 CH2
TYR	C N O CA CB CG CD1 CD2 CE1 CE2 CZ OH
VAL	C N O CA CB CG1 CG2

Table S2: List of 167 atom types used by our network. The left column lists the 20 standard protein residue types. The right column lists the corresponding atoms using the PDB naming scheme.

Model	CASP11		CASP12	
	Stage 1	Stage 2	Stage 1	Stage 2
Base	0.67	0.62	0.70	0.79
4 Atom Types	0.62	0.60	0.67	0.75
20 Atom Types	0.66	0.62	0.70	0.78
CA Only	0.68	0.63	0.70	0.79
No Orientation	0.65	0.58	0.69	0.75
+ Secondary Structure	0.65	0.62	0.68	0.79
+ Surface Area	0.69	0.63	0.70	0.79
No Router	0.57	0.54	0.59	0.68
Skip First Neighbor	0.67	0.62	0.73	0.80
95 % Confidence interval	0.03	0.01	0.04	0.02

Table S3: Pearson's correlation between scores obtained after retraining the network with a modification in the architecture. Pearson's correlation has been chosen because it has a narrow confidence interval (unlike prediction loss) and gives similar results as other correlations.

Model	CASP11		CASP12	
	Stage 1	Stage 2	Stage 1	Stage 2
Base	39.47	38.39	38.78	36.04
4 Types	41.26	38.39	41.80	36.72
20 Types	38.76	37.80	37.72	35.52
CA Only	49.04	43.14	52.49	42.45
No Orientation	45.55	41.09	46.11	39.46
+ Secondary Structure	40.06	38.91	39.96	36.77
+ Surface Area	40.26	38.27	40.38	36.65
No Router	37.23	39.84	35.04	36.94
Skip First Neighbor	39.51	39.52	38.25	36.82
95 % Confidence interval	0.08	0.03	0.10	0.04

Table S4: Training loss measured on the test datasets, after retraining the network with a modification in the architecture.