Bioinformatics, 35(18), 2019, 3313–3319
doi: 10.1093/bioinformatics/btz122
Advance Access Publication Date: 19 February 2019
Original Paper

Structural bioinformatics

Protein model quality assessment using 3D oriented convolutional neural networks

Guillaume Pagès, Benoit Charmettant and Sergei Grudinin*

Université Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, Grenoble 38000, France

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on October 4, 2018; revised on January 17, 2019; editorial decision on February 9, 2019; accepted on February 13, 2019

Abstract

Motivation: Protein model quality assessment (QA) is a crucial and yet open problem in structural bioinformatics. The current best methods for single-model QA typically combine results from different approaches, each based on different input features constructed by experts in the field. Then, the prediction model is trained using a machine-learning algorithm. Recently, with the development of convolutional neural networks (CNN), the training paradigm has changed. In computer vision, the expert-developed features have been significantly overpassed by automatically trained convolutional filters. This motivated us to apply a three-dimensional (3D) CNN to the problem of protein model QA.

Results: We developed Ornate (Oriented Routed Neural network with Automatic Typing)—a novel method for single-model QA. Ornate is a residue-wise scoring function that takes as input 3D density maps. It predicts the local (residue-wise) and the global model quality through a deep 3D CNN. Specifically, Ornate aligns the input density map, corresponding to each residue and its neighborhood, with the backbone topology of this residue. This circumvents the problem of ambiguous orientations of the initial models. Also, Ornate includes automatic identification of atom types and dynamic routing of the data in the network. Established benchmarks (CASP 11 and CASP 12) demonstrate the state-of-the-art performance of our approach among single-model QA methods.

Availability and implementation: The method is available at https://team.inria.fr/nano-d/software/Ornate/. It consists of a C++ executable that transforms molecular structures into volumetric density maps, and a Python code based on the TensorFlow framework for applying the Ornate model to these maps.

Contact: sergei.grudinin@inria.fr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Proteins are ubiquitous for virtually all biological processes. Identifying their role helps to understand and potentially control these processes. However, even though protein sequence determination is now a routine procedure, it is often very difficult to use this information to extract relevant functional knowledge about system under study. Indeed, the function of a protein relies on a combination of its chemical and mechanical properties, which are defined by its structure.

Identifying protein structure from its sequence is thus a very important, though a challenging task. Experimental structure identification is not possible in all of the cases, and is generally very tedious and expensive. Therefore, computational methods that try to predict protein structure from its sequence have emerged in the past. Most of these methods combine the sampling of protein conformations step with the model quality assessment (QA) step. The former generates protein conformations, while the latter scores these to select the ones that will be as close as possible to the native structure.

In this work, we only address the second problem and propose a novel method for protein model QA. This problem is challenging as it is shown by the fact that the Critical Assessment of protein Structure Prediction (CASP) community experiment (Moult *et al.*, 2018) has an

entire category dedicated to this specific topic (Cozzetto *et al.*, 2007). Indeed, the folding of a protein to its native conformation is driven by thermodynamic laws. This process can be formally characterized by the changes in free energy, which includes both enthalpic and entropic contributions. The former is defined by the potential energy contributions, while the latter describes the shape of the potential energy landscape. A proper estimation of the free energy differences is a very difficult and computationally expensive task, as it generally assumes knowledge about the protein environment, which is rarely available, and includes high-dimensional sampling.

Many methods for protein folding QA have already been developed. The goal of these methods is to predict the folding quality of a protein structure receiving its three-dimensional (3D) model as input. Generally, QA methods can be split into several classes. The best performing methods are often the consensus-based ones. This means that they do not score one single model but a whole set of them by comparing the models to each other. This class of methods is represented by Pcons (Lundström *et al.*, 2001) or 3D-Jury (Ginalski *et al.*, 2003), for example. These methods are among the best performers on various benchmarks, but they suffer from the fact that one model cannot be scored alone and its score depends on the quality of other models in the scoring set. The methods that do not use consensus are called single-model methods. Among the single-model methods, one can distinguish simple methods such as VoroMQA (Olechnovič and Venclovas, 2017) or RWplus (Zhang and Zhang, 2010), which rely on a single type of structural features (contact area or pairwise atomic distances, respectively). Composite methods such as SBROD (Karasikov *et al.*, 2019) aggregate many types of heterogeneous structural features. Meta-methods such as QProb (Cao and Cheng, 2016) or DeepQA (Cao *et al.*, 2016) integrate results from different methods to obtain better results. The boundaries between these categories are not always clear as some methods like Proq3D (Uziela *et al.*, 2017) aggregate both structural features and Rosetta energy terms (Leaver-Fay *et al.*, 2011).

The advent of machine-learning techniques together with the growing amount of known 3D protein structures have broaden our possibilities to construct novel model QA methods. Specifically, convolutional neural networks (CNN, also sometimes referred to as deep learning) have demonstrated outstanding capacities for learning hierarchical representations (Lee *et al.*, 2009). Very recently, 3D CNN has been applied to prediction of protein binding sites and also their interactions with ligands (Ragoza *et al.*, 2017; Jiménez *et al.*, 2017; Townshend *et al.*, 2018; Wallach *et al.*, 2015) and also protein QA tasks (Derevyanko *et al.*, 2018). However, one of the major hurdles for the success of 3D CNNs in this topic has been the uncertainty of choosing the reference orientation for the structures in the training and the test sets. For example, to circumvent this problem, Derevyanko *et al.* (2018) had to significantly augment the training set with random orientation of the input structures in 3D.

This work reports on a significant improvement of the 3D CNN method applied to the protein QA task, mostly due to the found solution to the orientation problem of the input 3D data. To do so, first, we decompose the global QA scoring task into a set of residue-wise local scoring tasks. First, the structure is not scored as a whole, but each residue is scored individually. Residue-wise scoring has been widely used, from the early 1990s (Lüthy *et al.*, 1992; Sippl, 1993) to the current QA scoring methods, such as VoroMQA (Olechnovič and Venclovas, 2017) and Proq3D (Uziela *et al.*, 2017). Second, each local scoring task is handled by a residue-wise CNN with the input data oriented according to the local backbone topology. Other improvements of our 3D CNN model include automatic identification of atom types and dynamic routing of the data in the network. The performance of our model in the model QA task surpasses other single-model methods that rely only on the structure of the model. It is also very close to the performance of composite meta-methods that also use sequence alignment and evolutionary information as additional features (Ray *et al.*, 2012).

2 Materials and methods

2.1 Residue-wise scoring

Our model is called Ornate, which stands for Oriented Routed Neural network with Automatic Typing. It relies on predicting local quality measures for each residue in a protein, provided a density map of its neighborhood. Residue-wise scoring brings several technical advantages when using 3D CNN. First, one protein structure contains many residues that provide multiple 3D examples to be used in training. Since the predicted score is local, the network can specifically learn local favored or undesirable 3D geometries. This would not be possible by predicting the global score. Second, CNN traditionally use a fixed input size. However, there are orders of magnitudes between the sizes of the smallest and the biggest proteins. Thus, choosing a fixed input size for the whole protein implies either constructing an oversized network, which will be costly to train and to run, or to be limited by the size of the structure to score. We should also notice that scaling the input structure to the input size of the network is very undesirable in our case, since we expect our network to learn some features that are not scale invariant (e.g. hydrogen bonds or secondary structure). Finally, Ornate naturally provides a score per protein residue. This can be valuable for certain applications, where local scores are required. To obtain the global score for the overall model QA task, one simply needs to combine all the predicted local scores. This can be done in multiple ways. The score given to a structure by Ornate is the mean of the scores given to each of its residues.

There are, generally, multiple options for ground-truth local quality measures (Olechnovič *et al.*, 2018). For our purposes, we required a relatively fast residue-wise measure. Therefore, we considered all-atom (a.a.) CAD score (Olechnovič *et al.*, 2013) and LDDT (Mariani *et al.*, 2013), and have chosen the former. This choice was motivated by the smoothness of CAD score, and the absence of an arbitrary score threshold, even though the two scores are very correlated (Olechnovič *et al.*, 2018).

2.2 Input

For the estimation of the residue-wise score, Ornate is trained on cubic volumetric maps of side 19.2 Å centered and oriented on the given residue. This way, each map represents a certain residue with its spatial neighborhood. Figure 1 shows an example of the protein volume captured by such a map.

2.2.1 Input orientation

Formally, the nth residue with its neighborhood is represented by a cubic map positioned and oriented according the positions of its backbone atoms. Construction of the local frames has already been used for other applications, e.g. structural alignment of proteins (Ritchie *et al.*, 2012). The $\vec{x}$ direction of the map coincides with the vector pointing from the carbon of the previous residue (C_{n-1}) to the nitrogen of the current residue (N_n). For the first residue, we define $\vec{x}$ with the vector pointing from the alpha carbon ($C\alpha_n$) of the current residue to C_n. The $\vec{y}$ direction of the map is perpendicular to $\vec{x}$ and is defined such that $C\alpha_n$ lies in the half-plane $O\vec{x}\vec{y}$ with $y > 0$ (see Fig. 2A). Finally, $\vec{z}$ is defined as a vector product, $\vec{z} = \vec{x} \times \vec{y}$.

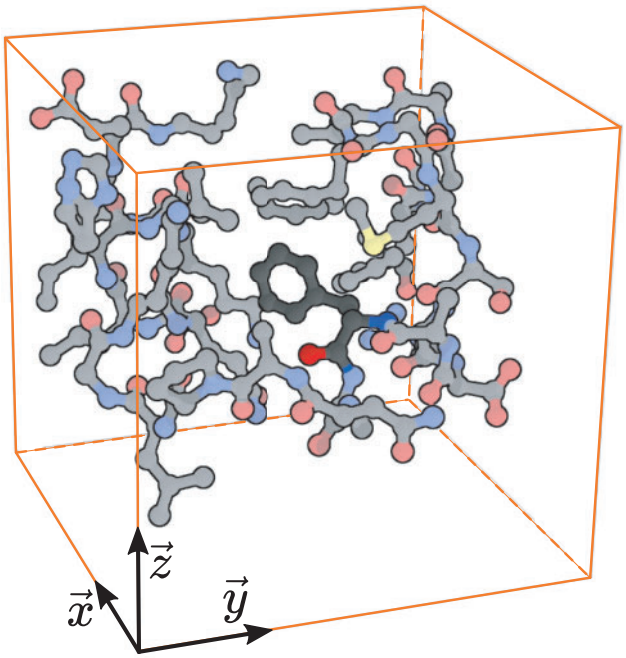

Fig. 1. Example of the volumetric input corresponding to one protein residue (here, Phe58 from the 1yrf structure). The atoms of the considered residue are shown in dark colors and the atoms of his neighborhood are shown in light colors. The orange box shows the boundaries of the considered neighborhood. Only the atoms within this neighborhood are shown

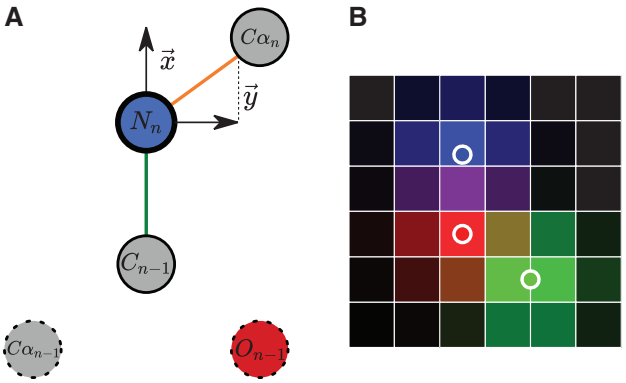

Fig. 2. (**A**) Illustration of the definition of the local frame for the *n*th residue and the atoms positionally constrained by this definition. The local frame is defined by the position of N_n, and the directions of C_{n-1} and $C\alpha_n$ with respect to N_n. The N_n atom is shown with a bold outline, and its position is fixed in the local frame. Thin outlines correspond to the atoms whose positions are partially constrained in space. Dashed outlines correspond to the atoms whose positions are unconstrained, but in practice do not vary much from the mean values. (**B**) Example of three atoms projected on a map with the presented method. For clarity, we only show 6×6 voxels of a 2D slice of the map. Three atoms of different atom types are represented by three circles of different colors. The figure is scaled so that the side of one voxel is 0.8 Å, and the inter-atomic distance is 1.4 Å, which is a typical bond length for heavy atoms in proteins

Once the three basis vectors are defined, we specify the origin of the map such that N_n is located at (6.1 Å, 6.6 Å, 9.6 Å) with respect to the map origin. This position has been chosen empirically such that all the atoms of all the residues among tested proteins fit in the map. This way, by definition, the position of N_n in the local frame is always the same, C_{n-1} is constrained along the $\vec{x}$ axis and $C\alpha_n$ is constrained in the $\vec{x}\vec{y}$ plane. In addition to these construction restraints, positional variance of other backbone atoms is also significantly reduced thanks to the fact that the values of the bond length and bond angles do not vary much. Also, a double bond between N_n and C_{n-1} forces the alpha carbon and the oxygen of the previous residue ($C\alpha_{n-1}$ and O_{n-1}) to lie in the same plane as N_n, C_{n-1} and $C\alpha_n$. We specifically designed the local frame to keep constant the positions of as many atoms as possible. Indeed, we believe that having some invariant patterns allows the CNN to learn input structure better and faster, similarly to a situation with a human's brain, which recognizes characters from a picture more easily if the picture is oriented correctly. By explicitly defining the origin and orientation of the input map with respect to the backbone atoms of the residue, we do not need the network to be rotationally invariant (Worrall *et al.*, 2017) and no data augmentation by rotating or translating the input is required (Van Dyk and Meng, 2001).

2.2.2 Input values

The input maps are constructed from the atomic representation of the position of the current residue and its neighboring atoms. The atomic representation of the structure is first transformed to a density function, then projected on a grid to obtain the map input for our CNN. The density function associates each point in space with a vector of 167 dimensions. These 167 dimensions correspond to the 167 different atom types that can be found in amino acids (without the hydrogens). A list of these atom types is given in Supplementary Material. More formally, let $\vec{a_i}$ be the position of the *i*th atom of the structure, σ be the width of the Gaussian kernel (we use $\sigma = 1$ Å) and t_i be the 167-dimensional unit vector whose only nonzero component corresponds to the type of the *i*th atom. The density function d at a point $\vec{v}$ then reads as

$$d(\vec{v}) = \sum_{i \leq N_{atoms}} \exp\left[-\left(\frac{\vec{v} - \vec{a_i}}{\sigma}\right)^2\right] t_i. \quad (1)$$

To project the density on a map, we split the map in $24 \times 24 \times 24$ voxels of side 0.8 Å, and assign to each voxel the value of the density function at its center. Figure 2B shows an example of the projection made with three atoms with different atom types, represented by red, green and blue colors. To reduce memory footprint, we store each component of a voxel value in one byte of data as a fixed-point number with a scaling value of 1/255. Thus, the map for one residue requires $24 * 24 * 24 * 167 = 2.3$ MB of memory storage. As a consequence, a density smaller than 1/255 will be regarded as zero. This naturally truncates to zero values of the Gaussian kernel with arguments larger than $\sqrt{\ln(255)}\sigma = 2.35$Å. It is thus possible not to consider atoms that are more distant than 2.35 Å from the map, and, with an appropriate neighbor search algorithm, to keep a linear complexity of the mapping algorithm with respect to the number of residues in the protein.

2.3 Network topology

The CNN architecture used in this work is inspired by CNNs from computer vision. Figure 3 summarizes the network's topology. A typical CNN design begins with convolutional layers that deal with high dimensions of the input spatially structured data, followed by fully connected layers after the dimensionality of the data has been reduced. In our design, the three convolutional layers learn high-level features while progressively coarsen them and reducing the data's dimensionality. Then, a set of fully connected layers combines these features and outputs a prediction. As the activation function,

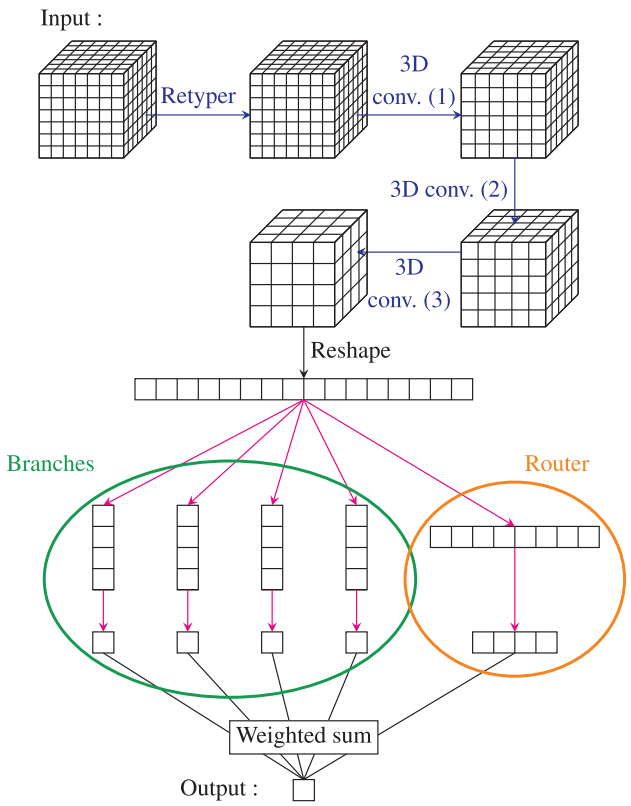

Fig. 3. Summary of the network topology. The blue arrows represent 3D convolutional layers and the red arrows represent fully connected layers. A more complete description is given in Supplementary Material

we used ELU (Clevert *et al.*, 2016), which has been proven to speed up learning in deep neural networks and lead to higher classification accuracies. We also include batch normalization layers (Ioffe and Szegedy, 2015) that accelerate the training and improve the accuracy of the network's predictions. Finally, we added two additional layers, a 'retyper' and a 'router' designed for specific purposes, which are explained below.

2.3.1 The retyper layer

A very original and uncommon pattern of the current network is the presence of the first layer that we called 'retyper'. This layer, which is technically a convolutional layer of size $1 \times 1 \times 1$ with 167 input channels and 15 output channels, projects each of the 167 atom types that exist in proteins to a feature space of dimension 15. By doing so, we reduce the dimensionality of the input by a factor of 11, and switch from a sparse data representation to a dense one. Indeed, a voxel with n nonzero components implies n atoms of different types located at a distance smaller than 2.35 Å from the voxel center. In practice, there can be at most a dozen of nonzero components in one voxel.

2.3.2 Router layer

After the convolutional layers, we apply a data routing layer. Our initial idea was to explicitly allow the combination of the features to be different depending on the residue type provided in input. We separately trained 20 different routes as a second part of our network, which were specific to each type of amino acids. However, to do so, we needed 20 times more training steps because only one route was trained at a time. In practice, some routes should require even more training steps, since the amino acid distribution is not even in proteins. Altogether, the gain from having a different model for each route was not worth the additional training.

As a second attempt, we later changed our network architecture to let the network learn the data routing as proposed in Ioannou *et al.* (2016). The idea here was to have a network called 'router' that predicts which route should be trusted to score these particular data. In this implementation, the data outputted by the convolutional layers are sent to every route and the final score is an average of the different outputs, weighted by the router predictions. The advantage compared to the previous technique is that the router can learn more relevant criteria than just the residue type to choose which route to select.

2.3.3 Topology design

To assess the importance of each of these specific designs, we trained several different models removing or adding additional features one by one. The results for nine different architectures are given in Supplementary Material. We measured the training rates, the training loss and the Person's correlation for the nine networks on the test datasets. In particular, we can see that the performance drops significantly if only four atom types are used, without using the local orientation of the input meshes, or without using the router architecture.

2.4 Training loss function

We chose to train Ornate to approximate the value of CAD score (Olechnovič *et al.*, 2013) of each residue. As a result, we set the training loss function for a residue r_i scored $s(r_i)$ by our network as:

$$\text{Training loss} = (s(r_i) - \text{CAD score}(r_i))^2 \tag{2}$$

The training loss is thus simply the squared difference between the prediction and the ground truth for each residue.

2.5 Training phase

As the training set, we used the server submissions for CASP 7, 8, 9, 10 stage 2 experiments. We also removed few structures whose CAD scores were equal to zero, or whose backbones were incomplete. We trained Ornate with 100 000 optimization steps using a stochastic gradient descent method (see Supplementary Material for more details). Each step optimizes the network on 40 consecutive residues from an input structure. When a structure is running out of residue, a new one is randomly selected. Thus, we used a total of 4 M residues for optimizing the network. This represent less than 20 000 structures, while stages 2 of CASP 7, 8, 9, 10 contain each about 10 000 models. Each structure is thus used at most once, meaning that the values of the training loss function were always computed for new structures.

Figure 4 shows the training loss during each step of the training. Since this loss is very fluctuant, we also plot a smoothed version of the loss defined at each training step n as:

$$\begin{aligned}\text{Training loss}_s(n) = &\frac{1}{1000}\text{Training loss}(n) \\ &+ \frac{999}{1000}\text{Training loss}_s(n-1)\end{aligned} \tag{3}$$

This smoothed version seems to reach a plateau after about 80 000 steps so we decided to stop the training at this point. The overall decrease of loss cannot be due to over-fitting, since each step was trained on new data.

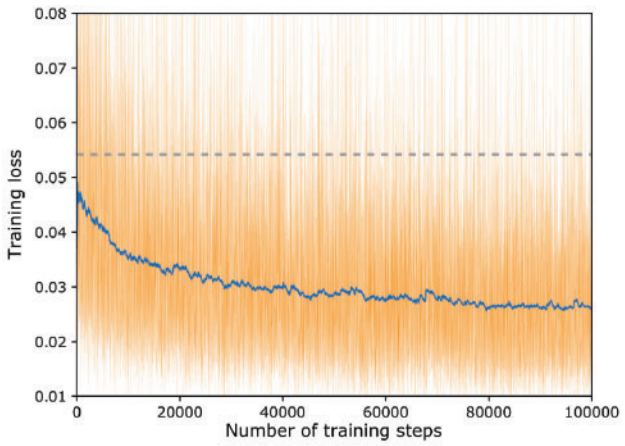

Fig. 4. Variation of the loss during 100 000 training steps. The training loss is shown with the thin line and the smoothed loss—with the thick line. The dashed line shows the variance of CAD score on the training set. It equals to the expected value of the training loss for a scoring scheme that always returns the average CAD score of the training set

3 Results and discussion

3.1 Comparison with the state of the art

We compared the results of our scheme with several other state-of-the-art QA methods. To do so, we used the same benchmark as Karasikov *et al.* (2019) and Cao and Cheng (2016). For a rigorous comparison, we trained Ornate with the data from CASP 7–10 server submissions, and blindly scored protein models from CASP 11 and CASP 12 server submissions, stages 1 and 2. Formally, for each target of CASP 11, and CASP 12, we have a model set $M = \{P_1, ..., P_n\}$ and a native structure P_0. We computed estimators of the performance of QA methods Q with respect to the ground-truth measure G, where $G(P)$ measures the similarity between the model P and native structure P_0. The prediction loss (PL) is defined as:

$$\mathrm{PL}(M) = \max_{P \in M} G(P) - G(\arg\max_{P \in M} Q(P)). \tag{4}$$

Let us define the average of a function F over a set $M = \{P_1, ..., P_n\}$ as

$$\langle F(M) \rangle = \frac{1}{n} \sum_{i=1}^{n} F(P_i). \tag{5}$$

Pearson's r is defined as

$$r_{G,Q}(M) = \frac{\langle GQ(M) \rangle - \langle G(M) \rangle \langle Q(M) \rangle}{\sqrt{\left(\langle G^2(M) \rangle - \langle G(M) \rangle^2 \right) \left(\langle Q^2(M) \rangle - \langle Q(M) \rangle^2 \right)}}. \tag{6}$$

Spearman's ρ measures the correlation between the rankings given by two functions

$$\rho_{G,Q}(M) = r_{\mathrm{rg}_G, \mathrm{rg}_Q}(M). \tag{7}$$

Kendall's τ is defined by

$$\tau_{G,Q}(M) = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{n(n-1)/2}, \tag{8}$$

where a pair $\{P_i, P_j\}$ is concordant if $(G(P_i) - G(P_j))(Q(P_i) - Q(P_j)) > 0$ and discordant if $(G(P_i) - G(P_j))(Q(P_i) - Q(P_j)) < 0$.

The average losses, Pearson's r, Spearman's ρ and Kendall's τ are an average of these indicators computed for each target. They estimate how well the scoring function can compare structures with the same sequence, and how well it picks the best among them. In addition, we computed global Pearson's r, Spearman's ρ and Kendall's τ on the union of all decoy sets. These estimate the capability of a method to compare quality of structures with different sequences, and thus to predict if a model is far or close to the native structure.

We compared our method only with single-model methods (we excluded consensus-based methods), which are listed below.

SBROD (Karasikov *et al.*, 2019) is a coarse-grain knowledge-based method trained using four types of structural features: residue-residue pairwise features, backbone atom-atom pairwise features, hydrogen bonding features and solvent-solvate features. We used the version from https://gitlab.inria.fr/grudinin/sbrod trained on CASP 5–10 server predictions. VoroMQA (Olechnovič and Venclovas, 2017) is a statistical potential trained on inter-atomic contact areas. We used the version included in the package voronota version 1.18.1877 from https://bitbucket.org/kliment/voronota/downloads/. RWplus (Zhang and Zhang, 2010) is a classical gold-standard statistical potential that uses a pairwise-dependent atomic potential, with a side chain orientation-dependent energy term. We used the binary provided at https://zhanglab.ccmb.med.umich.edu/RW/. 3DCNN (Derevyanko *et al.*, 2018) is another method that uses convolutional networks trained on protein density maps. We used the pretrained model from https://github.com/lamoureux-lab/3DCNN_MQA and followed the authors' QA script. Proq3D (Uziela *et al.*, 2017) is a method that combines many heterogeneous expert-defined features using a deep-learning algorithm. It contains different back-end models to fit different geometrical scores. To fairly compare Proq3D with our method, we used the version of Proq3D trained on CAD score with rotameric optimization enabled. We should also mention that Proq3D does not only rely on protein structure, as it uses a sequence database to extract the relevant evolutionary information (Ray *et al.*, 2012). We used the method available at https://bitbucket.org/ElofssonLab/proq3 with a version of UniRef sequence database (Suzek *et al.*, 2007) from 2013 that was already available when CASP 11 challenge started.

First, we evaluated the QA methods by comparing them with the global a.a. CAD score as a ground-truth score. Table 1 lists the results. We can see that our method is almost always ranked second after Proq3D. We should note that historically, the protein structure prediction community was biased toward using GDT-TS (Zhang and Skolnick, 2004) as a quality measure. Therefore, many of the machine learning-based methods (including those from our performance benchmark) have been specifically trained to approximate GDT-TS. Therefore, for a fair comparison, we ran an additional test. Here, we computed the loss, Pearson correlation and Kendall's τ using GDT-TS as the ground-truth score. Table 2 lists the performance results. We can see that Proq3D is again the best performing method (even though it was trained on CAD score). Here, our method performs, as expected, less impressive compared to the previous test (comparison with CAD score). In particular, the average correlations are not as high as for the VoroMQA or SBROD methods, which were specifically trained to match GDT-TS. However, it is the second best method for picking the best structure in three out of four datasets, and it is also among the best for the global correlations, meaning that our method is still useful to assess the absolute quality of models. It is also very interesting to see the performance comparison between Ornate and 3DCNN. Indeed, 3DCNN is the most similar method to our work, and it was our starting point. Thus, performing better than 3DCNN demonstrates the progress we made with the choice of the network architecture. Our method

Table 1. Performance of different methods for model quality assessment on CASP 11 and CASP 12 benchmarks

Set	St.	Model	Average P.L.	Average r	Average ρ	Average τ	Global r	Global ρ	Global τ
CASP 11	1	**Ornate***	0.030±0.004	0.70±0.02	0.64±0.03	0.50±0.04	0.79±0.02	0.77±0.02	0.58±0.03
		SBROD	0.042±0.006	0.68±0.03	0.60±0.03	0.46±0.04	0.59±0.03	0.57±0.03	0.39±0.04
		VoroMQA	0.036±0.005	0.69±0.03	0.60±0.03	0.46±0.04	0.73±0.02	0.73±0.02	0.53±0.03
		RWplus	0.056±0.007	0.63±0.03	0.56±0.03	0.42±0.04	0.17±0.05	0.16±0.05	0.10±0.05
		3DCNN	0.048±0.007	0.49±0.04	0.44±0.04	0.34±0.04	0.54±0.04	0.57±0.04	0.40±0.04
		Proq3D*	0.021±0.004	0.82±0.02	0.76±0.02	0.61±0.03	0.90±0.01	0.89±0.01	0.71±0.02
	2	**Ornate***	0.024±0.003	0.72±0.01	0.68±0.01	0.51±0.01	0.79±0.01	0.80±0.01	0.61±0.01
		SBROD	0.043±0.004	0.58±0.01	0.55±0.01	0.39±0.02	0.55±0.01	0.56±0.01	0.38±0.02
		VoroMQA	0.044±0.005	0.66±0.01	0.63±0.01	0.46±0.01	0.67±0.01	0.71±0.01	0.53±0.01
		RWplus	0.053±0.004	0.42±0.01	0.42±0.01	0.30±0.02	0.11±0.02	0.10±0.02	0.06±0.02
		3DCNN	0.052±0.005	0.50±0.01	0.50±0.01	0.36±0.02	0.60±0.01	0.63±0.01	0.45±0.01
		Proq3D*	0.024±0.003	0.71±0.01	0.69±0.01	0.51±0.01	0.89±0.004	0.89±0.004	0.71±0.01
CASP 12	1	**Ornate***	0.035±0.008	0.76±0.03	0.70±0.03	0.55±0.05	0.73±0.03	0.69±0.04	0.51±0.05
		SBROD	0.029±0.009	0.66±0.04	0.55±0.05	0.41±0.06	0.38±0.06	0.29±0.06	0.21±0.07
		VoroMQA	0.030±0.009	0.67±0.04	0.56±0.05	0.42±0.06	0.57±0.05	0.57±0.05	0.40±0.06
		RWplus	0.052±0.013	0.57±0.05	0.51±0.05	0.38±0.06	0.01±0.07	-0.01±0.07	-0.01±0.07
		3DCNN	0.054±0.013	0.34±0.06	0.18±0.06	0.12±0.07	0.26±0.06	0.22±0.06	0.15±0.07
		Proq3D*	0.023±0.007	0.81±0.02	0.75±0.03	0.60±0.04	0.83±0.02	0.80±0.02	0.61±0.04
	2	**Ornate***	0.028±0.005	0.78±0.01	0.75±0.01	0.57±0.02	0.81±0.01	0.79±0.01	0.60±0.02
		SBROD	0.049±0.007	0.69±0.01	0.63±0.02	0.47±0.02	0.51±0.02	0.49±0.02	0.34±0.02
		VoroMQA	0.048±0.008	0.73±0.01	0.69±0.01	0.53±0.02	0.64±0.02	0.66±0.01	0.50±0.02
		RWplus	0.063±0.006	0.64±0.02	0.60±0.02	0.43±0.02	0.05±0.03	0.05±0.03	0.04±0.03
		3DCNN	0.067±0.011	0.52±0.02	0.51±0.02	0.38±0.02	0.51±0.02	0.55±0.02	0.39±0.02
		Proq3D*	0.026±0.004	0.80±0.01	0.77±0.01	0.60±0.02	0.89±0.01	0.90±0.005	0.72±0.01

Note: The ground-truth measure is the global a.a. CAD score. The sign * indicates that the scoring function has been specifically trained to fit this measure. Confidence intervals at 95% are given. The three best performing methods are highlighted with increasing saturation, two methods whose confidence intervals overlap are ranked equally. P.L., r, ρ and τ stand for prediction loss, Pearson's r, Spearman's ρ and Kendall's τ, respectively.

Table 2. Performance of different methods for model quality assessment on CASP 11 and CASP 12 benchmarks

Set	St.	Model	Average P.L.	Average r	Average ρ	Average τ	Global r	Global ρ	Global τ
CASP 11	1	**Ornate**	0.077±0.009	0.47±0.04	0.37±0.04	0.28±0.05	0.64±0.03	0.63±0.03	0.44±0.04
		SBROD*	0.083±0.012	0.65±0.03	0.52±0.04	0.39±0.04	0.58±0.03	0.57±0.03	0.39±0.04
		VoroMQA	0.085±0.011	0.62±0.03	0.48±0.04	0.36±0.04	0.69±0.03	0.68±0.03	0.48±0.04
		RWplus	0.128±0.015	0.47±0.04	0.37±0.04	0.27±0.05	0.08±0.05	0.00±0.05	-0.02±0.05
		3DCNN*	0.087±0.013	0.52±0.04	0.39±0.04	0.29±0.05	0.59±0.03	0.39±0.03	0.29±0.04
		Proq3D	0.066±0.010	0.69±0.02	0.61±0.03	0.46±0.04	0.80±0.02	0.78±0.02	0.58±0.03
	2	**Ornate**	0.055±0.007	0.39±0.02	0.37±0.02	0.26±0.02	0.63±0.01	0.67±0.01	0.48±0.01
		SBROD*	0.058±0.007	0.43±0.01	0.41±0.01	0.29±0.02	0.55±0.01	0.57±0.01	0.39±0.02
		VoroMQA	0.066±0.008	0.42±0.01	0.41±0.01	0.29±0.02	0.65±0.01	0.69±0.01	0.51±0.01
		RWplus	0.088±0.010	0.17±0.02	0.19±0.02	0.13±0.02	0.06±0.02	0.03±0.02	0.01±0.02
		3DCNN*	0.066±0.007	0.40±0.02	0.39±0.02	0.27±0.02	0.64±0.01	0.67±0.01	0.48±0.01
		Proq3D	0.053±0.006	0.44±0.01	0.43±0.01	0.30±0.02	0.77±0.01	0.80±0.01	0.59±0.01
CASP 12	1	**Ornate**	0.113±0.024	0.57±0.05	0.50±0.05	0.37±0.06	0.55±0.05	0.48±0.05	0.34±0.06
		SBROD*	0.068±0.022	0.64±0.04	0.60±0.04	0.45±0.05	0.37±0.06	0.23±0.07	0.16±0.07
		VoroMQA	0.085±0.026	0.61±0.04	0.55±0.05	0.41±0.05	0.46±0.05	0.38±0.06	0.26±0.06
		RWplus	0.132±0.032	0.49±0.05	0.47±0.05	0.34±0.06	-0.27±0.07	-0.55±0.05	-0.38±0.06
		3DCNN*	0.085±0.013	0.44±0.05	0.22±0.04	0.16±0.05	0.49±0.05	0.60±0.04	0.43±0.05
		Proq3D	0.086±0.022	0.71±0.03	0.64±0.04	0.48±0.05	0.67±0.04	0.48±0.05	0.34±0.06
	2	**Ornate**	0.072±0.011	0.49±0.02	0.46±0.02	0.32±0.02	0.67±0.01	0.66±0.01	0.47±0.02
		SBROD*	0.079±0.010	0.61±0.02	0.55±0.02	0.40±0.02	0.47±0.02	0.49±0.02	0.34±0.02
		VoroMQA	0.106±0.017	0.56±0.02	0.50±0.02	0.36±0.02	0.61±0.02	0.60±0.02	0.45±0.02
		RWplus	0.103±0.018	0.42±0.02	0.38±0.02	0.27±0.02	-0.10±0.03	-0.10±0.03	-0.07±0.03
		3DCNN*	0.122±0.017	0.51±0.02	0.45±0.02	0.32±0.02	0.61±0.02	0.64±0.01	0.46±0.02
		Proq3D	0.060±0.011	0.60±0.02	0.54±0.02	0.39±0.02	0.81±0.01	0.80±0.01	0.60±0.02

Note: The ground-truth measure is GDT-TS. The sign * indicates that the scoring function has been specifically trained to fit this measure. Confidence intervals at 95% are given. The three best performing methods are highlighted with increasing saturation, two methods whose confidence intervals overlap are ranked equally. P.L., r, ρ and τ stand for prediction loss, Pearson's r, Spearman's ρ and Kendall's τ, respectively.

performs better on every single indicator, in both CAD score and GDT-TS tests, even though 3DCNN has been specifically trained to match GDT-TS.

3.2 Local scores

A particularity of Ornate is to also compute local scores for each model residue. This helps to predict which part of the model structure is poorly folded or should be refined. Figure 5 shows a few examples of CASP server predictions scored with Ornate (for targets T0854, T0367 and T0768) with both correctly modeled parts and poorly modeled ones. As a reference, we also show the same models colored according to the ground-truth local a.a. CAD scores, and the reference crystallographic structures. This figure demonstrates a good visual correlation between predictions of Ornate and the ground truth.

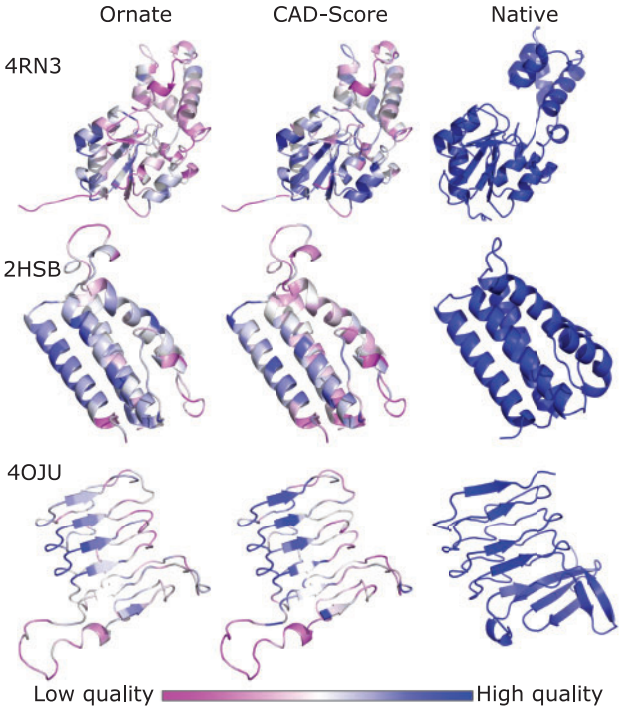

Fig. 5. Examples of local quality measures for three CASP server prediction targets T0854 (top, pdb code 4rn3), T0367 (middle, pdb code 2hsb) and T0768 (bottom, pdb code 4oju). The left column shows models colored according to the Ornate score. The center column shows models colored according to the ground-truth CAD score. The right column shows reference crystallographic structures. The three CASP models have respective GDT-TS measures of 0.657 (top), 0.634 (middle) and 0.469 (bottom), and CAD-score measures of 0.518 (top), 0.463 (middle) and 0.456 (bottom)

Table 3. Pearson correlations between CAD-score of individual residues and local scores computed by Ornate, VoroMQA and Proq3D on different datasets

Model	CASP 11		CASP 12	
	Stage 1	Stage 2	Stage 1	Stage 2
Ornate	0.494 ± 0.002	0.514 ± 0.001	0.466 ± 0.002	0.593 ± 0.001
VoroMQA	0.407 ± 0.002	0.417 ± 0.001	0.263 ± 0.002	0.463 ± 0.001
Proq3D	0.544 ± 0.001	0.550 ± 0.001	0.487 ± 0.002	0.656 ± 0.001

For a more quantitative assessment of the local scoring results, we computed the Pearson correlation between local a.a. CAD scores of all the residues in stages 1 and 2 of CASP 11–12 benchmarks and the local scores provided by Ornate, VoroMQA and Proq3D. These are listed in Table 3. Again, Proq3D shows the best performance among the three tested methods.

3.3 Computational details

We implemented the Ornate method using a combination of C++ and Python programming languages. The part for the generation of

input volumetric maps was written in C++. The CNN's training part uses Python with the TensorFlow framework (Abadi *et al.*, 2016). The computational time thus can also be decomposed into two parts that take approximately the same time. Creating a 3D map from a residue structure and its neighborhood takes about 30 ms (measured with an I7 CPU), and running the network for one map takes about 20 ms (measured on a GeForce GTX 680 GPU). Please also note that the latter time was measured for TensorFlow with GPU support, and it may be up to 100 times slower without it. Overall, the complexity of scoring one protein model grows linearly with the number of residues in the model structure. For example, scoring a mid-size protein structure with about 200 residues takes about 1 s.

4 Conclusion

This work presents Ornate, one of the first 3D CNN methods for the protein model QA problem. Ornate demonstrates a significant improvement over the previous 3D CNN attempts. This improvement was made possible thanks to several dedicated network topology designs that we introduced. These include residue-wise scoring, orientation of the input maps according to the backbone atoms, the retyper layer and data routing.

Ornate is competitive to most state-of-the-art single-model protein model QA methods. For example, when compared to the best of these, also trained to match CAD score, on stage 2 of CASP 11 (which seems to be the hardest dataset to score according to our measures), the average correlations and prediction losses of Ornate are at the same level as the ones from Proq3D. However, Proq3D, compared to Ornate, accesses additional sequence data. Even though Ornate does not reach the accuracy of the best meta-methods on all the indicators, its reasonable scoring time (about 1 s for mid-size proteins) makes it a good candidate to be integrated in such meta models. In addition, Ornate produces a smooth score with respect to atom positions in the model, as it has derivatives of all orders. Thus, its gradient can be easily computed. This property can be used for subsequent model refinement.

Acknowledgements

First, the authors thank Georgy Derevyanko from Concordia University, Montréal, who designed a preliminary version of a 3D CNN, which motivated us in constructing Ornate. The authors thank Kliment Olechnovič from Vilnius University for his support on using voronota, and Arne Elofsson and Karolis Uziela from Stockholm University for their support on Proq3D usage. Finally, the authors thank Stephane Redon from Inria, Grenoble, for his permanent interest and motivated discussions in deep-learning techniques, and also for his support on data visualization.

Funding

This work was supported by L'Agence Nationale de la Recherche [ANR-15-CE11–0029–03].

Conflict of Interest: none declared.

References

Abadi,M. *et al.* (2016). Tensorflow: a system for large-scale machine learning. In: *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, pp. 265–283.

Cao,R. and Cheng,J. (2016) Protein single-model quality assessment by feature-based probability density functions. *Sci. Rep.*, **6**, 23990.

Cao,R. *et al.* (2016) DeepQA: improving the estimation of single protein model quality with deep belief networks. *BMC Bioinform.*, **17**, 495.

Clevert,D.-A. *et al.* (2016) Fast and accurate deep network learning by exponential linear units (elus). In: *International Conf. on Learning Representations*.

Cozzetto,D. *et al.* (2007) Assessment of predictions in the model quality assessment category. *ProteinsStruct. Funct. Bioinform.*, **69**, 175–183.

Derevyanko,G. *et al.* (2018) Deep convolutional networks for quality assessment of protein folds. *Bioinformatics*, **34**, 4046–4053.

Ginalski,K. *et al.* (2003) 3D-Jury: a simple approach to improve protein structure predictions. *Bioinformatics*, **19**, 1015–1018.

Ioannou,Y. *et al.* (2016) Decision forests, convolutional networks and the models in-between. arXiv preprint arXiv: 1603.01250.

Ioffe,S. and Szegedy,C. (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning*, pp. 448–456.

Jiménez,J. *et al.* (2017) DeepSite: protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics*, **33**, 3036–3042.

Karasikov,M. *et al.* (2019) Smooth orientation-dependent scoring function for coarse-grained protein quality assessment. *Bioinformatics*, bty1037.

Leaver-Fay,A. *et al.* (2011) ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol.*, **487**, 545–574.

Lee,H. *et al.* (2009) Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 609–616.

Lundström,J. *et al.* (2001) Pcons: a neural-network–based consensus predictor that improves fold recognition. *Protein Sci.*, **10**, 2354–2362.

Lüthy,R. *et al.* (1992) Assessment of protein models with three-dimensional profiles. *Nature*, **356**, 83.

Mariani,V. *et al.* (2013) lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, **29**, 2722–2728.

Moult,J. *et al.* (2018) Critical assessment of methods of protein structure prediction (CASP)—Round XII. *Proteins Struct. Funct. Bioinform.*, **86**, 7–15.

Olechnovič,K. and Venclovas,Č. (2017) VoroMQA: assessment of protein structure quality using interatomic contact areas. *Proteins Struct. Funct. Bioinform.*, **85**, 1131–1145.

Olechnovič,K. *et al.* (2013) CAD-score: a new contact area difference-based function for evaluation of protein structural models. *Proteins Struct. Funct. Bioinform.*, **81**, 149–162.

Olechnovič,K. *et al.* (2018) Comparative analysis of methods for evaluation of protein models against native structures. *Bioinformatics*, bty760.

Ragoza,M. *et al.* (2017) Protein–ligand scoring with convolutional neural networks. *J. Chem. Inf. Model.*, **57**, 942–957.

Ray,A. *et al.* (2012) Improved model quality assessment using ProQ2. *BMC Bioinform.*, **13**, 224.

Ritchie,D.W. *et al.* (2012) Fast protein structure alignment using Gaussian overlap scoring of backbone peptide fragment similarity. *Bioinformatics*, **28**, 3274–3281.

Sippl,M.J. (1993) Recognition of errors in three-dimensional structures of proteins. *Proteins Struct. Funct. Bioinform.*, **17**, 355–362.

Suzek,B.E. *et al.* (2007) Uniref: comprehensive and non-redundant UniProt reference clusters. *Bioinformatics*, **23**, 1282–1288.

Townshend,R.J. *et al.* (2018) Generalizable protein interface prediction with end-to-end learning. arXiv preprint arXiv: 1807.01297.

Uziela,K. *et al.* (2017) ProQ3D: improved model quality assessments using deep learning. *Bioinformatics*, **33**, 1578–1580.

Van Dyk,D.A. and Meng,X.-L. (2001) The art of data augmentation. *J. Comput. Graph. Statist.*, **10**, 1–50.

Wallach,I. *et al.* (2015) AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. arXiv preprint arXiv: 1510.02855.

Worrall,D.E. *et al.* (2017) Harmonic networks: deep translation and rotation equivariance. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2 pp. 5028–5037.

Zhang,J. and Zhang,Y. (2010) A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction. *PLoS One*, **5**, e15386.

Zhang,Y. and Skolnick,J. (2004) Scoring function for automated assessment of protein structure template quality. *Proteins Struct. Funct. Bioinform.*, **57**, 702–710.