*Bioinformatics*, 38(19), 2022, 4505–4512
https://doi.org/10.1093/bioinformatics/btac551
Advance Access Publication Date: 13 August 2022
Original Paper

OXFORD

Structural bioinformatics

# Deep Local Analysis evaluates protein docking conformations with locally oriented cubes

**Yasser Mohseni Behbahani , Simon Crouzet , Elodie Laine* and Alessandra Carbone ***

Sorbonne Université, CNRS, IBPS, Laboratory of Computational and Quantitative Biology (LCQB), UMR 7238, Paris 75005, France

*To whom correspondence should be addressed.

Associate Editor: Lenore Cowen

Received on April 5, 2022; revised on July 4, 2022; editorial decision on August 6, 2022; accepted on August 8, 2022

## Abstract

**Motivation:** With the recent advances in protein 3D structure prediction, protein interactions are becoming more central than ever before. Here, we address the problem of determining how proteins interact with one another. More specifically, we investigate the possibility of discriminating near-native protein complex conformations from incorrect ones by exploiting local environments around interfacial residues.

**Results:** Deep Local Analysis (DLA)-Ranker is a deep learning framework applying 3D convolutions to a set of locally oriented cubes representing the protein interface. It explicitly considers the local geometry of the interfacial residues along with their neighboring atoms and the regions of the interface with different solvent accessibility. We assessed its performance on three docking benchmarks made of half a million acceptable and incorrect conformations. We show that DLA-Ranker successfully identifies near-native conformations from ensembles generated by molecular docking. It surpasses or competes with other deep learning-based scoring functions. We also showcase its usefulness to discover alternative interfaces.

**Availability and implementation:** http://gitlab.lcqb.upmc.fr/dla-ranker/DLA-Ranker.git

**Contact:** elodie.laine@sorbonne-universite.fr or alessandra.carbone@sorbonne-universite.fr

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

## 1 Introduction

Protein–protein interactions play a central role in virtually all biological processes. Reliably predicting who interacts with whom in the cell and in what manner would have tremendous implications for bioengineering and medicine. Hence, a lot of effort has been put into the development of methods for simulating protein–protein docking (Lensink *et al.*, 2007, 2020). While highly efficient algorithms can exhaustively sample the space of complex candidate conformations (Ritchie and Venkatraman, 2010), correctly evaluating and ranking these conformations remains challenging.

The classical docking and scoring paradigm has been recently challenged by the spectacular advances in protein structure prediction with AlphaFold version 2 (AF2) (Jumper *et al.*, 2021) and RosettaFold (Baek *et al.*, 2021). In particular, a handful of studies have showcased the potential of AF2, or a slightly modified version, in fold-and-dock strategies (Bryant *et al.*, 2022; Evans *et al.*, 2021; Humphreys *et al.*, 2021; Mirdita *et al.*, 2021). Nevertheless, they have also emphasized clear limitations. AF2 performs poorly on some eukaryotic complexes, antibody-antigen complexes and complexes displaying small interfaces (Bryant *et al.*, 2022; Evans *et al.*, 2021). In such cases, the output is limited to an unreliable conformation. In contrast, docking algorithms allow for the generation of conformational ensembles useful to guide the prediction of interfaces, to gain insight into protein sociability (Laine and Carbone, 2017) and to discover alternative binding modes and new partners (Dequeker *et al.*, 2022). These observations motivate the development of accurate and efficient methods assessing the quality of docking conformations.

The Critical Assessment of PRedicted Interactions (CAPRI) classifies predicted protein complex conformations in four categories, namely incorrect, acceptable, medium and high quality, based on the extent to which they differ from the corresponding experimental structures (Lensink *et al.*, 2017). Recently, several methods leveraging deep learning have been proposed to discriminate near-native (acceptable or higher quality) from incorrect conformations (Cao and Shen, 2020; Eismann *et al.*, 2021; Renaud *et al.*, 2021; Wang *et al.*, 2020, 2021). They adopt a 'global' perspective by assessing the quality of the full interface (Renaud *et al.*, 2021; Wang *et al.*, 2020, 2021) or even the complex as a whole (Eismann *et al.*, 2021). Standard 3D-convolutional neural networks (3D-CNN) have been applied to a voxelized 3D grid representing the entire interface (Renaud *et al.*, 2021; Wang *et al.*, 2020). This representation has two limitations. First, when a fixed-size cube is used as grid, it might not cover very large and/or discontinuous interfaces. Using a very large cube to accommodate any interface is memory inefficient.

Large cubes of fixed size may also hinder the accuracy in case of small interfaces due to the information vanishing after a few layers of pooling. Second, since the 3D-CNN does not benefit from the rotational symmetry endowed to the Euclidean space, it is sensitive to the orientation of the candidate conformation and its output may change upon rotation of the input in an uncontrolled fashion.

Rotational data augmentation was used in Renaud *et al.* (2021) to limit this effect but at the expense of dramatically increasing the computational cost for training the model. A more efficient solution is to use a SE(3)-equivariant CNN architecture instead of standard CNN. SE(3)-equivariant CNN makes use of spherical harmonics, a set of functions defined on the unit sphere, to guarantee that a rotation of the input results in the same rotation of the output (Cohen and Welling, 2016; Fuchs *et al.*, 2020; Thomas *et al.*, 2018; Weiler *et al.*, 2018). In Eismann *et al.* (2021), SE(3)-equivariant hierarchical convolutions were applied to a point-cloud representation of the whole conformation. Finally, graph-based representations, such as those used in GNN-DOVE (Wang *et al.*, 2021) and DeepRank-GNN (Réau *et al.*, 2021), are invariant to 3D rotations, but at the expense of losing information about the orientations of the atoms with respect to each other.

Alternatively, one can leverage the specific properties of proteins, whose building blocks (the amino acid residues) share the same chemical scaffold, to derive a SE(3)-equivariant representation. In single protein structure prediction, Ornate (Pagès *et al.*, 2019), Sato-3DCNN (Sato and Ishida, 2019) and more recently AlphaFold2 (Jumper *et al.*, 2021), benefit from these properties and make use of oriented local frames centered on each protein residue. Such representation circumvents the problem of 3D rotational symmetry without the need for rotational data augmentation nor for SE(3)-equivariant convolutional filters.

In this work, we investigate the possibility of discriminating near-native complex candidate conformations from incorrect ones by exploiting and combining two kinds of information: (i) local 3D-geometrical and physico-chemical environments around the interfacial residues and (ii) regions of the interface with different solvent accessibility. We represent the interface by the unique and well-determined set of locally oriented residue-centered cubes lying between the interacting proteins in the candidate conformation (Fig. 1A). The cubes are oriented by defining local frames based on the common chemical scaffold of amino acid residues in proteins. A cube encapsulates the local environment of the residue, i.e. the local geometry of the residue together with its neighboring atoms. No evolutionary information associated to residues is considered. Our motivations for such a representation are multiple:

1. The number of known protein–protein complex structures is fairly limited. Breaking down these structures into interfacial residue-centered local environments allows training on a much larger set of input samples (cubes) compared to the number of interfaces.
2. Our representation guarantees that the output is invariant to the global orientation of the input conformation while fully accounting for the relative orientation of a residue with respect to its neighbors.
3. We wanted to investigate the minimal unit of information at the interface which is necessary to predict the quality of an interaction. By relying on minimal units, i.e. residue-centered cubes, one can also evaluate interfaces between three or more proteins.
4. The set of cubes belonging to the interface can be organized in three subsets depending on the solvent accessibility of the interfacial residues. The cubes within each subset are independent from each other and from the geometry of the surface. We wanted to study the contribution of these three subsets in ranking docking conformations.

We propose Deep Local Analysis (DLA)-Ranker, a deep learning-based approach ranking candidate complex conformations by applying 3D-CNN to a set of locally oriented cubes representing the residues of its putative interface.

## 2 Materials and methods

Our goal is to design a classifier that can effectively distinguish near-native protein candidate conformations from incorrect ones by learning from a local representation of the structure of the interface. Such representation should account for the local geometrical arrangement of interfacial atoms in the Euclidean space and their physico-chemical properties.

### 2.1 Protein–protein interface representation

DLA-Ranker takes as input a cubic volumetric map centered and oriented on each putative interfacial residue (Fig. 1A). It exploits only information coming from a candidate complex conformation, without any knowledge about which residues are actually part of the native interface. The putative interface is defined as the set of residues displaying a change in solvent accessibility between the free (isolated) proteins and the candidate complex. We used NACCESS (Hubbard and Thornton, 1993) with a probe radius of 1.4 Å to compute residue solvent accessibility. To build the map, we adapted the method proposed in Pagès *et al.* (2019). The atomic coordinates of the input conformation are first transformed to a density function. The density $d$ at a point $\vec{v}$ is computed as

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

where $\vec{a_i}$ is the position of the *ith* atom, $\sigma$ is the width of the Gaussian kernel and is set to 1 Å and $t_i$ is a vector of dimension 169 encoding some characteristics of the protein atoms. Namely, the first 167 dimensions correspond to the atom types that can be found in amino acids (without the hydrogens) (Pagès *et al.*, 2019), and the 2 other dimensions correspond to the two partners, the receptor and the ligand. Then, the density is projected on a 3D grid comprising $24 \times 24 \times 24$ voxels of side 0.8 Å. For the *nth* residue, the $(\vec{x}, \vec{y}, \vec{z})$ directions and the origin of the cube are defined by the position of the atom $N_n$, and the directions of $C_{n-1}$ and $C\alpha_n$ with respect to $N_n$. The $X$-axis is parallel to the vector pointing from $C_{n-1}$ to $N_n$. The $Y$-axis is perpendicular to the $X$-axis and is defined such that $C\alpha_n$ lies in the half-plane $Oxy$ with $y > 0$. The $Z$-axis is defined as a vector product, $Z = X \times Y$. The origin of the cube is determined such that $N_n$ is located at position (6.1 Å, 6.6 Å and 9.6 Å). This choice ensures that all the atoms of the central residue fit in the cube. More details can be found in Pagès *et al.* (2019). Thanks to this local frame definition, the map not only is invariant to the candidate conformation initial orientation but also provides information about the atoms and residues relative orientations.

Depending on the location of the residues at the interface, their geometrical and physico-chemical environments are expected to be very different. For instance, the map computed for a residue deeply buried in the interface will be much more dense than that computed for a partially solvent-exposed residue at the rim. This motivated us to explicitly give some information to the network about the location of the input residue at the interface. To do so, we classified the interfacial residues in three structure classes, the Support (S), the Core (C) and the Rim (R) (Fig. 1A), as defined in Levy (2010). We one-hot encode the input residue class in a vector $u$ and append it to the embedding computed by DLA-Ranker (see below and Fig. 1B, concatenation layer). The SCR classification previously proved useful for the prediction and analysis of protein–protein and protein–DNA interfaces (Corsi *et al.*, 2020; Laine and Carbone, 2015; Raucci *et al.*, 2018).

### 2.2 DLA-Ranker architecture

The DLA-Ranker architecture comprises a projector, three 3D convolutional layers, a max pooling layer and three fully connected layers (Fig. 1B). The projector maps the feature vector of each voxel

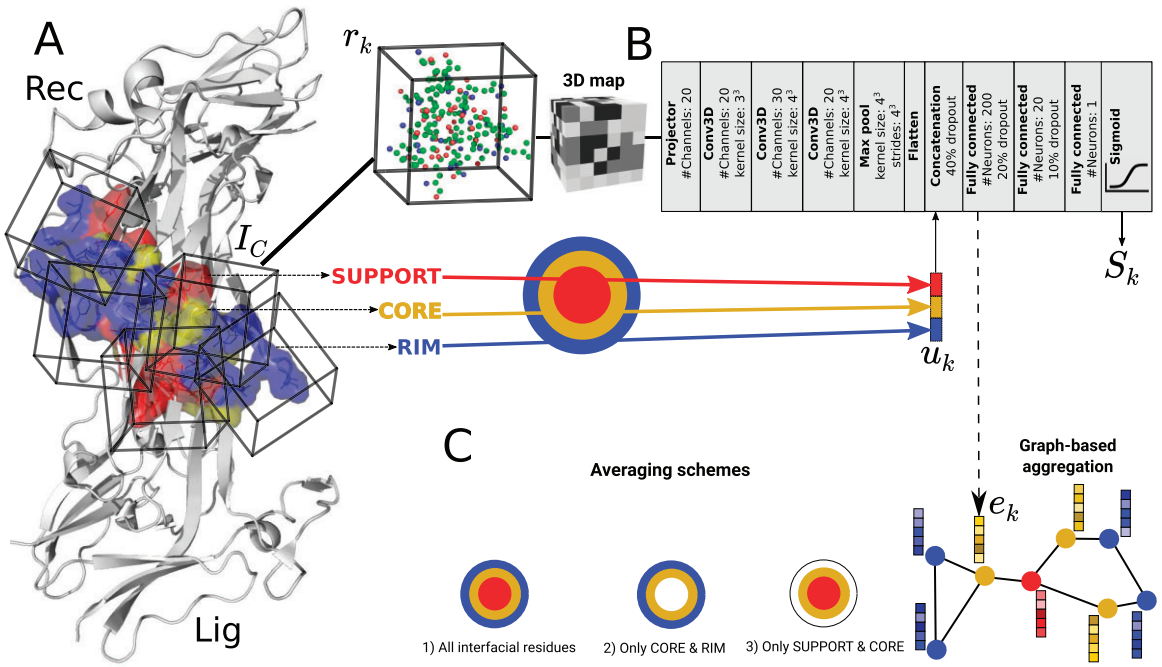


**Fig. 1.** Interface representation and DLA-Ranker architecture. (**A**) Representation of a protein complex putative interface as an ensemble of cubes ($I_C$). Each cube ($r_k \in I_C$) is centered and oriented on an interfacial residue. It contains atoms belonging to the residue and its local environment (Carbon: green, Oxygen: red, Nitrogen: blue, Sulfur: yellow). A cube is labeled as being part of the Support (red), Core (gold) or Rim (blue) of the interface (one-hot encoded vector $u_k$). (**B**) Architecture of DLA-Ranker neural network. For input cube $r_k$, the network has two outputs: score $S_k$ and embedding vector $e_k$. (**C**) The evaluation of the interface either by global averaging the local scores $S_k$ (1) over all interfacial residues, (2) over residues from SC and (3) over residues from CR, or by extracting embedding vectors $e_k$ and combining them through graph-based aggregation (A color version of this figure appears in the online version of this article.)

into a vector of size 20. Each convolutional layer is followed by a batch normalization layer. The max pooling layer exploits scale separability by preserving essential information of the input during coarsening of the underlying grid. The one-hot encoded vector of the residue structure class ($u$) is concatenated to the embedding derived from the convolutional layers (i.e. output of the flatten layer). To avoid overfitting, we used 40%, 20% and 10% dropout regularization on the input, first and second layers of the fully connected subnetwork, respectively. The last activation function (Sigmoid) outputs a score comprised between 0 and 1 for each input interfacial residue. The loss function is the binary cross-entropy measuring the difference between the probability distribution of the predicted output and the given label (0 or 1). The objective of training is to minimize this loss with respect to the trainable parameters: reaching higher output scores for the residues belonging to a near-native conformation and lower output scores for the residues of incorrect conformations. We used the Adam optimizer with a learning rate of 0.001 in TensorFlow (Abadi *et al.*, 2016).

## 2.3 Aggregation of individual residue-based scores

To evaluate a candidate conformation, DLA-Ranker applies global averaging on the individual residue scores over the interface. The predicted quality $Q$ of conformation $C$ is expressed as

$$Q_C = \frac{1}{|I_C|} \sum_{r_k \in I_C} S_k, \tag{2}$$

where $I_C$ is the ensemble of interfacial residues and $S_k$ is the score predicted by the network for the input 3D map centered on residue $r_k$.

To investigate whether we could improve on this global averaging baseline, we considered two approaches. First, we proposed two additional evaluation schemes based on an average restricted to a selection of subsets of residues at the interface: (i) residues of S and C regions and (ii) residues of C and R regions (Fig. 1C). Second, we applied different weights to the residues comprising the interface by using graph-based attention (Veličković *et al.*, 2018) (Fig. 1C and Supplementary Fig. S1). Namely, we extracted the embeddings $e_k$ computed by the first fully connected layer of DLA-Ranker and used them as node features in a graph representing the interface, where two nodes are linked if the distance between their associated residues is <5.0 Å. We apply one layer of self-attention and predict a unique score estimating the quality of the whole interface (Supplementary Fig. S1).

## 2.4 Metrics for comparing conformations

To estimate the deviation of a candidate conformation from the ground-truth experimental conformation, we relied on two metrics, namely L-RMSD and I-RMSD (Lensink *et al.*, 2017) (Supplementary Fig. S2). The L-RMSD (Ligand-Root Mean Square Deviation) measures the deviation displayed by the ligand between the candidate conformation and the ground-truth conformation, after superimposing the receptors of the two conformations (Supplementary Fig. S2B). The I-RMSD (Interface-Root Mean Square Deviation) measures the deviation of the interface, defined as the ensemble of residues having any heavy atom within 10 Å of the partner (Supplementary Fig. S2C). Both metrics are computed over the backbone atoms of the selected residues.

## 2.5 Datasets

To train and test DLA-Ranker and compare its performance with different approaches, we used three databases of docking conformations derived from the structural data contained in the Protein Data Bank (Berman *et al.*, 2000).

### 2.5.1 CCD4PPI: docking conformations produced by MAXDo

We compiled our primary database, which we call CCD4PPI, from two complete cross-docking experiments performed on the datasets P-262 (Dequeker *et al.*, 2019; Lagarde *et al.*, 2018) and PPDBv2 (Lopes *et al.*, 2013; Mintseris *et al.*, 2005) using the docking tool MAXDo (Sacquin-Mora *et al.*, 2008). Both P-262 (262 proteins) and PPDBv2 (168 proteins) cover a large variety of functional

classes, such as antibody–antigen, enzyme-regulator and substrates-inhibitor (Dequeker *et al.*, 2019). We set aside 20 pairs for testing (Supplementary Table S1) and selected 312 protein pairs for training purposes (Supplementary Fig. S3). For about half of the pairs, the docking was performed using the unbound forms of the proteins. The PDB chains in the associated ground-truth experimental complex structures have at least 70% sequence identity with the docked PDB chains. For the remaining half, the bound forms were used in the docking calculations. MAXDO represents proteins as coarse-grained rigid bodies. To reconstruct the high-resolution docked conformations, we used INTBuilder (Dequeker *et al.*, 2017) starting from the Euler angles provided by MAXDo. We selected the training set of protein pairs based on the quality of the docked conformations. Specifically, for P-262, we efficiently screened 27 million docked conformations with INTBuilder and the rigidRMSD library (Neveu *et al.*, 2018), and systematically evaluated their quality with respect to the experimentally resolved complex structures. For PPDBv2, we obtained the list of acceptable and incorrect conformations from Nadalin and Carbone (2018). In total, we identified 3902 acceptable or higher quality conformations (L-RMSD $<$ 10.0 Å and I-RMSD $<$ 4.0 Å) and we retained all of them for training DLA-Ranker. Among the ensemble of incorrect conformations, we selected a subset of 6038 for training. Specifically, we first filtered out the conformations with unfavorable (positive) docking energies. Then, for each protein pair, we selected the $10 \times N_{acc}$ best-scored conformations, where $N_{acc}$ is the number of acceptable conformations, and finally chose one-sixth of those randomly.

#### 2.5.2 BM5: docking conformations produced by HADDOCK

The Docking Benchmark version 5 (BM5) (Vreven *et al.*, 2015) comprises 231 non-redundant (at the SCOP family level) target complexes from multiple functional classes, including antibody–antigen and enzyme-inhibitor, and with the corresponding unbound protein structures. We considered a total of 449 158 candidate conformations coming from 142 dimer target complexes. They were generated, selected and labeled by Renaud and co-authors using the protocol reported in (Renaud *et al.*, 2021). Specifically, for each target complex, 25 300 docking models were generated using the integrative modeling platform HADDOCK (Dominguez *et al.*, 2003) in three stages: (i) rigid-body docking, (ii) semi-flexible refinement by simulated annealing in torsion angle space and (iii) final refinement by short molecular dynamics in water (Renaud *et al.*, 2021). Almost all (99%) the models were produced starting from the unbound structures of the proteins. To generate a suitable amount of near-native conformations, both *ab initio* docking and docking guided by the knowledge of the interface were performed. Then, the resulting set of conformations was reduced to avoid redundancy. The conformations with I-RMSD $\leq$4.0 Å were labeled as near-native. On average, each target complex has ≈230 near-native conformations and ≈2932 incorrect ones.

#### 2.5.3 Dockground: docking conformations produced by Gramm-X

We downloaded the Dockground database 1.0 (Kundrotas *et al.*, 2018; Liu *et al.*, 2008) from http://dockground.compbio.ku.edu/downloads/unbound/decoy/decoys1.0.zip. It comprises 61 target complexes for which candidate conformations were generated by the Fast Fourier Transform-based method GRAMM-X (Tovchigrechko and Vakser, 2005) starting from the unbound structures of the proteins. On average, each target complex is associated with 108 candidate conformations, of which 9.83 are acceptable (L-RMSD $\leq$ 5.0 Å) and 98.5 are incorrect. The incorrect conformations represent only a small fraction of the full docking conformational ensemble. They were chosen because they display a degree of shape complementarity similar to the near-native ones and they yield a maximally spread spatial distribution around the latter (Kundrotas *et al.*, 2018). For comparison purposes, we used the same division of the dataset into four non-redundant groups as that reported in Wang *et al.* (2021). Any two complexes coming from different groups share <30% sequence identity and display a TM-score lower than 0.5 (Zhang and Skolnick, 2005).

### 2.6 Training protocol

We used CCD4PPI to optimize DLA-Ranker hyperparameters. In total, we explored about 10 different architectures by varying the number of convolutional layers, the number of neurons in the fully connected layers, and the dropout rates. We chose the best-performing architecture and used it for producing our final results and performing the comparisons with the other methods. We trained several independent models of DLA-Ranker using each of the three considered databases. Using CCD4PPI, we trained 5 models over 20 epochs through a 5-fold cross-validation procedure on the 312 protein pairs (Supplementary Fig. S4). For comparison purposes, we reproduced the same training protocols as those reported for DeepRank (Renaud *et al.*, 2021) and GNN-DOVE (Wang *et al.*, 2021) on BM5 and Dockground, respectively. Specifically, to compare DLA-Ranker with DeepRank, we performed 10-fold cross-validation by splitting the set of 142 dimers selected from BM5 in 114 for training, 14 for validation and 14 for testing. In total, 140 target complexes were used in the test sets (complexes BAAD and 3F1P were not included in the testing). We should stress that, contrary to what was done in Renaud *et al.* (2021), we did not augment the input conformational ensemble by random rotations since DLA-Ranker is not sensitive to the orientation of the input conformation. To compare DLA-Ranker with GNN-DOVE, we trained four models following 4-fold cross-validation on Dockground as reported in Wang *et al.* (2021). For each model, we used three non-redundant groups for training and validation (45 or 44 complexes) and the remaining one for testing (15 or 14 complexes). In all three databases, the incorrect conformations are much more abundant than the near-native ones. To compensate the effect of imbalanced training sets and elevate the importance of errors made on near-native poses compared to incorrect ones, we assigned higher weights to the loss of the acceptable class. We used class weights (0.823, 1.273), (0.54, 6.75) and (0.071, 0.929) for CCD4PPI, Dockground and BM5, respectively.

### 2.7 Evaluation metrics

We used hit rate and enrichment factor to evaluate the performance of DLA-Ranker in ranking candidate conformations. Hit rate curves show the fraction of target complexes in the test set with at least one near-native conformation within the top-ranked conformations. Enrichment factor for an individual target complex is defined as the fraction of acceptable conformations found in the top-ranked conformations. In case of CCD4PPI, we ranked the conformations using a consensus of the five trained models. To do so, we first ordered the conformations according to their scores computed from each trained model. Then, we discretized the ranks into six bins, namely labels top1, top5, top10, top50, top100 and top200. This way we could represent each conformation as a sequence of ranking labels predicted by five models. Finally, we 'lexicographically' ordered these labels and reported the hit rate of each individual complex separately.

## 3 Results

### 3.1 Identifying near-native conformations

We first assessed DLA-Ranker's ability to correctly rank candidate conformations. We selected the 1000 conformations best scored by MAXDo for each of the 20 test protein pairs from CCD4PPI and we re-ranked them according to the $Q$ scores predicted by DLA-Ranker. We primarily considered a consensus of the five trained models (see Section 2) and compared the obtained rankings with those provided by MAXDO (Fig. 2A). The latter evaluates conformations using a physics-based scoring function very similar to that of ATTRACT (Zacharias, 2003). For most of the pairs, DLA-Ranker assigned high $Q$ scores to the near-native conformations and discriminated them from the incorrect ones (Fig. 2A and Supplementary Fig. S5). The top-ranked conformation was near-native in two-thirds of the protein pairs (Fig. 2A). DLA-Ranker achieved better performance than MAXDo in 11 cases. DLA-Ranker's performance does not depend on the sequence similarity between the test protein pairs and the training pairs (Supplementary Table S1). For instance, it performs very well on

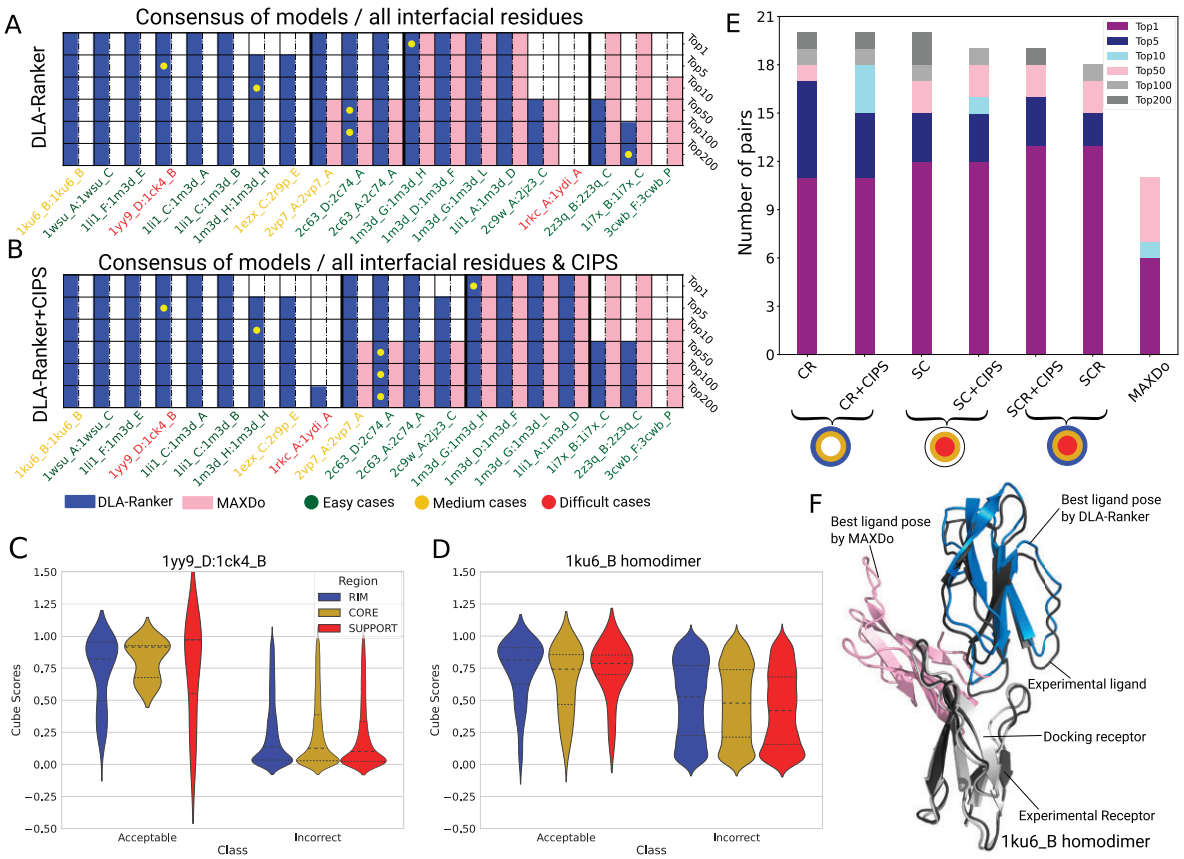


**Fig. 2.** DLA-Ranker performance on CCD4PPI database. (**A** and **B**) Ranking results per protein pair when all interfacial residues are used for train and test according to experimental setup 1 (C). For each pair, we report whether some near-native conformations were found in the top 1, 5, …, 200 out of a total of 1000 conformations generated and selected by MAXDo. A colored cell indicates the presence of at least one acceptable conformation in the corresponding topX. The pink color corresponds to MAXDo, while the blue color corresponds to DLA-Ranker (A) or DLA-Ranker combined with CIPS (B). For each topX, the yellow dot indicates the pair with the highest enrichment factor. The PDB ids are colored according to the magnitude of the conformational change between the docked forms and the bound forms. Green: none or small. Orange: medium. Red: large. (**C** and **D**) Distribution of individual scores based on S, C, R classes for acceptable and incorrect poses of complex 1yy9_D:1ck4_B (C) and 1ku6_B homodimer (D). (**E**) Comparison between different methods. The SCR, SC and CR DLA-Ranker models were trained and tested on all interfacial residues, only those in the support and core, or only those in the core and rim, respectively. (**F**) Best-ranked candidate conformations for the 1ku6_B homodimer. The reference complex structure is in black, the docked receptor in grey, the ligand conformation selected by MAXDo in pink and that selected by DLA-Ranker in blue (A color version of this figure appears in the online version of this article.)

the 1ku6_B and 2vp7_A homodimers (Fig. 2A), both sharing <30% sequence identity with any pair from the training set. In contrast, it fails to identify a near-native conformation in the top 200 for the 1rkc_A:1ydi_A pair sharing more than 70% sequence identity with the training set. This pair is also very challenging for MAXDo. DLA-Ranker's ability to single out near-native conformations for protein pairs not seen during training and not similar to the training pairs was confirmed when considering the models individually (Supplementary Fig. S6). Overall, DLA-Ranker performance also does not depend on the extent of conformational change between the docked protein forms and the bound forms (Fig. 2A and B, label colors). For instance, one of the cases where it performs very well, the 1ku6_B homodimer, displays a substantial rearrangement (Fig. 2F). Combining DLA-Ranker with the pair potential CIPS (Nadalin and Carbone, 2018) improved the results (Fig. 2B). In particular, it allowed enriching the top 200 subset in near-native conformations for the difficult case of 1rkc_A:1ydi_A, and surpassing MAXDo for the pair 2c9w_A:2jz3_C.

We further investigated the behavior of DLA-Ranker for the different sub-regions of the interface, namely the support, core and rim on two pairs of the database, 1yy9_D:1ck4_B and 1ku6_B homodimer. For both pairs, we observed a wide range of predicted scores within each sub-region (Fig. 2C and D). The score distributions for the three sub-regions often display similar shapes. Nevertheless, it may happen that DLA-Ranker performs significantly differently from one sub-region to the other, as exemplified by the pair 1yy9_D:1ck4_B. In this case, the scores predicted for the residues lying in the support of the interface are not discriminative enough. Averaging the residues' individual scores over the three interface sub-regions allows correctly classifying the conformations. At the residue level, DLA-Ranker can analyze per-residue scores across near-native conformations to highlight to what extent each residue fits in the interface (Supplementary Fig. S7A and B).

## 3.2 Comparison with other scoring functions

We compared DLA-Ranker with two deep learning-based scoring functions, namely DeepRank (Renaud *et al.*, 2021) and GNN-DOVE (Wang *et al.*, 2021). We used all interfacial residues for training and assessed different sub-region combinations (three averaging schemes: SCR, SC and CR) for testing. For both comparisons, DLA-Ranker performance was assessed using cross-validation, where the protein pairs used for testing do not share any homology with those used for training (see Section 2).

DeepRank applies standard 3D convolutions to a unique voxelized grid representing the interface. On a collection of 10 test sets of 14 target complexes from BM5 (see Section 2), DLA-Ranker significantly outperforms DeepRank (Fig. 3 and Supplementary Fig. S8). It yields a higher enrichment for both the 'raw' conformations produced by the rigid-body docking (Fig. 3A) and the semi-flexibly refined conformations (Fig. 3B). The enrichment curves obtained on the set of conformations further refined through molecular dynamics simulations in explicit water are almost superimposed (Supplementary Fig. S8).

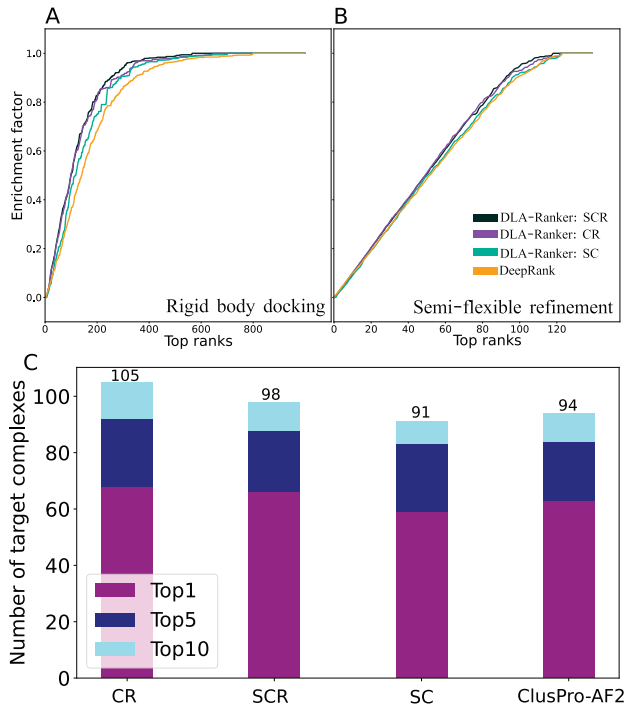


**Fig. 3.** Performance of DLA-Ranker on the 140 dimers of the BM5 database. (**A** and **B**) A comparison between the performance of DLA-Ranker (score averaging schemes SCR, CR and SC) and DeepRank (orange). Each curve reports the median enrichment over 10 test sets of 14 target complexes (see Section 2). See Supplementary Figure S8 for both median and the interval between 25% to 75% percentiles. (A) Only rigid body docking decoys. (B) Decoys with semi-flexible refinement. See Supplementary Figure S8 for the performance on decoys with water refinement. (**C**) A comparison between combination of HADDOCK and DLA-Ranker and ClusPro-AF2 in protein complex structure prediction in terms of number of target complexes with at least one acceptable or higher quality conformation at top1, top5 and top10 (A color version of this figure appears in the online version of this article.)

GNN-DOVE represents the interface as a graph and captures the information on the intermolecular interactions using graph attention mechanisms (Wang *et al.*, 2021). DLA-Ranker and GNN-DOVE display comparable hit rates on Dockground (Supplementary Fig. S9). While GNN-DOVE identifies a near-native conformation in the top 5 for more complexes than DLA-Ranker, DLA-Ranker covers more complexes when looking at the top 15 conformations. The results differ from one fold to another and this observation may be explained by the small size of the database. It contains about 5000 conformations versus ~10 000 for CCD4PPI and 450 000 for BM5 (see Section 2). In the second fold, we observe a lower performance for DLA-Ranker, due to the presence of an outlier complex, namely the ribonuclease inhibitor complex (1DFJ_E_I). The structure of this complex displays several loops on the interface (Supplementary Fig. S10A). By comparison, the other structures of the ribonuclease inhibitor complex available in the PDB have more structured interfaces (Supplementary Fig. S10A). The t-SNE (t-distributed stochastic neighbor embedding) analysis (averaged over the interface) of 1DFJ_E_I shows less separability compared to those of other complexes from the test set (Supplementary Fig. S10B–E).

### 3.3 Influence of the interface description

We investigated whether DLA-Ranker could still discriminate near-native from incorrect conformations with a partial description of the interfaces. To do so, we re-trained DLA-Ranker on CCD4PPI using two different subsets of the interfacial residues: (i) the support and core (SC), or (ii) the core and rim (CR). In the test phase, we aggregated the predicted residue-based scores over the same combination as that used during training (Fig. 1C). The results obtained on the 20 test protein pairs from CCD4PPI show that DLA-Ranker captures sufficient information with a partial description of the interface (Fig. 2E). The CR model yielded the best overall performance, and allowed to retrieve near-native conformations in the top 5 for almost all protein pairs (see also Supplementary Fig. S11). In addition, we assessed the partial aggregation schemes on BM5 (Fig. 3 and Supplementary Fig. S8) and Dockground (Supplementary Fig. S9) by the models trained using all interfacial residues. The results are consistent with those on CCD4PPI, with the combination of core and rim yielding a higher performance than the combination of support and core.

We also checked whether we could exploit the topological information of the interface to aggregate the learned residue-based representations. We extracted the embeddings learned by DLA-Ranker on Dockground and used them as node features in a graph representation of the interface (Fig. 1C). We observed that the graph-based aggregation does not improve over the global averaging scheme (Supplementary Fig. S12C–F). This result can be explained by the fact that the individual embeddings already encode global information about the interface since the labels used during training (acceptable or incorrect) are defined at the level of the interface (Supplementary Fig. S12A). This limits the learning capacity of the graph representation, which thus tends to overfit the training set (Supplementary Fig. S12B). The similarity between the embeddings in the training set causes homogeneous attention weights and as a result, the topology will not influence the learning.

### 3.4 Comparison with ClusPro-AF2

We compared our approach to the recently proposed ClusPro-AF2 protocol (Ghani *et al.*, 2021), where AF2 (Jumper *et al.*, 2021) is used to refine and complement the candidate conformations generated and selected by the docking tool ClusPro (Kozakov *et al.*, 2017). ClusPro-AF2's overall performance on the test set of 140 dimers from BM5 is similar to those we obtained by applying DLA-Ranker on the candidate conformations produced by HADDOCK (Fig. 3C). Moreover, using only the residues located in the core and the rim of the interfaces for DLA-Ranker evaluation increases the number of complexes for which a near-native conformation is found in the top 5 and 10 (Fig. 3C, see CR). Considering top 10 ranking, there are 19 complexes for which ClusPro-AF2 predicts acceptable or higher quality conformations, while DLA-Ranker cannot find any acceptable one. Five of these complexes (2OT3, 2I9B, 1ATN, 1RKE and 1R8S) have very few acceptable conformations in the ensemble of poses generated by HADDOCK. Reciprocally, there are 23 complexes that are well predicted by DLA-Ranker and are particularly challenging cases for ClusPro-AF2. These include complexes between proteins coming from a pathogen and its host (1EFN, 4H03, 2A9K, 1AK4 and 1MAH), complexes from the immune system (1GHQ, 1SBB, 1KXQ, 4M76 and 2I25), enzyme-inhibitor complexes (1PXV, 1JTD and 2ABZ) and regulatory complexes (1GLA and 1B6C). While ClusPro-AF2 produces only conformations of very low quality for these complexes, DLA-Ranker is able to identify at least one near-native conformation for 10 of these complexes at top 1, 3 in the top 5 and 2 complexes in the top 10.

### 3.5 Unraveling alternative interfaces

Finally, we explored the potential of DLA-Ranker to discover alternative interfaces. As a case study, we considered the SQD1 enzyme which can self-assemble into homodimers (1qrr) and homotetramers (1i24). We docked the protein (chain 1qrr_A) against itself using ATTRACT and evaluated all interfacial residues detected in the 3000 best candidate conformations with DLA-Ranker. In Figure 4A, we show the propensity of these residues to have a score higher than 0.5 according to DLA-Ranker. We can clearly identify three patches of residues, which appear in acceptable interfaces (Fig. 4A, see residues in red). The first one corresponds to the homodimeric interface found in 1qrr (Fig. 4A, the other copy of the protein, i.e. the partner is in green). The second one corresponds to another interface found in the homotetramer 1i24 (Fig. 4A, partner in violet). Finally, the third one is supported by the homotetramer 1wvg, whose chains are homologous to the SQD1 enzyme [*E*-value = 8.58e−4, identified using the PPI3D web server

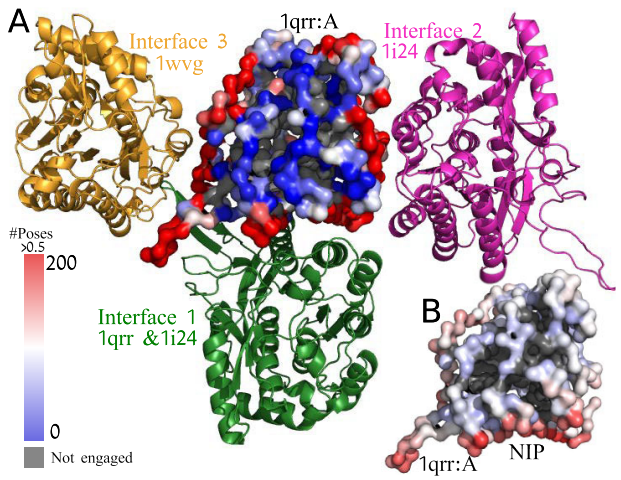


**Fig. 4.** Identification of multiple interaction interfaces for the SQD1 enzyme. (**A**) The surface of the protein (chain 1qrr_A) is colored according to the number of conformations (over a total of 3000) where each residue was found at the interface and was assigned a score higher than 0.5 by DLA-Ranker. Three red patches appear on the surface corresponding to: (i) interface 1 (partner in green, PDB codes: 1qrr, 1i24), (ii) interface 2 (partner in violet, PDB code: 1i24) and (iii) interface 3 (partner in gold, PDB code: 1wvg). (**B**) The Normalized Interface Propensity (NIP) shows the tendency of a residue to be part of an interaction site and computed by considering the fraction of docking poses where a residue is found at the interface (Dequeker *et al.*, 2019; Fernández-Recio *et al.*, 2004). It is plotted on 1qrr_A with a color scale going from red (high) to blue (low propensity), and highlights interface 1 but not interfaces 2 and 3, unlike DLA-Ranker (A color version of this figure appears in the online version of this article.)

(Dapkūnas *et al.*, 2017)] (Fig. 4A, partner in gold). Moreover, the third interface is evolutionary conserved and predicted as an interacting region by JET2Viewer (Ripoche *et al.*, 2017) (Supplementary Fig. S13B). Altogether, this analysis reveals that DLA-Ranker can be useful to detect multiple binding modes by evaluating individual residues across conformational ensembles. By comparison, looking only at the propensity of each residue to be located at the interface in the candidate conformations (Dequeker *et al.*, 2019; Fernández-Recio *et al.*, 2004), without accounting for DLA-Ranker scores, one can clearly identify the first interface but not the two others (Fig. 4B). We further compared the ability of ATTRACT and DLA-Ranker in identifying acceptable conformations representative of the different interfaces. ATTRACT and DLA-Ranker (based on the SCR score averaging scheme) find at least one acceptable hit for each of the two first interfaces in the top 22 and 28, respectively. This rank improves to 17 for DLA-Ranker if averaging scheme SC is used (Supplementary Fig. S13D and E).

### 3.6 Runtime and memory usage

The calculations were performed on two GPU clusters: (i) workstations with GPU: NVIDIA GeForce RTX 3090 (24 GB RAM) and CPU: AMD Ryzen 95950X and (ii) workstations with GPU: V100 (16 or 32 GB RAM). Training one network on all the conformations from 142 BM5 complexes on a single machine of the first cluster took 312 hours. There is no minimum GPU memory requirement. For example, for some experiments, we trained the models on an NVIDIA TITAN Xp (8 GB RAM) GPU. Nevertheless, a large GPU memory allows us to increase the batch size and speed up the learning process. The average inference time (representation of the interface and prediction of scores) is 0.45 s using CPU and 0.38 s using GPU for a conformation on a user's machine with GPU: NVIDIA Quadro RTX 3000 and CPU: Intel(R) Core(TM) i7-10875H CPU @ 2.30 GHz.

## 4 Discussion

We have shown that it is possible to evaluate complex candidate conformations by learning local 3D atomic arrangements at the interface. We have developed a deep learning-based approach explicitly accounting for the relative orientations of the protein residues while being insensitive to the global orientation of the protein. The method achieves performance better or similar to the state of the art. We obtained the best performance by averaging the per-residue scores predicted over the core and the rim of the interface. Beyond the results reported here, we have explored different aspects of the DLA-Ranker model by changing the input data representation, the network architecture, the hyperparameter values, and the hardware. Specifically, we tested the impact of reducing the number of atom types to 4 instead of the 167 default residue-specific atom types. We observed that the performance was not significantly impacted by this modification. The advantage of this model is that the calculation of the volumetric map, the training and the inference is much faster and with less hardware requirements (better usage of hard drive, RAM and GPU RAM). In addition, we tested the impact of removing the SCR interface description and the receptor-ligand distinction from the features. We observed that these two pieces of information, alone or combined, improved the performance. Increasing the number of layers and model parameters did not improve the performance and resulted in overfitting. The use of dropouts improved the performance.

DLA-Ranker can be applied to conformational ensembles generated by docking to identify near-native conformations and to discover alternative interfaces. It can be combined with more classical scoring functions. It can also be used to evaluate complexes of any size and is not limited to binary complexes. We envision many applications for the local-environment-based approach of DLA-Ranker, including the identification of physiological interfaces, the discovery of small subsets of cubes dedicated to functional tasks, the construction of phenotypic mutational landscapes and the prediction of binding affinity.

## Acknowledgements

We thank the Institute for Development and Resources in Intensive Scientific Computing (IDRIS-CNRS) for giving us access to their Jean Zay supercomputer. We thank Sergei Grudinin and his team for helping us with the initial source code of Ornate.

## Funding

From the French Agence Nationale de la Recherche: ANR-21-CE17-0046 SolvingMEFVariants (AC).

*Conflict of Interest*: none declared.

## Data availability

Data and software underlying this article are available at http://gitlab.lcqb.upmc.fr/dla-ranker/DLA-Ranker.git

## References

Abadi,M. *et al.* (2016) {TensorFlow}: a system for {Large-Scale} machine learning. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, *Savannah, USA*, pp. 265–283.

Baek,M. *et al.* (2021) Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, **373**, 871–876.

Berman,H.M. *et al.* (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.

Bryant,P. *et al.* (2022) Improved prediction of protein-protein interactions using AlphaFold2. *Nat. Commun.*, **13**, 1265.

Cao,Y. and Shen,Y. (2020) Energy-based graph convolutional networks for scoring protein docking models. *Proteins*, **88**, 1091–1099.

Cohen,T.S. and Welling,M. (2016) Steerable CNNs. *arXiv:1612.08498 [cs, stat]*. https://doi.org/10.48550/arXiv.1612.08498.

Corsi,F. *et al.* (2020) Multiple protein-DNA interfaces unravelled by evolutionary information, physico-chemical and geometrical properties. *PLoS Comput. Biol.*, **16**, e1007624.

Dapkūnas,J. *et al.* (2017) The PPI3D web server for searching, analyzing and modeling protein–protein interactions in the context of 3D structures. *Bioinformatics*, **33**, 935–937.
Dequeker,C. *et al.* (2017) INTerface builder: a fast protein–protein interface reconstruction tool. *J. Chem. Inf. Model.*, **57**, 2613–2617.
Dequeker,C. *et al.* (2019) Decrypting protein surfaces by combining evolution, geometry, and molecular docking. *Proteins*, **87**, 952–965.
Dequeker,C. *et al.* (2022) From complete cross-docking to partners identification and binding sites predictions. *PLoS Comput. Biol.*, **18**, e1009825.
Dominguez,C. *et al.* (2003) HADDOCK: a protein protein docking approach based on biochemical or biophysical information. *J. Am. Chem. Soc.*, **125**, 1731–1737.
Eismann,S. *et al.* (2021) Hierarchical, rotation-equivariant neural networks to select structural models of protein complexes. *Proteins*, **89**, 493–501.
Evans,R. *et al.* (2021) Protein complex prediction with AlphaFold-Multimer. bioRxiv, https://doi.org/10.1101/2021.10.04.463034.
Fernández-Recio,J. *et al.* (2004) Identification of protein–protein interaction sites from docking energy landscapes. *J. Mol. Biol.*, **335**, 843–865.
Fuchs,F.B. *et al.* (2020) SE(3)-transformers: 3D Roto-Translation equivariant attention networks. *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*. https://doi.org/10.48550/arXiv.2006.10503.
Ghani,U. *et al.* (2021) Improved docking of protein models by a combination of Alphafold2 and ClusPro. *bioRxiv.* https://doi.org/10.1101/2021.09.07.459290.
Hubbard,S. and Thornton,J. (1993). *NACCESS, Computer Program*. Department of Biochemistry and Molecular Biology, University College, London.
Humphreys,I.R. *et al.* (2021) Computed structures of core eukaryotic protein complexes. *Science*, **374**, eabm4805.
Jumper,J. *et al.* (2021) Highly accurate protein structure prediction with AlphaFold. *Nature*, **596**, 583–589.
Kozakov,D. *et al.* (2017) The ClusPro web server for protein–protein docking. *Nat. Protoc.*, **12**, 255–278.
Kundrotas,P.J. *et al.* (2018) Dockground: a comprehensive data resource for modeling of protein complexes. *Protein Sci.*, **27**, 172–181.
Lagarde,N. *et al.* (2018) Hidden partners: using cross-docking calculations to predict binding sites for proteins with multiple interactions. *Proteins*, **86**, 723–737.
Laine,E. and Carbone,A. (2015) Local geometry and evolutionary conservation of protein surfaces reveal the multiple recognition patches in protein-protein interactions. *PLoS Comput. Biol.*, **11**, e1004580.
Laine,E. and Carbone,A. (2017) Protein social behavior makes a stronger signal for partner identification than surface geometry. *Proteins*, **85**, 137–154.
Lensink,M.F. *et al.* (2007) Docking and scoring protein complexes: CAPRI 3rd edition. *Proteins*, **69**, 704–718.
Lensink,M.F. *et al.* (2017) Modeling protein–protein and protein–peptide complexes: CAPRI 6th edition. *Proteins*, **85**, 359–377.
Lensink,M.F. *et al.* (2020) Modeling protein-protein, protein-peptide, and protein-oligosaccharide complexes: CAPRI 7th edition. *Proteins*, **88**, 916–938.
Levy,E.D. (2010) A simple definition of structural regions in proteins and its use in analyzing interface evolution. *J. Mol. Biol.*, **403**, 660–670.
Liu,S. *et al.* (2008) Dockground protein–protein docking decoy set. *Bioinformatics*, **24**, 2634–2635.
Lopes,A. *et al.* (2013) Protein-protein interactions in a crowded environment: an analysis via cross-docking simulations and evolutionary information. *PLoS Comput. Biol.*, **9**, e1003369.
Mintseris,J. *et al.* (2005) Protein–protein docking benchmark 2.0: an update. *Proteins*, **60**, 214–216.
Mirdita,M. *et al.* (2022) ColabFold—making protein folding accessible to all. Nat. Meth., **19**, 679–682.
Nadalin,F. and Carbone,A. (2018) Protein–protein interaction specificity is captured by contact preferences and interface composition. *Bioinformatics*, **34**, 459–468.
Neveu,E. *et al.* (2018) RapidRMSD: rapid determination of RMSDs corresponding to motions of flexible molecules. *Bioinformatics*, **34**, 2757–2765.
Pagès,G. *et al.* (2019) Protein model quality assessment using 3D oriented convolutional neural networks. *Bioinformatics*, **35**, 3313–3319.
Raucci,R. *et al.* (2018) Local interaction signal analysis predicts protein-protein binding affinity. *Structure*, **26**, 905–915.e4.
Renaud,N. *et al.* (2021) DeepRank: a deep learning framework for data mining 3D protein-protein interfaces. *Nat. Commun.*, **12**, 7068.
Ripoche,H. *et al.* (2017) JET2 viewer: a database of predicted multiple, possibly overlapping, protein–protein interaction sites for PDB structures. *Nucleic Acids Res.*, **45**, D236–D242.
Ritchie,D.W. and Venkatraman,V. (2010) Ultra-fast FFT protein docking on graphics processors. *Bioinformatics*, **26**, 2398–2405.
Réau,M. *et al.* (2021) DeepRank-GNN: a graph neural network framework to learn patterns in Protein-Protein interfaces. *bioRxiv*, https://doi.org/10.1101/2021.12.08.471762.
Sacquin-Mora,S. *et al.* (2008) Identification of protein interaction partners and protein–protein interaction sites. *J. Mol. Biol.*, **382**, 1276–1289.
Sato,R. and Ishida,T. (2019) Protein model accuracy estimation based on local structure quality assessment using 3D convolutional neural network. *PLoS One*, **14**, e0221347.
Thomas,N. *et al.* (2018) Tensor field networks: rotation- and translation-equivariant neural networks for 3D point clouds. https://doi.org/10.48550/arXiv.1802.08219.
Tovchigrechko,A. and Vakser,I.A. (2005) Development and testing of an automated approach to protein docking. *Proteins: Structure, Function, and Bioinformatics*, **60**, 296–301.
Veličković,P. *et al.* (2018) Graph attention networks. In: *International Conference on Learning Representations (ICLR 2018), Vancouver, Canada*. https://doi.org/10.48550/arXiv.1710.10903.
Vreven,T. *et al.* (2015) Updates to the integrated protein–protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2. *J. Mol. Biol.*, **427**, 3031–3041.
Wang,X. *et al.* (2020) Protein docking model evaluation by 3D deep convolutional neural networks. *Bioinformatics*, **36**, 2113–2118.
Wang,X. *et al.* (2021) Protein docking model evaluation by graph neural networks. *Front. Mol. Biosci.*, **8**, 647915.
Weiler,M. *et al.* (2018) 3D steerable CNNs: learning rotationally equivariant features in volumetric data. In: *Advances in Neural Information Processing Systems*, Vol. **31**, Neural Info Process Sys F Publisher, La Jolla.
Zacharias,M. (2003) Protein–protein docking with a reduced protein model accounting for side-chain flexibility. *Protein Sci.*, **12**, 1271–1282.
Zhang,Y. and Skolnick,J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.*, **33**, 2302–2309.