ProNA2020 predicts protein–DNA, protein–RNA, and protein–protein binding proteins and residues from sequence

Jiajun Qiu [1,2], Michael Bernhofer [1,2], Michael Heinzinger [1,2], Sofie Kemper [1], Tomas Norambuena [3], Francisco Melo [3,4] and Burkhard Rost [1,5,6,7]

*1 - **Department of Informatics,** I12-Chair of Bioinformatics and Computational Biology, Technical University of Munich (TUM), Boltzmannstrasse 3, 85748, Garching, Munich, Germany*
*2 - **TUM Graduate School,** Center of Doctoral Studies in Informatics and Its Applications (CeDoSIA), Garching, 85748, Germany*
*3 - **Molecular Bioinformatics Laboratory,** Facultad de Ciencias Biológicas, Pontificia Universidad Católica de Chile, Santiago, Chile*
*4 - **Institute of Biological and Medical Engineering,** Pontificia Universidad Católica de Chile, Santiago, Chile*
*5 - **Columbia University,** Department of Biochemistry and Molecular Biophysics, 701 West, 168th Street, New York, NY, 10032, USA*
*6 - **Institute of Advanced Study (TUM-IAS),** Lichtenbergstr. 2a, 85748, Garching/Munich, Germany*
*7 - **Germany & Institute for Food and Plant Sciences (WZW) Weihenstephan,** Alte Akademie 8, 85354 Freising, Germany*

***Correspondence to Jiajun Qiu:** Fax: +49 (89) 289 19414.* jiajunqiu@hotmail.com

https://doi.org/10.1016/j.jmb.2020.02.026
Edited by Rita Casadio

Abstract

The intricate details of how proteins bind to proteins, DNA, and RNA are crucial for the understanding of almost all biological processes. Disease-causing sequence variants often affect binding residues. Here, we described a new, comprehensive system of *in silico* methods that take only protein sequence as input to predict binding of protein to DNA, RNA, and other proteins. Firstly, we needed to develop several new methods to predict whether or not proteins bind (per-protein prediction). Secondly, we developed independent methods that predict which residues bind (per-residue). Not requiring three-dimensional information, the system can predict the actual binding residue. The system combined homology-based inference with machine learning and motif-based profile-kernel approaches with word-based (ProtVec) solutions to machine learning protein level predictions. This achieved an overall non-exclusive three-state accuracy of 77% ± 1% (±one standard error) corresponding to a 1.8 fold improvement over random (best classification for protein–protein with F1 = 91 ± 0.8%). Standard neural networks for per-residue binding residue predictions appeared best for DNA-binding (Q2 = 81 ± 0.9%) followed by RNA-binding (Q2 = 80 ± 1%) and worst for protein–protein binding (Q2 = 69 ± 0.8%). The new method, dubbed ProNA2020, is available as code through *github* (https://github.com/Rostlab/ProNA2020.git) and through PredictProtein (www.predictprotein.org).

Introduction

Physical interactions between proteins and large DNA, RNA, and proteins crucially determine all essential biological processes, including mechanisms relevant for health and disease [1,2]. The development of new drugs requires detailed molecular understanding of the binding residues [3]. Typically, binding residues are only available through the detailed three-dimensional (3D) structure of a protein. UniProt now (Dec. 2019) contains 179 million protein sequences [4], of which, fewer than 0.36% contain the experimental protein structure data from X-ray crystallography and NMR spectroscopy in the Protein Database, PDB [5], whereas good 3D models of structures are available for fewer than 20% of all the residues of all known proteins [6]. For all of those, binding residues remain largely unknown. However, even knowing which residues are involved in binding without knowing the binding pocket or any details of the 3D structure might already help in designing experiments. Often, it might already help to know that a protein binds to DNA|RNA or other proteins. Despite the pivotal importance of transient physical protein–-protein interactions (PPIs), some important proteins appear not to bind *in vivo* to any other protein [1]. Possibly 6–8% of all proteins in a eukaryote might bind RNA (**RBPs**: RNA-binding proteins) [7]. For eukaryotes, the fraction of DNA-binding proteins (**DBPs**) appears similar to that of RBPs (6–7%) [8]; for prokaryotes, typically 2–3% of a genome encodes DBPs [8].

Typically, proteins binding other proteins, DNA, or RNA form the targets of structure-based drug design [9]. Understanding protein binding residues becomes a basis for structure-based drug design. Drug molecules usually affect the interaction between the target protein and its ligand [10]. However, fewer than 0.36% of all proteins of known sequence in UniProt correspond to a known experimental 3D structure in the PDB [4,5]. Therefore, it is essential to build computational tools to reliably and rapidly identify protein-, DNA- and RNA-binding proteins or residues.

Given that structure annotations remain missing for most proteins (for >120 million in June 2019), there continues to be a high demand even for low-resolution predictions of aspects pertaining to proteins binding protein, DNA, and RNA from sequence alone. Not surprisingly, many *in silico* methods cater to this need and predict binding proteins (protein binds or not) or binding residues (which residues bind) from sequence. These include (sorted by date) methods optimized for per-protein predictions (protein binds or not) DNABIND [11], SomeNA [12], and StackDPPred [13] for DNA binding, and RBPPred [14], SPOT-RNA [15] and TriPepSVM [16] for RNA binding. Other aspects are provided by tools optimized for per-residue predictions (predicting which residues bind), including some that predict binding for DNA and RNA (sorted by date): DRNApred [17] and NucBind [18], and others capturing all three targets: hybridNAP [19] and DisoRDPbind [20]. The later predicts binding in intrinsically unstructured proteins. However, we are not aware of any existing method combining machine learning prediction and homology-based inference of per-protein and per-residue binding for the three most important large macromolecules (PPI, DNA, or RNA) into one comprehensive system.

Here we present a novel sequence-based system for the comprehensive identification of proteins that bind to protein, DNA, and RNA and the prediction of the residues involved in binding. One crucial novelty of this work is the demonstration that per-protein predictions are performed only very poorly by methods optimized on per-residue predictions, i.e. users need different tools to predict which protein binds a protein, DNA or RNA (per-protein) and where it binds (per-residue) if it does. Toward this end, we also demonstrate how very different machine learning methods can be combined best and how predictions without using evolutionary information may contribute to performance. Another methodological novelty was the embedding of natural language processing (NLP) concepts [21]. Our new system has three major advantages over some existing approaches. Firstly, it combines and assesses per-protein and per-residue prediction in the same framework. All prediction methods are grafted into a common framework although they require very different individual solutions. Secondly, it combines homology-based inference with machine learning (also done by: DisoRDPbind [20]). Thirdly, all the three major macromolecules (protein, DNA, and RNA) are integrated into one hierarchical prediction with sustained performance estimates for the entire system (also done by hybridNAP [19] and DisoRDP-bind [20]).

Materials and Methods

Data sets

Reducing sequence redundancy in data sets

For all data sets, UniqueProt reduced redundancy such that no protein pair in the set had sequence similarity of HVAL>0 [22] (e.g. corresponding to 20% pairwise sequence identity for alignments longer than 250 residues) or PSI-BLAST E-value>10^{-3} with the minimum alignment length of 45 residues [22]. Redundancy was reduced to avoid overestimating performance [23].

Data sets for per-residue information (PPI, DNA, and RNA)

DNA-protein binding data was extracted from the Protein–DNA Interface Database (PDIdb, version April 2010 [24]). PDIdb contained 992 entries of proteins with high-resolution 3D structure from the Protein Data Bank (PDB [5] with 1317 different protein chains binding DNA. **RNA-protein** binding data was extracted from the Protein–RNA Interface Database (PRIDB, version RB1179 [25]). PRIDB contained 1179 non-redundant PDB protein chains binding RNA. All PDB entries were mapped to UniProtKB sequences using SIFTS [4,26]. Only 3D structures from X-ray crystallography with resolutions <2.5 Å (0.25 nm) were included; DNA or RNA (in the following NA) interactions were considered only when the closest pair of atoms (between protein and NA) was within 6Å (0.6 nm). **Protein-Protein binding** data was provided by Tobias Hamp [27]. Structures were obtained from PDB (2015) with a resolution of <2.5 Å. After removing all structures from the PPI set mapping to fewer than two different UniProtKB IDs and the proteins with fewer than five residues within 6 Å (0.6 nm) of any atom of the other protein, the protein–protein binding data sets contained 3957 PPIs from 2914 unique proteins representing the species diversity of the PDB. Although reducing redundancy, we maintained alternative binding residues. Assume, A–B (A binds B), A–B′, and EVAL(B,B′)>T, EVAL(A,B)<T, EVAL(A,B′)<T (where T is the threshold for redundancy reduction; EVAL(A,B) the PSI-BLAST Expectation-value, or E-value, for the alignment between A and B). We removed B′ from the data set, but kept the labels of "interacting residues" on A marked by the interaction A–B′. We deliberately did not consider homo-dimers

assuming that they bind in a biophysically different manner from the type of transient physical PPIs that the prediction method targeted [28]. All data sets are available through github (https://github.com/Rostlab/ProNA2020.git); statistics are provided in Tables 1 and 2.

Data sets for per-protein information

Besides the proteins used in per-residue data set, proteins with the experimental annotations were also collected in positive data set for per-protein (described in the next section). Total numbers of non-redundant proteins: protein binding/not binding: 524/282, DNA-binding/not DNA|RNA-binding: 199/555, RNA-binding/not DNA|RNA-binding: 263/555 (Table 2).

GO annotations for negatives (only per-protein)

Due to a variety of reasons, experimentally characterized negatives are rare. To compensate for that, we used GO annotations [29] with experimental evidence codes as proxies for negatives and those used for homology-based inference. We collected proteins with the experimental annotations of protein binding (GO:0005515), DNA-binding (GO:0003677), and RNA-binding (GO:0003723). All proteins with neither of those three, nor with any indirect annotations (keywords: *DNA*, *RNA*, *nucleotide*) served as negatives. This procedure was only applied for per-protein predictions (e.g. protein binds DNA or not). For all per-residue predictions (e.g. which residues bind DNA), all residues NOT annotated to bind in a particular PDB chain (e.g. DNA) served as negatives.

Independent data sets for comparisons to existing methods

In order to compare our new method to others, we built new sets without sequence redundancy (HVAL < 0 [22]) to the proteins used for developing our method. We also applied another HVAL < 5 filter to rule out possible overlap between any protein used for testing ProNA2020 components and those proteins used to develop the prediction methods used as input through the PredictProtein [30] server; this applied in particular to predicted secondary structure and solvent accessibility. The advantage of this solution was that we could compare tools based on the same data sets for proteins not similar to those used for development. The problem was that these rigorous

Table 1. Non-redundant[a] cross-validation[b] set for per-residue predictions.[c]

	No. of binding residues	No. of non-binding residues	No. of all residues	Percentage binding
Protein-binding residues	29,438	78,608	108,046	27.2%
DNA-binding residues	6644	19,227	25,871	25.7%
RNA-binding residues	8588	21,538	30,126	28.5%

[a] For all data sets, UniqueProt reduced redundancy such that no protein pair in the set had sequence similarity of HVAL > 0 (corresponding to 20% pairwise sequence identity for alignments longer than 250 residues).

[b] Cross-validation: We separated the whole development/cross-validation set into five parts. Training used three of five (training set); one of five (cross-training set) was used to optimize hyper-parameters (incl. different input feature combinations, window sizes, combinations of methods). For all decisions, optimal was defined as the highest F1 score. The last of the five was used to evaluate the performance of the final model (testing set). The sets were rotated five times such that each protein in the data set had been used for testing (and cross-training) exactly once.

[c] Per-residue prediction: prediction of which residue in a protein binds DNA|RNA|protein (or combinations thereof). All residues NOT observed to bind were considered NOT binding.

Table 2. Non-redundant[a] cross-validation[b] set for per-protein predictions.[c,d]

Data set	Number of binding proteins
Protein-binding proteins	524
Negative for protein-binding proteins	282
DNA-binding proteins	199
RNA-binding proteins	263
Negative for DNA and RNA-binding proteins	555
Overlap between protein-binding negative and DNA/RNA-binding negative[a]	108

[a] For all data sets, UniqueProt reduced redundancy such that no protein pair in the set had sequence similarity of HVAL > 0 (corresponding to 20% pairwise sequence identity for alignments longer than 250 residues).

[b] Cross-validation: We separated the whole development/cross-validation set into five parts. Training used three of five (training set); one of five (cross-training set) was used to optimize hyper-parameters (incl. different input feature combinations, window sizes, combinations of methods). For all decisions, optimal was defined as the highest F1 score. The last of the five was used to evaluate the performance of the final model (testing set). The sets were rotated five times such that each protein in the data set had been used for testing (and cross-training) exactly once.

[c] Per-protein prediction: prediction that a protein binds DNA|RNA|protein (or combinations thereof) as opposed to where it binds, i.e. the binding residues. Toward this task, we need to consider a representative data set of proteins NOT binding.

[d] When testing the performance of the whole system, the overlap between neither protein-binding nor DNA/RNA-binding served as the data set for non-binding.

constraints resulted in relatively small sets. PDB sequences from 2010 were selected to assess DNA- and RNA-binding; PDB sequences from 2016 for PPI. All data sets were processed (resolution, distance threshold, and redundancy reduction) in the same way as the development data sets (Tables 1 and 2), namely: PDB resolutions <2.5 Å; binding residues within 6 Å of molecule (statistics in Table 3). PISA server is used to define the biological interface [31].

Prediction methods

Homology-based inference

Homology-based inference refers to the following process. Assume that a particular phenotype (e.g. protein binds DNA) is known for protein X, and that protein U has a sequence similarity to X exceeding some threshold (EVAL(U,X)>T), above which the phenotype is typically conserved between evolutionarily related proteins. Then we will infer that U has the same phenotype as X (e.g. U also binds DNA). The alignments for homology-based inference were generated by PSI-BLAST using the following standard protocol implemented, e.g. in the PredictProtein Server [30]. For each protein, build the PSI-BLAST profile using an 80% non-redundant database combining UniProt and PDB (two iterations, inclusion threshold E-value $\leq 10^{-3}$). These profiles were then aligned against all proteins with experimental annotation of binding (proteins have experimental annotations of protein binding (GO:0005515), DNA-binding (GO:0003677), and RNA-binding (GO:0003723))(inclusion E-value $\leq 10^{-3}$). PSI-BLAST hits to the protein in the test set were excluded to avoid over-estimate [32].

Cross-training and testing

All hyper-parameter optimizations were done on the cross-training sets. This included the choice of alternative machine learning methods (e.g. between profile-kernel SVM and ProtVec Local). All results for the final estimates of performance were compiled either on the test set or on the independent test set. No parameter was optimized on these. For instance, the decision to combine SVM and ProtVec Local on each node of the per-protein level prediction rather than to use the single best at each node (Fig. 1) appeared optimal for the cross-training set, not for the independent test set (we did provide the estimate for the combination, i.e. not the one performing best in comparison to other methods). Overall, different parts from the identical data set served as training, cross-training, and testing sets; all were rotated through so that every protein in the redundancy-reduced set was used for testing exactly once and for cross-training exactly once, implying that the cross-training and testing sets were identical (Fig. S1): five-fold cross-validation was accomplished by using three splits of the data for training, one for cross-training (optimize hyper-parameters, including number of hidden units in NN, early stop) and one for testing. Overall, we optimized the parameters (such as the number of node, learning rate for NN; *k*-mer, σ for profile-kernel) and features for residue-level prediction in the cross-training set and tested the final performance on the testing set. This implied that we actually trained five different machine learning models for each task, and that each protein from the main development data set was used for testing/cross-training exactly once. We picked the optimal hyper-parameters with best average performance in cross-training splits. This along with avoiding feature-selection decreased the likelihood of over-fitting. In fact, the choice of input units essentially followed what had been best for earlier methods developed in our lab.

Random prediction

All performance values were compared to random predictions. A random prediction was created by choosing a random number between 0 and 1, if >0.5, the residue was predicted as binding. The random per-protein predictions used the same tree-like hierarchical prediction system as the machine learning method (Fig. 1).

Prediction methods

When training the various machine learning models, protein binding and nucleotide binding were considered as separate tasks solved by two different systems of decision trees (Fig. 1, Table S1; each node represented one binary machine learning model typically trained on different data sets with different inputs and outputs).

(1) **Per-protein: profile-kernel SVM.** Support Vector Machines (SVMs) were implemented through WEKA [33]. The profile-kernel function

Table 3. Non-redundant[a] independent[b] test data set.

	For per-protein predictions		For per-residue predictions	
	No. of binding proteins	No. of non-binding proteins	No. of binding residues	No. of non-binding residues
Protein-binding	209	52	5174	10,447
DNA-binding	109	152	3645	8345
RNA-binding	57	204	1444	4711

[a] For all data sets, UniqueProt reduced redundancy such that no protein pair in the set had sequence similarity of HVAL > 0 (corresponding to 20% pairwise sequence identity for alignments longer than 250 residues). In addition, none of those proteins had HVAL > 0 to any protein used for development of any of the methods compared.

[b] Independent test set refers to the fact that those experimental measurements have become available AFTER the data sets used for the development of ProNA2020. Again not only were those proteins new, they also differed significantly in terms of sequence similarity (HVAL < 0) to any that had been available before.

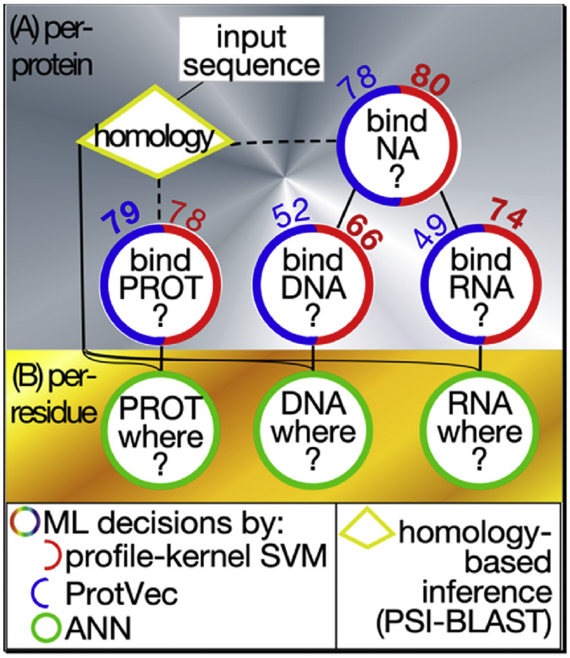

Fig. 1. Hierarchical prediction system. The branches represent the paths for the protein sorting, the nodes mark particular prediction methods (circles: machine learning (ML) models, rhombus: homology-based inference. Full lines mark part of the hierarchy the system will follow (higher in the image: earlier in the processing hierarchy). In contrast, dashed lines (from the homology-based inference) are those that might lead to bypass full lines. (A) Per-protein: The top silver gray panel is the major novelty of this contribution, namely the integration of modules specialized for per-protein level prediction. These are four ML modules predicting whether a query binds any: nucleotide (NA), proteins (PROT), DNA, or RNA. The values above the red/blue ML nodes give the F1 score of profile-kernel SVMs (red) and ProtVec (blue) based on the cross-training set (best method in bold numbers). (B) Per-residue: The lower gold panel marks per-residue predictions that have been integrated into servers before. The green circles mark three separate prediction methods predicting which residues bind PROT, DNA, and RNA. Proteins are filtered through the per-protein prediction on top and passed only to the module found appropriate by the previous step. Upon request, the sorting can be bypassed if users know the binding mode (PROT|DNA| RNA) of the query protein.

mapped the PSSM profile of each protein family to a vector indexed by all possible subsequences of length *k* from the alphabet of amino acids. Another parameter σ in the profile-kernel SVM was the threshold to decide when a particular k-mer was considered to be conserved in the multiple sequence alignment (family) or not. So each element in the final vector represented one particular *k*-mer and its score gave the number of occurrences of this *k*-mer that was below a certain user-defined threshold σ. The dot product between two *k*-mer vectors reflected the similarity of two protein sequence profiles. The best combinations of profile kernel parameters (k, σ) and of SVMs were found through 5-fold cross-validation [32–34].

(2) **Per-protein: protein vectors (ProtVec).** Continuous vector representation, as a distributed representation for words, has been recently established in NLP as an efficient way to capture semantic/syntactic units [21,35]. The basic underlying idea is to elucidate the meaning of a word through its context, i.e. neighboring words. Words with similar vectors show multiple degrees of similarity. For instance, *vector(king)* − *vector(man)* + *vector(woman)* is closest to *vector(queen)* [21,35].

The method ProtVec [21,35] applies this concept of so-called skip-gram natural language models to protein sequences. In this way, consecutive amino acids are grouped into words and the whole protein sequence becomes a sentence described by an n-dimensional vector by considering contexts of different size (i.e. word lengths). These n-dimensional vectors were input into the downstream machine learning.

We used the Word2Vec [21,35] to re-implement our own version of *ProtVec* (referred to as *ProtVec Local*). Parameters optimized included the dimensionality of the feature vectors (size), the maximum distance between words within a sentence (window), and the minimum number of the words (min_count). We also tested different word lengths *k* of consecutive residues (k-mer, e.g. the enzyme lactase begins with the 3-mer MEL), and whether or not to use the feature "phrase". Using "phrase" implied to automatically detect common phrases (multiword expressions) from a stream of sentences. The best combination was found by five-fold cross-validation [21,35]. For the subsequent machine learning algorithm, we compared SVM, Random Forests (RF), and Neural Networks (NN).

(3) **Per-residue: neural networks and smoothing filter.** Following earlier publications [2,36], we applied a two-step process to predict per-residue binding residues. First level: We trained standard feed-forward neural networks with back-propagation and momentum term using the sliding-window approach as input (for a window size of w, when predicting for residue j, all residues from j − INT(w/2) to j + INT(w/2) were included). All input features were taken from PredictProtein [30] including, but not limited to, predicted secondary structure, predicted relative solvent accessibility, and biophysical properties of amino acids. The combinations of features and other hyper-parameters

(e.g. window sizes and hidden units) were optimized on the cross-training set using the F1 score (complete list of features: SOM Tables S2 snd S3). Second level: The final prediction score for a residue was calculated by the average of the positive values in the certain window as follows:

$$score = \frac{1}{\omega}\sum_{i=-(\omega-1)/2}^{(\omega-1)/2} raw\_score_i, (raw\_score_i > 0) \qquad (1)$$

Reliability index (prediction strength)

The reliability (or strength) of a prediction was described through a *reliability index* (RI) ranging from 0 (weak prediction) to 100 (confident prediction). For per-protein predictions, the RIs were computed directly from the machine learning output. For per-residue predictions, the RIs were computed from the second-level scores (Eq. (2)). For homology-based inferences from PSI-BLAST, RIs were compiled from the percentage pairwise sequence identity (PIDE). As in our settings PSI-BLAST did not find any relations at PIDE < 10%, prediction performance did not change for PIDE ≤ 10 (Fig. S4). Thus, RIs were re-normalized accordingly [32].

Performance evaluation

Many publications fall short of comprehensively assessing performance through a diversity of measures [37,38]. While we tried to avoid this pitfall, we also tried to confine additional analyses that only confirmed previous results to the Supporting Online Material (SOM) wherever possible to eschew obfuscation.

Proteins might bind more than one target. Thus, we intrinsically had to assess a multi-class problem. For several aspects of the evaluation, we simplified by calculating the per-protein performance for each class, by only considering that class. With the standard acronyms (TP: true positives, observed and predicted in class C; TN: true negatives, observed and predicted in non-C; FP: false positives: predicted in C, observed in non-C; FN: false negatives: predicted in non-C, observed in C), we applied the standard definitions:

$$\begin{aligned} PRE(C) &= PrecisionC \\ &= TP/(TP + FP); REC(C) \\ &= RecallC = TP/(TP + FN); \end{aligned}$$

$$\begin{aligned} &NPV(C) = TN/(TN+FP); TNR(C) = TN/(TN+FN) \\ &F1(C) = 2*PRE(C)*REC(C)/(PRE(C)+REC(C)) \end{aligned} \qquad (2)$$

We also provided the confusion matrix containing the raw values for TP, TN, FP, and FN for the test set of each of our methods separately. Toward this end, we only provided results for the cross-validation test set due to the larger data set size. These raw numbers are particularly relevant to correct for overall estimates [39]; for that correction, estimates based on larger data sets appear most helpful. In addition, we monitored the overall two-state accuracy (Q_2) and the Matthews correlation coefficient (MCC):

$$\begin{aligned} &Q2 = (TP + TN)/(TP + TN + FP + FN) \\ &MCC(C) = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \end{aligned} \qquad (3)$$

The overall non-exclusive three-class accuracy on the protein level was defined as:

$$Accuracy(A) = \frac{1}{n}\sum\nolimits_{i=1}^{n} \frac{|prd_i \cap obs_i|}{|prd_i \cap obs_i|} \qquad (4)$$

where $prd_i|obs_i$ are the numbers of classes predicted|observed for protein *i.* For instance, if protein A binds DNA and other proteins, and the prediction is *RNA&Protein* binding, the Accuracy(A) would be 1/3; the random prediction would reach $A_{random} = 43 \pm 1\%$.

Family size comparison

The number of sequences in each protein family was obtained from https://pfam.xfam.org/. For a protein with multiple families, the largest family was assigned.

Error estimates

Error rates for the evaluation measures were estimated by bootstrapping [40] (without replacement to render more conservative estimates), i.e. by re-sampling the set of proteins/residues used for the evaluation 1000 times and calculating the standard deviation over those 1000 different results. Each of these sample sets contained 50% of the original proteins/residues (picked randomly, again: without replacement).

Method comparison

We did compare performance with other methods task by task using the following publicly available methods. For DNA binding, these were DNAbinder [41], DNABIND [11], NucBind [18], SomeNa [12], and StackDPPred [13]. For RNA binding, these were RNABindRPlus [42], RBPPred [14], SomeNa [12],SPOT-RNA [15], and TriPepSVM [16]. For protein binding, these were BSpred [43], iPPBS-

PseAAC [44], InteractionSites [36], LORIS [45], PPIS [46], and SPRINGS [47]. The following multi-class binding prediction methods were included: DisoRDPbind [20], DRNApred [17], hybridNAP [19], and NucBind [18]. One important novelty of this work is the finding that different machine-learning methods are needed to predict where a protein binds (per-residue level), and whether a protein binds (per-protein level). Toward this end, we can turn a method optimized for the per-residue level into a per-protein prediction by simply considering that the method predicted the protein not to bind if no residue was predicted as binding (modes of assessment summarized in Table 4).

Results

Tree-like hierarchy for prediction system complicates assessment

We implemented an intuitive tree-like hierarchy for the entire per-protein prediction system (Fig. 1). While the system was not optimized for performance, at each node in the hierarchy (Fig. 1), we tried different solutions for the machine learning and for the combination of machine learning and homology-based inference (Methods). Methods were assessed on their specific tasks and on how they performed embedded into the hierarchy (Table 4). For instance, assume the *DNA-binding* ML module correctly predicts protein P to bind DNA. Assume further that the first module *nucleotide-binding* made a mistake (Fig. 1: top right circle, Table 4: unknown binding mode). Then the *DNA-binding* module would never be activated, i.e. the system would classify incorrectly although the isolated module was indeed correct. Both aspects needed assessment because users might over-ride some components of the system. All decisions (hyper-parameter optimizations) were done on the cross-training set (Methods), NOT on the test set.

Per-protein: profile-kernel SVM and ProtVec best together

We created two versions of machine-learning classifications for each node in our protein level prediction tree-like hierarchy (Fig. 1, Tables S4 and S5): one used a profile kernel SVM and the other the skip-gram like *ProtVec* approach. For each node, the better solution was identified on the cross-training set (Fig. 1: values above circles valid for cross-training). Thus, the performance values were relevant only to set up the final system. For some tasks, ProtVec performed better (Fig. 1: blue values, numerically higher for protein binding); however, for most, the profile kernel SVM did (Fig. 1: red values, significantly better for DNA- and RNA-binding). The best result originated from running both methods for a protein and then choosing the one with the higher score. Overall, the profile-kernel performed better on proteins from larger families (Fig. 2, $P = 0.05$).

Homology-based inference embedded into the prediction system

Merging machine learning directly with homology-based inference might improve both [32]. We measured sequence similarity through PSI-BLAST at a threshold of $T = 10^{-15}$, i.e. the annotation was inferred for a query protein Q if its sequence similarity to a protein of known binding K was below T (PSI-BLAST expectation E-value(Q,K) < 10^{-15}; Fig. S2). For combination, we used homology-based inference (PSI-BLAST) where available (below threshold $T < 10^{-15}$), and machine learning prediction, otherwise. This combination outperformed the machine learning method, reaching an overall performance of 77 ± 1% (Eq. (4)). For all three classes, the combined predictions improved over machine-learning (Fig. S3, Table S6) and significantly over random (Fig. 3A, Table S7).

Per-residue predictions

All per-residue prediction methods were standard two-layer feed-forward neural networks, trained exclusively on a subset of protein from each class (e.g. to learn the prediction of DNA-binding residues, only proteins observed to bind DNA were used). There are two ways to assess the final system. Firstly, we measured performance for proteins known to e.g. bind DNA. Toward this end, each prediction task was tested separately, e.g. when

Table 4. Summary of three prediction modes.

	Performance measures	Description
Protein sorting mode	Accuracy, Q_2, PRE, REC, NPV, TNR, F1, MCC	Per-protein level prediction
Residue known binding mode	Q_2, PRE, REC, NPV, TNR, F1, MCC	Per-residue level prediction for proteins for which it is known THAT they bind protein/DNA/RNA for which the residue is predicted (no sorting needed)
Residue unknown binding mode	Q_2, PRE, REC, NPV, TNR, F1, MCC	Per-residue level prediction for proteins for which it is NOT known what they bind and for which the residue is predicted (mistakes in protein sorting are added to mistakes in per-residue prediction)

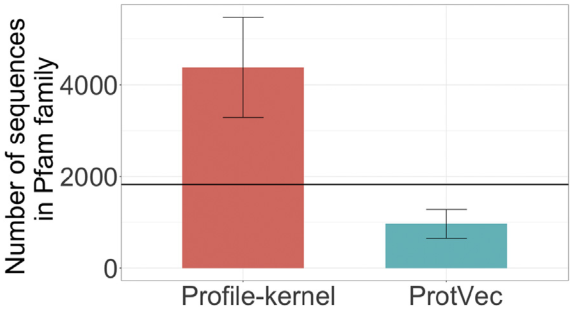

Fig. 2. Correct predictions exclusive to profile-kernel SVM vs. ProtVec. Bases for this plot are all proteins correctly predicted by only one of the two per-protein prediction algorithms, namely either by the profile-kernel SVM or by the ProtVec. The y-axis shows the average number of family members in each of the families. The horizontal black line gives the average over all families. Clearly, the profile-kernel SVMs do better for unusually large families, while the ProtVec tends to win for unusually small families.

testing DNA-binding, all DNA-binding proteins were assessed with respect to per-residue performance and all proteins experimentally known to bind DNA and those known not to bind for per-protein performance. This constitutes the standard way in which all other methods have been tested (Fig. 3A, B, D). The 2nd level filter smoothened spikes (Eq. (1) averaging over adjacent residues); it increased precision (Eq. (2)) to PRE(protein) = 46 ± 0.3% (from 35 ± 0.2% without filter), to PRE(DNA) = 57 ± 0.6% (from 48 ± 0.4%), and to PRE(RNA) = 54 ± 1% (from 46 ± 1%; Tables S8 and S5). DNA residue-binding reached the highest MCC (0.42 ± 0.006), followed by RNA residue-binding (MCC = 0.36 ± 0.006) and protein residue-binding (MCC = 0.25 ± 0.003 Fig. 3D, Tables S8 and S5). The MCC improvement was similar (Eq. (2); Fig. 3B). The improvement over random was again highest for DNA-binding (Fig. 3B, Tables S8 and S5).

Secondly, we assessed the entire sorting system, i.e. per-protein mistakes reduced per-residue performance (Fig. 3C). Overall, DNA-, RNA-binding reached similar performance; protein-binding was slightly below (Fig. 3C, Table S9). All per-residue prediction methods performed better on non-binding than on binding residues, e.g. reflected by very high levels of the overall two-state per-residue accuracy Q_2 (Eq. (3)) which was dominated by non-binding (Table 1). The test-set results were Q_2 68–70%, 80–82%, and 79–81% for protein, DNA, RNA, respectively (ranges encapsulated ± one standard error rounded to closest integer; details about error estimates are provided in Table S9). With respect to DNA/RNA confusion, 24% of the DNA binding residues were mis-predicted as RNA binding residues (Table S10).

The detailed inspection of particular examples for typical predictions (Fig. 4) suggested that ProNA2020 identified some core of a binding residue (yellow in Fig. 4). This was impressive because the method "sees" only sequence, i.e. has no notion of "binding residue", instead it only predicts "binding residues".

Predictions strength measured by reliability index (RI) correlated with performance

The confidence of each prediction was measured through a reliability index (RI) that scaled from −100 (high confidence for non-binding) to 100 (high confidence for binding). Technically, RI reflected the strength of a prediction. For homology-based

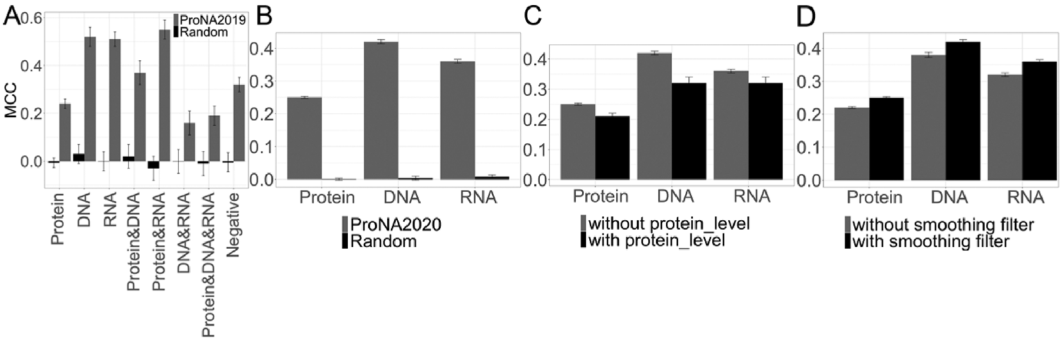

Fig. 3. Test set performance of ProNA2020. All plots show performance for the test set used to assess our new system. The first two panels give the MCC (Eq. (2)) for the per-protein (panel **A**) and per-residue predictions (panel **B**). Our new method, ProNA2020, improved over random (black vs. gray bars) by many standard deviations (±σ shown at each bar). The second two panels both give per-residue performance. Panel **C** compares values with or without errors of the protein sorting system: dark bars: with sorting (i.e. with system errors); gray without sorting (i.e. without system errors). The dark bars provide estimates for predicting binding residues without any prior knowledge; the gray bars estimate performance for users who know that their protein was a binding protein and want to find the residues involved in binding. Panel **D** compares performance between the raw ML solution (gray bars) and the smoothing filter (dark bars) that improved for all classes.

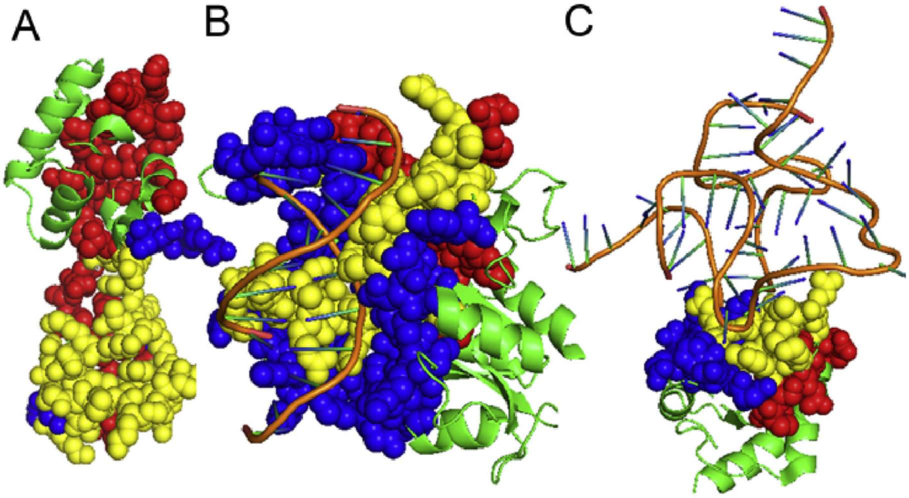

Fig. 4. Representative per-residue predictions. We picked three proteins of known 3D structure to visualize correct and incorrect predictions of binding residues for protein, DNA, and RNA. Coordinates were taken from the PDB [5]. Although each prediction was an average case for its task (complete distribution of predictions in Fig. S6), all three happened to be examples of relatively small "chains" (i.e. protein domain-like regions) that almost entirely bind. Yellow marks correctly predicted residues, blue residues observed in the binding but not predicted (under-predicted false negatives) and magenta residues predicted but not observed (over-predicted false positives). Panel **A** shows the protein binding prediction (6HA7 [57], Q2 = 71%), panel **B** gives a DNA binding prediction (5DWA [58], Q2(this protein) = 78%), and panel **C** samples an RNA binding prediction (5XTM [59], Q2(this protein) = 76%). Note that none of the 3D information was used for the prediction.

inference, the RIs were normalized values for percentage pairwise sequence identities read of the PSI-BLAST alignments (Fig. S4). For the per-protein machine learning predictions, the RIs were taken directly from the ML method output (Method). For the per-residue level, the RIs were taken from the smoothened values (Methods). The binding prediction, higher RIs corresponded to more precise (high PRE, Eq. (2)) but fewer (lower REC, Eq. (2)) predictions (Fig. 5). For instance, for the per-protein sorting, the subset of predictions stronger than 0 (RI $\geq$ 0) reached levels of >60% precision for DNA and RNA (Fig. 5A: full blue and red lines at x = 0). This level was reached for about 70% of all predictions (Fig. 5A: dashed blue and red lines at x = 0). Prediction strength correlated also with performance for the per-residue predictions of binding proteins, e.g. for RI > 0 about 50% of all protein–protein binding residues were correctly predicted (Fig. 5B: full green line), and these constituted over 40% of all the PP-binding predictions (Fig. 5B: dashed green line). For the prediction of non-binding, reversely, lower RIs implied better predictions (Fig. S5).

ProNA2020 performed best in independent comparison

To compare our new method, ProNA2020, with others, we added another independent test set without significant sequence similarity (HVAL<0) to sets used for development. For the per-protein sorting (protein sorting mode, Table 4), ProNA2020 reached the highest F1 score and MCC in protein-binding, RNA-binding, and DNA-binding prediction (Fig. 6, Table S11). Values for precision and recall never are directly comparable because some methods find different balance points, i.e. perform very well on one of the two at the price of performing poorly on another. For instance, hybridNAP reached a recall of 100% on DNA binding and RNA binding at the cost of levels of precision below 42% for DNA and below 22% for RNA. On the other extreme end, SPOT-RNA reached high precision for RNA and DisoRDPbind for protein–protein, but both achieved this at rather low recall (DisoRDPbind 41% for protein–protein, SPOT-RNA 33% for RNA). DisoRDPbind even achieved a second highest MCC in protein binding prediction by the high precision (MCC: 0.21, Fig, 6), because most other methods predicted all proteins as protein binding (NPV = 0 Table S11). Overall, for per-protein prediction, ProNA2020 numerically outperformed all state-of-the-art sequence-based binding protein prediction methods tested (in terms of F1 and MCC; in terms of Q2 for RNA binding, SPOT-RNA and TriPepSVM did better due to under-prediction, Table S11).

Methods developed to predict which residues bind e.g. DNA (per-residue level) could be employed to predict which proteins bind DNA (per-protein level). Our results highlighted the problems originating from such an approach: for all prediction tasks, all per-residue methods clearly over-predicted binding on the per-protein level. This led to very high levels of *Recall*

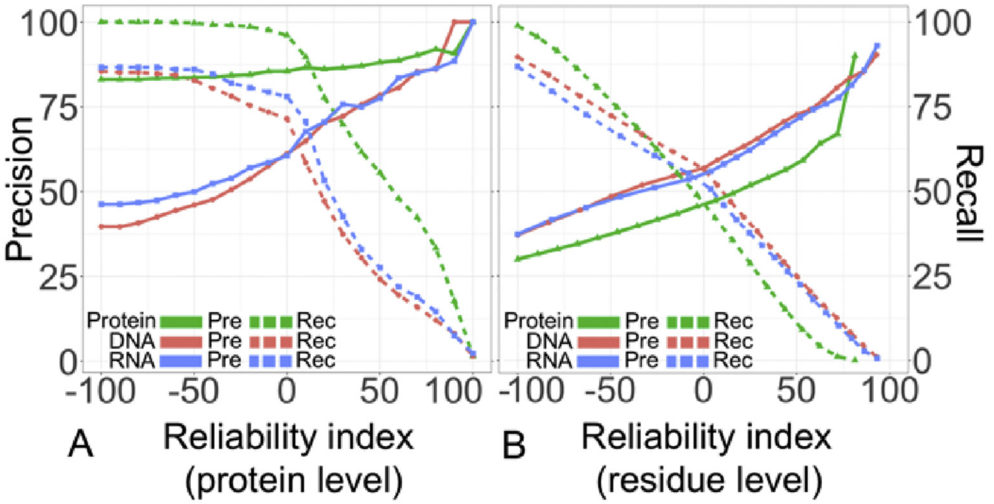

Fig. 5. Reliability index (RI) to focus on best predictions. All machine learning solutions reflect the strength of a prediction even for binary classifications (binding/not). These graphs relate prediction strength to performance. The x-axes give prediction strength as the reliability index (from −100: very non-binding to 100: very binding). The y-axes reflect the percentage precision (full lines, Eq. (2)) and recall (dashed lines, Eq. (2)) for proteins binding to DNA (red), RNA (blue), and other proteins (green). The left panel (A) shows the per-protein methods and the right one (B) the per-residue predictions. For all models, precision is proportional to prediction strengths, i.e. predictions with higher RI are, on average, better. All plots are cumulative, e.g. answering the question: if you looked at all per-residue predictions for DNA (panel B red full line) or RNA (panel B blue full line) with RI > 50 about 75% of all residues you looked at are expected to be correct predictions. Above that threshold, the methods have found slightly over 12.5% of all residues observed to bind DNA (B: dashed red) and RNA (B: dashed blue).

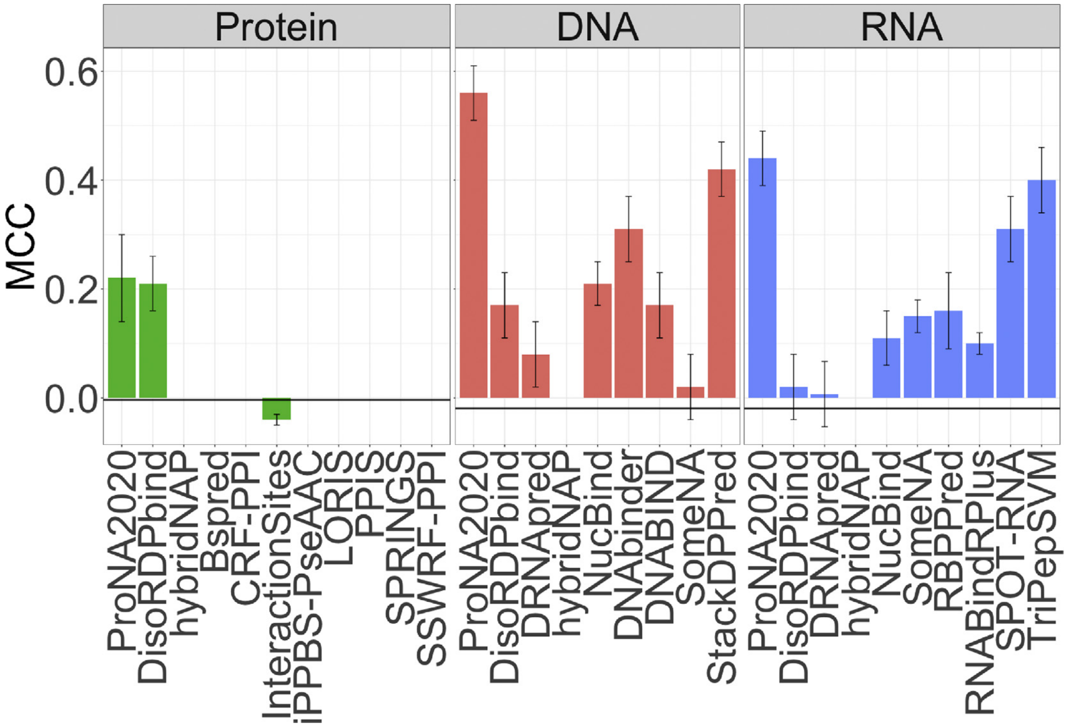

Fig. 6. Per-protein prediction of ProNA2020 in comparison for independent data set. All values are based on three new independent data sets (protein, DNA, and RNA, Table 1) without significant level of sequence similarity to those proteins used for development of all methods. The y-axis gives the MCC (Eq. (2)). Error bars define ±one standard error. All numbers were compiled on exactly the same data set. The horizontal black lines mark random predictions. Note that most data sets were imbalanced, most extreme that for protein–protein binding, as a result all but two methods (DisoRDPbind and ProNA2020) reached the same MCC (Table S11) by simply always predicting protein–protein binding, i.e. by never correctly rejecting any protein. Consequently, the MCC (Eq. (2)) was exactly 0 for all methods (Table S11) other than DisoRDPbind (MCC = 0.21 ± 0.05, Table S11) and ProNA2020 (MCC = 0.22 ± 0.08, Table S11).

at low levels of *Precision* (Table S11) and relatively low F1 scores. This problem was less severe for the identification of proteins that bind other proteins: all methods reached relatively high levels for the independent test set which contained few non-binding proteins, i.e. over-prediction of binding was rewarded, in the most extreme: always predicting binding resulted in F1 = 89%, Q2 = 80% (Precision = 80%, Recall = 100%). Consequently, the negative predictive value (NPV, Eq. (3)) for those methods might be as low as 0% (on a scale of 0–100, Table S11); the MCCs were also all 0 (Fig. 6, Table S11).

Comparing the per-residue level performance, we had to, again, distinguish the two different scenarios. First, users do not know whether or not their query Q binds (residue unknown binding mode, Table 4). Second, they do know that it binds and want to find out where it binds (residue known binding mode, Table 4). For the first scenario (unknown binding mode), no method reached higher F1 or MCC (Table 5 and Table S11, F1: unknown mode) for any task than ProNA2020. For per-residue RNA binding predictions, RNABindRPlus reached a highest MCC together with ProNA2020 (MCC = 0.40), but a slightly lower F1 than ProNA2020 ($F1_{ProNA2020} = 46$ vs. $F1_{RNABindRPlus} = 45$).

Overall, our new method, ProNA2020, appeared to be the best among all state-of-the-art per-residue prediction methods we tested with these new independent data sets. ProNA2020 clearly significantly outperformed other multi-task predictions: DRNApred, NucBind, hybridNAP, and DisoRDPbind (Table 5).

For the second scenario (known binding mode, Table 4), we e.g. only used RNA binding proteins for the per-residue RNA-binding comparison (Table 5 rightmost column, Table S13). ProNA2020 reached the highest F1 score and MCC in the DNA and protein binding per-residue prediction. The higher values were statistically significant (difference more than two standard errors, i.e. p < 0.1; Table 5). For RNA binding, ProNA2020 numerically reached the top MCC, followed by NucBind and RNABindRPlus; however, those two were within a single standard error of the top value, i.e. the differences were statistically not significant (Table 5). Statistically significantly lower was rank four with the other multi-task methods, namely hybridNAP with F1 = 34%, albeit at an MCC of 0.08 (Table 5). For protein binding, ProNA2020 came consistently on top highest F1 and MCC (Table S13). Performance was almost same between overall independent test

Table 5. Overall per-residue performance for independent test set[a].

Method	Binding	Unknown binding mode		Known binding mode	
		F1	MCC	F1	MCC
DisoRDPbind [20]^3	*DNA*	19 ± 3	0.09 ± 0.02	19 ± 3	0.04 ± 0.02
DRNApred [17]^2		28 ± 3	0.13 ± 0.03	30 ± 3	0.10 ± 0.03
hybridNAP [19]^3		35 ± 2	0.12 ± 0.02	40 ± 1	0.08 ± 0.02
NucBind [18]^2		35 ± 5	0.16 ± 0.07	52 ± 2	0.47 ± 0.02*
SomeNA [12]^3		44 ± 2	0.31 ± 0.03	45 ± 2	0.27±±0.04
ProNA2020^3		**60** ± **2**	**0.49** ± **0.02**	**66** ± **1**	**0.50** ± **0.02**
DisoRDPbind [20]^3	*RNA*	15 ± 4	0.05 ± 0.03	20 ± 4	0.04 ± 0.03
DRNApred [17]^2		21 ± 5	0.08 ± 0.06	26 ± 5	0.07 ± 0.04
hybridNAP [19]^3		26 ± 3	0.11 ± 0.02	34 ± 2	0.08 ± 0.03
NucBind [18]^2		20 ± 6	0.03 ± 0.06	43 ± 5*	**0.37** ± **0.05***
RNABindRPlus [42]		45 ± 4*	0.40 ± 0.04*	**50** ± **3***	0.36 ± 0.03*
SomeNA [12]^2		23 ± 2	0.19 ± 0.04	25 ± 3	0.17 ± 0.06
ProNA2020^3		**46** ± **3**	**0.40** ± **0.03**	**50** ± **2**	**0.37** ± **0.03**
DisoRDPbind [20]^3	*Protein*	5 ± 2	−0.03 ± 0.03	5 ± 2	−0.001 ± 0.008
hybridNAP [19]^3		37 ± 2*	0.14 ± 0.02	39 ± 2	0.11 ± 0.02
BSpred [43]		18 ± 2	−0.04 ± 0.02	20 ± 1	−0.036 ± 0.009
CRF-PPI [60]		31 ± 2	0.02 ± 0.01	38 ± 2	0.03 ± 0.01
InteractionSites [36]		14 ± 1	0.05 ± 0.02	15 ± 1	0.05 ± 0.02
iPPBS-PseAAC [44]		20 ± 1	0.04 ± 0.02	22 ± 1	0.027 ± 0.008
LORIS [45]		31 ± 2	0.001 ± 0.007	36 ± 1	0.005 ± 0.008
PPIS [46]		32 ± 2	0.01 ± 0.01	38 ± 2	0.02 ± 0.01
SPRINGS [47]		32 ± 2	0.004 ± 0.007	35 ± 2	−0.01 ± 0.008
SSWRF-PPI [61]		33 ± 2	0.02 ± 0.01	38 ± 2	0.02 ± 0.01
ProNA2020^3		**42** ± **3**	**0.28** ± **0.03**	**47** ± **3**	**0.28** ± **0.03**

[a] **Methods**: superscript numbers give number of tasks for methods that address more than one (maximum is three: DNA, RNA, protein). **Mode-unknown**: for a query protein Q it is **not** known whether it binds DNA/RNA/Protein, instead, this binding has to also be predicted. Methods incorrectly predicting that Q binds DNA will likely mis-predict more residues than those correctly rejecting such a binding mode. Thus, values on right are mostly higher than on left. **Mode-known:** for a query protein Q it is known that it binds DNA/RNA/protein. For instance, when assessing methods for the DNA per-residue prediction, only DNA-binding proteins are presented. **Percentages** for F1 and MCC (Eq. (2)). **BOLD values and * marks**: the numerically top method in each mode is bolded; methods within two standard errors of the numerical top (p-value of difference >0.1).

set and PISA reduced independent test set (biology interface only) (Table S14).

Predictions different for prokaryotes and eukaryotes and similar for unknown data

Separately analyzing the performance for prokaryotic and eukaryotic proteins, we first observed that our training data had more residues annotated as binding RNA in prokaryotes than in eukaryotes (5351 vs. 2308, Table S16); the percentage of RNA-binding residues was also almost twice as high in prokaryotes than in eukaryotes (38% vs. 20%, Table S16); the corresponding percentages were slightly higher in prokaryotes than in eukaryotes for protein-binding (31% vs. 26%, Table S16) and this ratio was inversed for DNA-binding (24% vs. 29%, Table S16). Protein- and RNA-binding residues were predicted substantially better for prokaryotes than for eukaryotes (F1(protein) = 48 ± 0.4 vs. 45 ± 0.4; F1(RNA) = 63 ± 0.2 vs. 49 ± 0.3; Table S15). In contrast, DNA-binding residues were predicted better in eukaryotes (F1(DNA) = 54 ± 0.9 vs. 60 ± 0.8; Table S15). The differences in the amount of binding data used for training correlated but did not explain the differences in performance: protein: observed ratio binding residue (prokaryote/eukaryote) = 1.2 vs. performance (F1) of 1.05; DNA: observed ratio: 0.8, performance 0.9; RNA: observed ratio 1.9, performance 1.3.

Often experimental data sets are biased and machine learning methods inherit the training bias. For instance, all methods predicting the effects of single amino acid variants (SAVs) upon protein function perform very similar for the tiny data sets with experimental annotations, although they perform very differently for proteins without annotations [48]. The independent test sets helped to assess whether or not methods behave the same way for annotated proteins used for development and those not used. Obviously, we cannot "assess" performance for proteins without annotations. However, what we can do is to at least analyze whether the score distributions from a prediction method look similar for proteins of known and unknown function. Toward this end, we applied ProNA2020 to all human proteins and found the distribution of prediction scores to resemble that for the data sets with experimental annotations (Fig. S7).

Discussion

New system works overall better than previous tools

The major objective of this work was the combination of several prediction tasks into one comprehensive prediction system for the prediction of protein–protein, protein–DNA, and protein–RNA binding. The system included the per-protein level to automatically handle predictions for entirely sequenced organisms or metagenomes for which many proteins remained without annotations for these binding modes. The system also combined homology-based inference and machine learning to help users to the best possible prediction for each case. Many of these ideas had been realized before, e.g. the multi-task predictions (for nucleotides: SomeNA [12], DRNApred [17], and NucBind [18]; for nucleotides and proteins: DisoRDPbind [20] and hybridNAP [19]), or per-protein and per-residue level predictions (SomeNA [12]), or the combination of homology-based and machine learning (DisoRDPbind [20]). However, no system had really simultaneously addressed all aspects.

All data sets were too small for out-of-the-box Deep Learning. *Word2vec,* used so successfully by Google [33] and others, including for proteins [35,49] and in *ProtVec* [21], did provide interesting new angles (Fig. 1: blue numbers from *ProtVec*). However, profile-kernel SVMs tailored to protein prediction [12,27,34] performed better overall (Fig. 1: red mostly higher than blue numbers). Similar trends have been observed for other applications in biology [27,32,50–53]. The profile-kernel SVM mines evolutionary information as contained in multiple sequence alignments of protein families, while *ProtVec* aspires at understanding the protein sequence in a different way through NLP. It seems that the machine learning model underlying ProtVec might be too simplistic to achieve this objective. Less simplistic models reach further [54,55]. One problem for profile-kernel SVMs are un-informative (lack of diversity) and incorrect alignments. In such cases, ProtVec can perform better.

The *ProtVec*-like solution performed particularly well for the top-level protein–protein and protein–NA (nucleic acid) sorting (Fig. 1). For these, it outperformed or was *on par* with the profile-kernel SVM (Fig. 1: middle top and left top circle). Conversely, the profile-kernel SVMs clearly performed better for DNA and RNA (Fig. 1: middle circles on right and in center). One common trend was that the larger the data set, the relatively better the *ProtVec*. The finding that the best combination used whichever prediction had the highest score (reliability) suggested that methods had learned independent aspects.

One task often implicitly left to the user is the combination of homology-based inference with machine learning. Building such a combination into a system can improve and simplify predictions [32]. For ProNA2020, performance also improved through in-built combination of machine learning with homology-based inference (Fig. S3). For example, protein-binding protein Q9Y3Y4 cannot be predicted by machine learning, while Q9Y3Y4 hits another protein-binding protein Q9T0K5 through homology-based inference.

The non-redundant independent data set was composed of proteins for which experimental data became available after the proteins used for development (cross-validation). Thus, this set was completely "novel" with respect to independently testing our method. However, several of the other methods compared had access in their development to some (older methods) or most (newer methods) of those proteins, i.e. our independent comparison was conservative in that it likely under-estimated the performance of our methods with respect to that of others. Nevertheless, in this test, no other method statistically significantly outperformed our method and no method combined as many crucially relevant components into a system as ours. Some performance measures cannot be directly compared between methods, e.g. precision and recall: each method finds a different balance. Is method M1 with Precision = 60% and Recall = 30% better than M2 with P = 40%, R = 50%? The only way to answer is through composite scores such as the F1 or MCC. When scanning such composite scores, our new method ProNA2020 reached numerically the highest value for all three per-protein predictions (Table S11, Fig. 6) and for all per-residue assessments (Table 5).

Another important feature of our prediction system that is not assessed through the independent test set is the integration of homology-based inference. By design, the independent test set could not be subjected to homology-based inference, i.e. the method comparison was confined to assessing the machine learning part of ProNA2020. Other methods use homology-based inference (e.g. SBI). In fact, for some or all of the proteins in the independent data set, those methods might have used SBI instead of *de novo* prediction.

Overall, we accomplished our goals: we developed the most comprehensive and most automated system for the prediction of binding of proteins to DNA, RNA, and other proteins. The only limitation of the system are specific predictions: it cannot predict which proteins, DNA, or RNA in particular will bind, only that they will bind and where in the protein that will happen. In absence of knowing 3D structure, the system can also not identify entire binding residues: although when mapping it onto 3D structures (Fig. 4), we observed that parts of binding residues non-consecutive in sequence and close in space had been predicted; however, without the knowledge of 3D structure, this information would not have been available. Thus, the prediction of many non-consecutive protein binding residues might indicate two separate binding pockets, or one very large one. The comprehensive system, ProNA2020, consists of parts, none of which appeared worse than any state-of-the-art prediction method, and while the system will be available to users as a whole, the separate components are also available for expert users through github.

Estimates for sustained performance challenging

When assessing machine learning, proper cross-validation is essential. This includes to have non-redundant data sets and to separate all hyper-parameter optimization and model choice (based on the cross-training set, Fig. S1) from the performance estimates for the final method, for which we used two test sets—the first from our original data set (Fig. S1) and the other independent test set, which most likely had not been used for the development of other methods and clearly not used by us (Methods). We applied the final test sets only to the system that was found best using the cross-training set. This implied that some of the results shown had to be taken from this "development phase" (Fig. 1, Fig. S3), while others were taken from the test set (Fig. 3) or the independent test set (Fig. 6, Table 5). Only these results reflected the final performance estimates for the method. Values for cross-training and testing results might differ more than the estimates of standard errors suggest; this is just an aspect of development. In contrast, if values differed between test and independent test sets, this would suggest some mistake in performance estimates. Indeed, all differences (F1) between the independent and the cross-validation test set remained within less than a single standard error (Table 5, Table S11). Thus, these differences did not challenge the technical correctness of our estimates. Consistent performance of ProNA2020 in cross-validation and the independent test sets suggested that there was rather limited bias from the development set, in particular, in comparison to other methods, some of which tended to perform below the levels published when faced with new proteins between independent test set and publication (Table 5: rightmost two column, Table S13).

Many of our performance comparisons were complicated by the small sets of proteins with experimental annotations that are neither sequence similar to any protein used by any of the methods compared, nor sequence similar to each other. This double constraint has complicated comparisons in many fields of protein prediction, in particular when high-resolution data continues to be impossible for high-throughput experiments. When each novel structure continues to cost over $100,000 [56], data sets with "only" 108 novel protein binding proteins (independent test set, Table 3) carry very high value. Some methods (alphabetically: NucBind [18] and RNABindRPlus [42]) reached a similar value on the independent data set as published. Others remained below the expectations. For one of those, namely for DisoRDPbind [20], the difference was easily explained by that it only focused on the binding residues on the disorder region. Unfortunately, we could not analyze this separately, because for none of the proteins in our independent data set did we find experimental annotations about disorder.

Another particular problem often arising from proper cross-validation is that some alternative way of solving a problem might turn out to be best according to the cross-training set (Fig. 1, e.g. numbers in blue vs. those in red), but not best for the test or the independent test set. We encountered this for the final solution for the protein sorting system: whichever prediction method (profile-kernel SVM or ProtVec Local) had the highest score at each node of the per-protein sorting (Fig. 1) was best for the cross-training but was not best for the independent test set. Proper procedure, in cases such as this, is to trust the procedure and stick with the cross-training results, at the expense of reducing the values in the direct face-to-face comparison to other methods.

Conclusion

Each component of ProNA2020 essentially outperformed the state-of-the-art methods in per-protein sorting (Table S11, Fig. 6). With respect to most criteria, ProNA2020 also outperformed most per-residue prediction methods. When it did not outperform, it was *on par*, or at least not worse by a statistically significant margin (Table 5, Tables S12 and S13). Our method ProNA2020 is available through *github* (below), so that users could combine different components of our system with their solutions. One important novelty is the combination of per-protein sorting and per-residue prediction. We did not use existing annotations, such as Pfam domains, or Swiss-Prot annotations explicitly as input. Therefore, our system is available to be applied to high-throughput analyses, such as comparisons on the level of entire proteomes between organisms. Toward that end, ProNA2020 is available through https://github.com/Rostlab/ProNA2020.git and PredictProtein (http://www.predictprotein.org).

Acknowledgement

We thank Tim Karl for technical and Inga Weise (both TUM) for administrative assistance. We also are happy to very much thank the anonymous reviewers who pointed out many shortcomings in our initial submission and thereby crucially contributed to improving this work. Financial support was obtained from the program of China Scholarship Council. This work was supported by a grant from the Alexander von Humboldt foundation through the German Ministry for Research and Education (BMBF: Bundesministerium fuer Bildung und Forschung), as well as by the Bavarian Ministry for Education. Particular thanks to all who make databases available and all those who contribute their experimental data to such public resources.

Conflict of Interest

None declared.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jmb.2020.02.026.

Received 28 November 2019;
Received in revised form 17 February 2020;
Accepted 23 February 2020
Available online 04 March 2020

Keywords:
binding protein prediction;
binding residue prediction;
profile kernel SVM;
ProtVec;
machine learning

Abbreviations used:
3D structure, three-dimensional coordinates of protein structure; **AUC**, area under the ROC curve; **DBP**, DNA-binding protein; **FPR**, false positive rate; **PPI**, protein—-protein interaction; **ProtVec**, protein vector; **RBP**, RNA-binding protein; **RI**, reliability index; **SVM**, support vector machine; **TPR**, true positive rate; **NA**, used to describe either DNA or RNA.

References

[1] P. Liu, L. Yang, D. Shi, X. Tang, Prediction of protein-protein interactions related to protein complexes based on protein interaction networks, BioMed Res. Int. 2015 (2015), 259157.

[2] Y. Ofran, V. Mysore, B. Rost, Prediction of DNA-binding residues from sequence, Bioinformatics 23 (2007) i347—i353.

[3] C. Sacca, S. Teso, M. Diligenti, A. Passerini, Improved multi-level protein-protein interaction prediction with semantic-based regularization, BMC Bioinf. 15 (2014) 103.

[4] L. Breuza, S. Poux, A. Estreicher, M.L. Famiglietti, M. Magrane, M. Tognolli, et al., The UniProtKB guide to

the human proteome, Database : Off. J. Bio. Databases Curation 2016 (2016).

[5] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, et al., The protein Data Bank, Nucleic Acids Res. 28 (2000) 235–242.

[6] S. Bienert, A. Waterhouse, T.A. de Beer, G. Tauriello, G. Studer, L. Bordoli, et al., The SWISS-MODEL Repository-new features and functionality, Nucleic Acids Res. 45 (2017) D313–D319.

[7] J. Si, J. Cui, J. Cheng, R. Wu, Computational prediction of RNA-binding proteins and binding sites, Int. J. Mol. Sci. 16 (2015) 26303–26317.

[8] J. Si, R. Zhao, R. Wu, An overview of the prediction of protein DNA-binding sites, Int. J. Mol. Sci. 16 (2015) 5194–5215.

[9] A.C. Anderson, The process of structure-based drug design, Chem. Biol. 10 (2003) 787–797.

[10] J.L. Ludington, Protein binding site analysis for drug discovery using a computational fragment-based method, Methods Mol. Biol. 1289 (2015) 145–154.

[11] A. Szilagyi, J. Skolnick, Efficient prediction of nucleic acid binding function from low-resolution protein structures, J. Mol. Biol. 358 (2006) 922–933.

[12] P. Hönigschmid, Improvement of DNA- and RNA- Protein Binding Prediction, Technical University Munich, Munich, 2012.

[13] A. Mishra, P. Pokhrel, M.T. Hoque, StackDPPred: a stacking based prediction of DNA-binding protein from sequence, Bioinformatics 35 (2019) 433–441.

[14] X. Zhang, S. Liu, RBPPred: predicting RNA-binding proteins from sequence using SVM, Bioinformatics 33 (2017) 854–862.

[15] Y. Yang, H. Zhao, J. Wang, Y. Zhou, SPOT-Seq-RNA: predicting protein-RNA complex structure and RNA-binding function by fold recognition and binding affinity prediction, Methods Mol. Biol. 1137 (2014) 119–130.

[16] A. Bressin, R. Schulte-Sasse, D. Figini, E.C. Urdaneta, B.M. Beckmann, A. Marsico, TriPepSVM: de novo prediction of RNA-binding proteins based on short amino acid motifs, Nucleic Acids Res. 47 (2019) 4406–4417.

[17] J. Yan, L. Kurgan, DRNApred, fast sequence-based method that accurately predicts and discriminates DNA- and RNA-binding residues, Nucleic Acids Res. 45 (2017) e84.

[18] H. Su, M. Liu, S. Sun, Z. Peng, J. Yang, Improving the prediction of protein-nucleic acids binding residues via multiple sequence profiles and the consensus of complementary methods, Bioinformatics 35 (2019) 930–936.

[19] J. Zhang, Z. Ma, L. Kurgan, Comprehensive review and empirical analysis of hallmarks of DNA-, RNA- and protein-binding residues in protein chains, Briefings Bioinf. 20 (4) (2019) 1250–1268.

[20] Z. Peng, L. Kurgan, High-throughput prediction of RNA, DNA and protein binding regions mediated by intrinsic disorder, Nucleic Acids Res. 43 (2015) e121.

[21] E. Asgari, M.R. Mofrad, Continuous distributed representation of biological sequences for Deep proteomics and genomics, PloS One 10 (2015), e0141287.

[22] S. Mika, B. Rost, UniqueProt: creating representative protein sequence sets, Nucleic Acids Res. 31 (2003) 3789–3791.

[23] B. Rost, Enzyme function less conserved than anticipated, J. Mol. Biol. 318 (2002) 595–608.

[24] T. Norambuena, F. Melo, The protein-DNA interface database, BMC Bioinf. 11 (2010) 262.

[25] B.A. Lewis, R.R. Walia, M. Terribilini, J. Ferguson, C. Zheng, V. Honavar, et al., PRIDB: a Protein-RNA interface database, Nucleic Acids Res. 39 (2011) D277–D282.

[26] S. Velankar, J.M. Dana, J. Jacobsen, G. van Ginkel, P.J. Gane, J. Luo, et al., SIFTS: structure integration with function, taxonomy and sequences resource, Nucleic Acids Res. 41 (2013) D483–D489.

[27] T. Hamp, B. Rost, Evolutionary profiles improve protein-protein interaction prediction from sequence, Bioinformatics 31 (2015) 1945–1950.

[28] Y. Ofran, B. Rost, Analysing six types of protein-protein interfaces, J. Mol. Biol. 325 (2003) 377–387.

[29] C. Gene Ontology, J.A. Blake, M. Dolan, H. Drabkin, D.P. Hill, N. Li, et al., Gene Ontology annotations and resources, Nucleic Acids Res. 41 (2013) D530–D535.

[30] G. Yachdav, E. Kloppmann, L. Kajan, M. Hecht, T. Goldberg, T. Hamp, et al., PredictProtein–an open resource for online prediction of protein structural and functional features, Nucleic Acids Res. 42 (2014) W337–W343.

[31] E. Krissinel, K. Henrick, Inference of macromolecular assemblies from crystalline state, J. Mol. Biol. 372 (2007) 774–797.

[32] T. Goldberg, M. Hecht, T. Hamp, T. Karl, G. Yachdav, N. Ahmed, et al., LocTree3 prediction of localization, Nucleic Acids Res. 42 (2014) W350–W355.

[33] E. Frank, M. Hall, L. Trigg, G. Holmes, I.H. Witten, Data mining in bioinformatics using Weka, Bioinformatics 20 (2004) 2479–2481.

[34] T. Hamp, T. Goldberg, B. Rost, Accelerating the original profile kernel, PloS One 8 (2013), e68459.

[35] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems2013. p. 3111-3119.

[36] Y. Ofran, B. Rost, ISIS: interaction sites identified from sequence, Bioinformatics 23 (2007) e13–e16.

[37] M. Littmann, K. Selig, L. Cohen, Y. Frank, P. Hönigschmid, E. Kataka, et al., Validity of machine learning in biology and medicine increased through collaborations across fields of expertise, Nat. Mach. Intell. 2 (2020) 18–24.

[38] M. Vihinen, How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis, BMC Genom. 13 (Suppl 4) (2012) S2.

[39] V. Marot-Lassauzaie, M. Bernhofer, B. Rost, Correcting mistakes in predicting distributions, Bioinformatics 34 (2018) 3385–3386.

[40] B. Efron, R. Tibshirani, Statistical data analysis in the computer age, Science 353 (1991) 390–395.

[41] M. Kumar, M.M. Gromiha, G.P. Raghava, Identification of DNA-binding proteins using support vector machines and evolutionary profiles, BMC Bioinf. 8 (2007) 463.

[42] R.R. Walia, L.C. Xue, K. Wilkins, Y. El-Manzalawy, D. Dobbs, V. Honavar, RNABindRPlus: a predictor that combines machine learning and sequence homology-based methods to improve the reliability of predicted RNA-binding residues in proteins, PloS One 9 (2014), e97725.

[43] S. Mukherjee, Y. Zhang, Protein-protein complex structure predictions by multimeric threading and template recombination, Structure 19 (2011) 955–966.

[44] J. Jia, Z. Liu, X. Xiao, B. Liu, K.C. Chou, Identification of protein-protein binding sites by incorporating the physicochemical properties and stationary wavelet transforms into pseudo amino acid composition, J. Biomol. Struct. Dyn. 34 (2016) 1946–1961.

[45] K. Dhole, G. Singh, P.P. Pai, S. Mondal, Sequence-based prediction of protein-protein interaction sites with L1-logreg classifier, J. Theor. Biol. 348 (2014) 47–54.

[46] G.H. Liu, H.B. Shen, D.J. Yu, Prediction of protein-protein interaction sites with machine-learning-based data-cleaning and post-filtering procedures, J. Membr. Biol. 249 (2016) 141–153.

[47] K.D. Gurdeep Singh, Priyadarshini P. Pai, Sukanta Mondal, SPRINGS: Prediction of Protein-Protein Interaction Sites Using Artificial Neural Networks PeerJ PrePrints, 2014.

[48] J. Reeb, M. Hecht, Y. Mahlich, Y. Bromberg, B. Rost, Predicted molecular effects of sequence variants link to system level of disease, PLoS Comput. Biol. 12 (2016), e1005047, https://doi.org/10.1371/journal.pcbi.

[49] J.M. Cejuela, A. Bojchevski, C. Uhlig, R. Bekmukhametov, S. Kumar Karn, S. Mahmuti, et al., nala: text mining natural language mutation mentions, Bioinformatics 33 (2017) 1852–1858.

[50] R. Kuang, C.S. Leslie, A.S. Yang, Protein backbone angle prediction with machine learning approaches, Bioinformatics 20 (2004) 1612–1621.

[51] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, et al., Profile-based string kernels for remote homology detection and motif extraction, J. Bioinf. Comput. Biol. 3 (2005) 527–550.

[52] W.S. Noble, R. Kuang, C. Leslie, J. Weston, Identifying remote protein homologs by network propagation, FEBS J. 272 (2005) 5119–5128.

[53] I. Melvin, E. Ie, R. Kuang, J. Weston, W.N. Stafford, C. Leslie, SVM-Fold: a tool for discriminative multi-class protein fold and superfamily recognition, BMC Bioinf. 8 (Suppl 4) (2007) S2.

[54] M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaeev, F. Matthes, et al., Modeling the Language of Life – Deep Learning Protein Sequences, bioRxiv, 2019, https://doi.org/10.1101/614313.

[55] M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes, et al., Modeling aspects of the language of life through transfer-learning protein sequences, BMC Bioinf. 20 (2019) 723.

[56] J. Liu, G.T. Montelione, B. Rost, Novel leverage of structural genomics, Nat. Biotechnol. 25 (2007) 849–851.

[57] Y. Yan, C. Rato, L. Rohland, S. Preissler, D. Ron, MANF antagonizes nucleotide exchange by the endoplasmic reticulum chaperone BiP, Nat. Commun. 10 (2019) 541.

[58] G. Tamulaitiene, V. Jovaisaite, G. Tamulaitis, I. Songailiene, E. Manakova, M. Zaremba, et al., Restriction endonuclease AgeI is a monomer which dimerizes to cleave DNA, Nucleic Acids Res. 45 (2017) 3547–3558.

[59] K. Oshima, X. Gao, S. Hayashi, T. Ueda, T. Nakashima, M. Kimura, Crystal structures of the archaeal RNase P protein Rpp38 in complex with RNA fragments containing a K-turn motif, Acta Crystallogr. F Struct. Biol. Commun. 74 (2018) 57–64.

[60] Z.S. Wei, J.Y. Yang, H.B. Shen, D.J. Yu, A cascade random forests algorithm for predicting protein-protein interaction sites, IEEE Trans. NanoBioscience 14 (2015) 746–760.

[61] K.H. Zhi-Sen Wei, Jing-Yu Yang, Hong-Bin Shen, Dong-Jun Yu, Protein-protein interaction sites prediction by ensembling SVM and sample-weighted random forests, Neurocomputing 193 (2016) 201–212.