iDNA-Prot: Identification of DNA Binding Proteins Using Random Forest with Grey Model

Wei-Zhong Lin¹,², Jian-An Fang¹, Xuan Xiao²,³*, Kuo-Chen Chou³

¹ Information Science and Technology School, Donghua University, Shanghai, China, ² Computer Department, Jing-De-Zhen Ceramic Institute, Jing-De-Zhen, China, ³ Gordon Life Science Institute, San Diego, California, United States of America

Abstract
DNA-binding proteins play crucial roles in various cellular processes. Developing high throughput tools for rapidly and effectively identifying DNA-binding proteins is one of the major challenges in the field of genome annotation. Although many efforts have been made in this regard, further effort is needed to enhance the prediction power. By incorporating the features into the general form of pseudo amino acid composition that were extracted from protein sequences via the “grey model” and by adopting the random forest operation engine, we proposed a new predictor, called iDNA-Prot, for identifying uncharacterized proteins as DNA-binding proteins or non-DNA binding proteins based on their amino acid sequences information alone. The overall success rate by iDNA-Prot was 83.96% that was obtained via jackknife tests on a newly constructed stringent benchmark dataset in which none of the proteins included has ≥25% pairwise sequence identity to any other in a same subset. In addition to achieving high success rate, the computational time for iDNA-Prot is remarkably shorter in comparison with the relevant existing predictors. Hence it is anticipated that iDNA-Prot may become a useful high throughput tool for large-scale analysis of DNA-binding proteins. As a user-friendly web-server, iDNA-Prot is freely accessible to the public at the web-site on http://icpr.jci.edu.cn/bioinfo/iDNA-Prot or http://www.jci-bioinfo.cn/iDNA-Prot. Moreover, for the convenience of the vast majority of experimental scientists, a step-by-step guide is provided on how to use the web-server to get the desired results.

Introduction
DNA-binding proteins play a vitally important role in many biological processes, such as recognition of specific nucleotide sequences, regulation of transcription, and regulation of gene expression. At present, several experimental techniques (such as filter binding assays, genetic analysis, chromatin immunoprecipitation on microarrays, and X-ray crystallography) have been used for identifying DNA-binding proteins. Although these techniques can provide a detailed picture about the binding, they are both time-consuming and expensive [1]. Particularly, the number of newly discovered protein sequences has been increasing extremely fast. For example, in 1986 the Swiss-Prot [2] database contained only 3,993 protein sequence entries, but now the number has jumped to 530,264 according to the release 2011_07 on 28-Jun-2011 by the UniProtKB/Swiss-Prot at http://www.expasy.org/docs/relnotes/relnstat.html, meaning that the number of protein sequence entries now is more than 134 times the number from about 25 years ago. Facing the avalanche of new protein sequences generated in the postgenomic age, it is highly desired to develop automated methods for rapidly and effectively identifying and characterizing DNA-binding proteins based on the protein sequence information alone.

Actually, numerous predictors were developed in this regard. For instance: Shanahan et al. [3] demonstrated how structural features were employed to determine whether a protein of known structure and unknown function was a DNA-binding proteins or not; Ahmad et al. [4] depicted how to distinguish DNA-binding and non DNA-binding proteins with net charge, net dipole moment and quadrupole moment, respectively; Nordhoff et al. [5] introduced identification and characterization of DNA-binding proteins by mass spectrometry, which was regarded as the most sensitive and specific analytical technique available for protein identification [6]. All the aforementioned methods were considerably relied on the results from biochemical experiments. Among the existing methods, those which are purely based on theoretical approaches are of using various classifying engines, such as support vector machine (SVM) [6,7,8,9,10,11,12,13,14,15], artificial neural network (ANN) [16,17,18,19,20,21], random forest [22,23,24], nearest neighbor [25], and boosted decision trees [26].

In addition to using different prediction engines, a remarkable difference among the existing methods is in that different features were extracted to represent protein samples. For instance, Bhardwaj [13] used a 40-D (dimensional) feature vector to formulate a protein sample that contains positive potential surface patches, overall charge of the protein, and overall surface composition. Yu et al. [10] constructed a 132-D feature vector to represent a protein sequence by using the information of hydrophobicity, predicted secondary structure, solvent accessibility...
ity, normalized van der Waals volume, polarity, and polarizability. Bhardwaj and Lu [9] represented the sample of a protein by harnessing the 70 features of the DNA-binding residues, including the residue’s identity, charge, solvent accessibility, average potential, secondary structure, neighboring residues, and location in a cationic patch. Kumar et al. [14] extracted the features from the PSSM (Position-Specific Scoring Matrix) profile obtained from PSI-BLAST [27] to represent the protein sample. Subsequently, a different approach was proposed [22] to encode each protein sequence with 116 features by incorporating various physico-chemical properties of amino acids. Meanwhile, Nanni and Lumini [15] proposed a method to represent a protein sample by combing ontologies and dipeptide composition. Later, the same authors [6] introduced the grouped weight to represent protein samples for predicting DNA-binding proteins. Langlois and Lu [1] represented a protein sample with 472 features, of which 240 were secondary structure features, 231 dipeptide and Lu [1] represented a protein sample with 472 features, of which 240 were secondary structure features, 231 dipeptide composition features, and one for the total charge over its amino acid sequence.

However, the existing predictors have the following shortcomings. (1) The extracting features are very complicated and their dimensions are too large. Particularly, during the prediction process, some of the existing predictors even need the informations of query proteins that were obtained from other experiments. (2) The computational time needed by these predictors is usually very long; for instance, the predictor IDBP [23] would usually take about 30 minutes for predicting one query protein. (3) Most predictors did not provide a web-server for the public usage, while the others claimed they did but are currently not in working condition and hence their practical application value is quite limited.

In view of this, the present study was initiated in an attempt to develop a new and more powerful predictor by addressing the aforementioned three problems.

According to a recent comprehensive review [28], to establish a really useful statistical predictor for a protein system, we need to consider the following procedures: (i) construct or select a valid benchmark dataset to train and test the predictor; (ii) formulate the protein samples with an effective mathematical expression that can truly reflect their intrinsic correlation with the attribute to be predicted; (iii) introduce or develop a powerful algorithm (or engine) to operate the prediction; (iv) properly perform cross-validation tests to objectively evaluate the anticipated accuracy of the predictor; (v) establish a user-friendly web-server for the predictor that is accessible to the public. Below, let us describe how to deal with these steps.

### Materials and Methods

#### 1. Benchmark datasets

DNA-binding protein sequences were collected from the Protein Data Bank (PDB) release 03-May-2011 at http://www.pdb.org/, in which there are 72,838 structures. By searching the keywords of “Protein-DNA complex” and “DNA binding” through the “Advanced Search Interface”, we extracted 3,689 structures from (PDB).

To construct a high quality benchmark dataset with a wider coverage scope and lower homology bias, the data obtained above were screened strictly according to the following criteria. (1) Sequences with less than 50 amino acid (AA) residues were removed because they might just belong to fragments [29]. (2) Sequences with more than 10 consecutive character of “X” were taken away because they contained too many unknown amino acids. (3) To reduce redundancy and homology bias, the PISCES [30,31] was utilized to cutoff those sequences that have ≥25% pairwise sequence identity to any other in the dataset. Finally, we obtained 212 DNA-binding proteins. Similarly, 212 non DNA-binding protein domains were randomly picked from the data bank. Accordingly, the benchmark dataset $S_{\text{Bench}}$ thus obtained consists of 424 protein sequences of which half are DNA-binding protein sequences and the other half non-binding protein sequences. Their accession codes and sequences are given in Information S1.

#### 2. A novel pseudo amino acid composition of grey model

To develop a powerful predictor for a protein system, one of the keys is to formulate the protein samples with an effective mathematical expression that can truly reflect their intrinsic correlation with the attribute to be predicted [28]. To realize this, the concept of pseudo amino acid composition (PseAAC) was proposed [32] to replace the simple amino acid composition (AAC) for representing the sample of a protein. According to Eq. 6 of [28], the general form of PseAAC for a protein $\mathbf{P}$ can be formulated as

$$
\mathbf{P} = [\psi_1, \psi_2, \ldots, \psi_n, \ldots, \psi_\Omega]^T
$$

where $\mathbf{T}$ is a transpose operator, while the subscript $\Omega$ is an integer and its value as well as the components $\psi_1, \psi_2, \ldots$ will depend on how to extract the desired information from the amino acid sequence of $\mathbf{P}$.

In this study, we are to use the grey model parameters to define the $\Omega$ elements in Eq. 1. In 1989, Deng [33] proposed a grey system theory to investigate the uncertainty of a system. According to this theory, if the information of a system investigated is fully known, it is called a “white system”; if completely unknown, a “black system”; if partially known, a “grey system”. The model developed on the basis of such a theory is called “grey model”, which is a kind of nonlinear and dynamic model formulated by a differential equation. The grey model is particularly useful for solving complicated problems that are lack of sufficient information, or need to process uncertain information and reduce random effects of acquired data.

In the grey system theory, an important and generally used model is called GM(1,1). It is quite effective for monotonic series, with good simulating effect and small error, as reflected by the fact that using the GM(1,1) model has remarkably improved the success rates in predicting protein structural classes [34]. However, if the series concerned are not monotonic, the simulating effect of GM(1,1) would not be good and its error might be quite large.

To overcome the above problem, the grey system theory used in the current study is a special grey dynamic model called GM(2,1), which can be used to handle the oscillation series. In GM(2,1) the strategy of minimum squares will be adopted to determine the uncertain parameters, as can be briefly described below.

One of the most commonly used approaches in the grey system theory is the accumulative generation operation (AGO), which can convert a series without any obvious regularity into a strict monotonic increasing series so as to reduce the randomness and enhance the smoothness of the series, and minimize interference from the random information. Let us assume that $X^{(0)} = (x^{(0)}(1), x^{(0)}(2), \ldots, x^{(0)}(n))$ is the original series of real numbers with an irregular distribution, and it is a non-negative original data sequence. Then, $X^{(1)} = (x^{(1)}(1), x^{(1)}(2), \ldots, x^{(1)}(n))$ is viewed as the first-order accumulative generation operation (1-AGO) series for $X^{(0)}$; i.e., the components in $X^{(1)}$ are given by
The GM(2,1) model can be expressed by the following second-order grey differential equation with one variable:

$$x^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i), \quad k = 1, 2, \ldots , n$$

(2)

The GM(2,1) model can be expressed by the following second-order grey differential equation with one variable:

$$x^{(1)}x^{(0)}(k) + a_1 x^{(0)}(k) + a_2 x^{(1)}(k) = b$$

(3)

where:

$$x^{(1)}x^{(0)}(k) = x^{(0)}(k) - x^{(0)}(k - 1), \quad k = 2, 3, \ldots , n$$

(4)

$$z^{(1)}(k) = 0.5x^{(1)}(k) + 0.5x^{(1)}(k - 1), \quad k = 2, 3, \ldots , n$$

(5)

In Eq. 3, the coefficients $-a_1$ and $-a_2$ are the developing coefficients, and $b$ the influence coefficient. Then we have

$$P = [a_1, a_2, b]^{T}$$

(6)

Thus, it follows by the least-squares method that

$$P = (B^{T}B)^{-1}B^{T}U$$

(7)

where

$$B = \begin{bmatrix}
-x^{(0)}(2) & -z^{(1)}(2) & 1 \\
-x^{(0)}(3) & -z^{(1)}(3) & 1 \\
\vdots & \vdots & \vdots \\
-x^{(0)}(n) & -z^{(1)}(n) & 1
\end{bmatrix}$$

(8)

$$U = \begin{bmatrix}
z^{(1)}x^{(0)}(2) \\
z^{(1)}x^{(0)}(3) \\
\vdots \\
z^{(1)}x^{(0)}(n)
\end{bmatrix}$$

(9)

The least-square estimator for the coefficients $-a_1$, $-a_2$ and $b$ should carry some intrinsic information contained in the discrete data sequence $X^{(0)}$ sampled from the system investigated. In view of this, the incorporation of these coefficients into the general form of PseAAC [Eq. 1] will make the formulation of a protein sample better to reflect its intrinsic correlation with the attribute to be predicted. This is the key of the novel approach. The concrete procedures are as follows.

A protein sequence is composed of 20 different types of native amino acids denoted by A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W and Y. Before using the grey dynamic model GM(2,1), we need to represent the protein sequence by a series of real numbers. Thus, according Eq. 1, the protein $P$ can be formulated with a new mode of PseAAC as given by

$$P = [\psi_1, \psi_2, \ldots, \psi_{20}]^{T}$$

(11)

where $\psi_i (i = 1, 2, \ldots, 20)$ are the occurrence frequencies of the 20 different types of amino acids in the protein concerned, while $\psi_{21, 22, 23}$ represent the absolute value of coefficients $a_1, a_2$ and $b$, respectively.

3. Predicting algorithm

The three coefficients obtained in the above section, in addition to the 20 components in the classical amino acid composition [36], can be used to form a new mode of PseAAC, with $\Omega = 20 + 3 = 23$ components. Thus, according Eq. 1, the protein $P$ can be formulated with a new mode of PseAAC as given by

$$f(x) = \frac{1}{1 + e^{-x}}$$

(10)

Through the above function, each of the 20 amino acids can be translated into numerical variable within the region of $(0, 1)$ (Table 1). With the numerical codes thus obtained, we can convert a protein sequence to a series of real numbers. Thus, the three coefficients $[a_1, a_2, b]$ for any protein sequence can be derived with the grey model GM(2,1) by following Eqs. 2–9.

### Table 1. The numerical codes of 20 native amino acids.

<table>
<thead>
<tr>
<th>Amino acid</th>
<th>Factor score [35]</th>
<th>$f(x)$ = $\frac{1}{1 + e^{-x}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-0.733</td>
<td>0.325</td>
</tr>
<tr>
<td>C</td>
<td>-0.862</td>
<td>0.297</td>
</tr>
<tr>
<td>D</td>
<td>-3.656</td>
<td>0.025</td>
</tr>
<tr>
<td>E</td>
<td>1.477</td>
<td>0.814</td>
</tr>
<tr>
<td>F</td>
<td>1.691</td>
<td>0.869</td>
</tr>
<tr>
<td>G</td>
<td>1.330</td>
<td>0.791</td>
</tr>
<tr>
<td>H</td>
<td>-1.673</td>
<td>0.158</td>
</tr>
<tr>
<td>I</td>
<td>2.131</td>
<td>0.849</td>
</tr>
<tr>
<td>K</td>
<td>0.533</td>
<td>0.630</td>
</tr>
<tr>
<td>L</td>
<td>-1.505</td>
<td>0.182</td>
</tr>
<tr>
<td>M</td>
<td>2.219</td>
<td>0.902</td>
</tr>
<tr>
<td>N</td>
<td>1.299</td>
<td>0.720</td>
</tr>
<tr>
<td>P</td>
<td>-1.628</td>
<td>0.164</td>
</tr>
<tr>
<td>Q</td>
<td>-3.005</td>
<td>0.047</td>
</tr>
<tr>
<td>R</td>
<td>1.502</td>
<td>0.818</td>
</tr>
<tr>
<td>S</td>
<td>-4.760</td>
<td>0.008</td>
</tr>
<tr>
<td>T</td>
<td>2.213</td>
<td>0.901</td>
</tr>
<tr>
<td>V</td>
<td>-0.544</td>
<td>0.367</td>
</tr>
<tr>
<td>W</td>
<td>0.672</td>
<td>0.662</td>
</tr>
<tr>
<td>Y</td>
<td>3.097</td>
<td>0.957</td>
</tr>
</tbody>
</table>

The numerical codes of 20 native amino acids.

($\Omega = 20+3=23$)
the same distribution for all trees in the forest. It has been shown that combining multiple trees produced in randomly selected subspaces can significantly improve the prediction accuracy. RF performs a type of cross-validation by using out-of-bag samples. During the training process, each tree is constructed using a different bootstrap sample from the original data. For the detailed description about of the RF algorithm, refer to the papers [41,42,43].

The RF algorithm is available via the link at http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm. Recently, the RF tool for the MATLAB windows is also available at http://code.google.com/p/randomforest-matlab that has two important functions: one is “classRF_train” for training given data and returning the prediction model, and the other is “classRF_predict” for predicting query input with the prediction model. The classifier in this paper was developed based on the RF tool for the MATLAB windows.

The classifier thus established is called iDNA-Prot, which can be used to predict whether a protein can bind with DNA according to its sequence information alone.

For practical applications, a web-server of iDNA-Prot was established at the web-site http://icpr.jci.edu.cn/bioinfo/iDNA-Prot. Moreover, for the convenience of the vast majority of experimental scientists, a step-by-step guide on how to use the web-server is given in Information S3, by which users can easily get their desired results without the need to follow the complicated mathematic equations involved in developing the iDNA-Prot predictor.

Results and Discussion

In statistical prediction, the following three cross-validation methods are often used to examine a predictor for its effectiveness in practical application: independent dataset test, subsampling test, and jackknife test [44]. However, of the three test methods, the jackknife test is deemed the most objective [45]. The reasons are as follows. (1) For the independent dataset test, although all the proteins used to test the predictor are outside the training dataset used to train it so as to exclude the “memory” effect or bias, the way of how to select the independent proteins to test the predictor could be quite arbitrary unless the number of independent proteins is sufficiently large. This kind of arbitrariness might result in completely different conclusions. For instance, a predictor achieving a higher success rate than the other predictor for a given independent testing dataset might fail to keep so when tested by another independent testing dataset [44]. (2) For the subsampling test, the concrete procedure usually used in literatures is the 5-fold, 7-fold or 10-fold cross-validation. The problem with this kind of subsampling test is that the number of possible selections in dividing a benchmark dataset is an astronomical figure even for a very simple dataset, as demonstrated by Eqs.28–30 in [28]. Therefore, in any actual subsampling cross-validation tests, only an extremely small fraction of the possible selections are taken into account. Since different selections will always lead to different results even for a same benchmark dataset and a same predictor, the subsampling test cannot avoid the arbitrariness either. A test method unable to yield a unique outcome cannot be deemed as a good one. (3) In the jackknife test, all the proteins in the benchmark dataset will be singled out one-by-one and tested by the predictor trained by the remaining protein samples. During the process of jackknifing, both the training dataset and testing dataset are actually open, and each protein sample will be in turn moved between the two. The jackknife test can exclude the “memory” effect. Also, the arbitrariness problem as mentioned above for the independent dataset test and subsampling test can be avoided because the outcome obtained by the jackknife cross-validation is always unique for a given benchmark dataset. Accordingly, the jackknife test has been increasingly and widely used by those investigators with strong math background to examine the quality of various predictors (see, e.g.,[46,47,48,49,50,51,23,33,35,36,37]). In view of this, here the jackknife cross-validation was also used to examine the prediction quality of the current predictor.

The results thus obtained with iDNA-Prot on the benchmark dataset $\mathcal{S}^{\text{Bench}}_{\text{trans}}$ of Information S1 are given in Table 2, from which we can see that the overall success rate was 83.96% in identifying proteins as DNA-binding proteins and non-DNA-binding proteins.

Furthermore, as a demonstration to show that the current predictor iDNA-Prot is superior to the existing ones, let us compare iDNA-Prot with DNA-Prot [22]. The reason we chose DNA-Prot for comparison is because among the existing methods for predicting DNA-binding proteins, the reported success rate achieved by DNA-Prot [22] is the highest. The datasets used to train and test DNA-Prot [22] as well as its standalone version can be obtained from http://www3.ntu.edu.sg/home/EPNSugan/index_files/dnaprot.htm.

The training dataset $\mathcal{S}^{\text{tran}}_{\text{test}}$ used for DNA-Prot [22] contains 146 DNA-binding proteins and 250 non-DNA-binding proteins.

The data used to test DNA-Prot [22] contain the following three sets: (i) testing dataset-1, $\mathcal{S}^{\text{test}}_{\text{trans}}$, consisting of 92 DNA-binding proteins and 100 non DNA-binding proteins; (ii) testing dataset-2, $\mathcal{S}^{\text{test}}_{\text{trans}}$, consisting of 823 DNA-binding proteins and 823 non DNA-binding proteins; and (iii) testing dataset-3, $\mathcal{S}^{\text{test}}_{\text{trans}}$, consisting of 88 DNA-binding proteins and 233 non DNA-binding proteins. All these datasets were elaborated in [22].

However, it was found (see Information S2) that there are 10 identical protein sequences between the 146 DNA-binding proteins in the training dataset $\mathcal{S}^{\text{tran}}_{\text{test}}$ and the 92 DNA-binding proteins in the 1st testing dataset $\mathcal{S}^{\text{test}}_{\text{trans}}$; that 19 identical protein sequences between the 146 DNA-binding proteins and the 88 DNA-binding proteins in the 3rd testing dataset $\mathcal{S}^{\text{test}}_{\text{trans}}$; and that 94 identical protein sequences between the

<table>
<thead>
<tr>
<th>Protein type</th>
<th>Number of proteins</th>
<th>Number of correct prediction</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNA-binding protein</td>
<td>212</td>
<td>179</td>
<td>84.43%</td>
</tr>
<tr>
<td>Non DNA-binding protein</td>
<td>212</td>
<td>177</td>
<td>83.49%</td>
</tr>
<tr>
<td>Overall</td>
<td>424</td>
<td>356</td>
<td>83.96%</td>
</tr>
</tbody>
</table>

*The following parameters were used for Random Forest algorithm: the number of tree grown was 560 and the number of predictors sampled for splitting at each node was 5. doi:10.1371/journal.pone.0024756.t002
250 non-DNA-binding proteins in the training dataset $S_{\text{tran}}$ and the 233 non-DNA-binding proteins in the 3rd testing dataset $S_{\text{3test}}$. In other words, the so-called independent datasets used by DNA-Prot [22] were actually not independent and hence would lead to over-estimated success rates.

Accordingly, to perform an objective and fair comparison of iDNA-Prot with DNA-Prot [22], let us construct a real independent dataset by randomly picking some DNA-binding proteins and non DNA-binding proteins from PDB (Protein Data Bank) according to the following criteria. These proteins must not occur in the training dataset of iDNA-Prot nor in the training dataset for DNA-Prot [22], and that none of the proteins included has more than 40% sequence identity to any other in a same subset. By doing so, we obtained a real independent dataset $S_{\text{Ind}}$, in which 122 proteins are DNA-binding proteins and 122 non DNA-binding proteins. The sequences and accession numbers for such 244 independent proteins are given in the Information S4.

Listed in Table 3 are the tested results obtained respectively by DNA-Prot [22] and iDNA-Prot on the independent dataset in the Information S3.

<table>
<thead>
<tr>
<th>Protein type</th>
<th>DNA-Prot [22]</th>
<th>iDNA-Prot</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNA-binding protein</td>
<td>87/122 = 71.31%</td>
<td>109/122 = 89.34%</td>
</tr>
<tr>
<td>Non DNA-binding protein</td>
<td>101/122 = 82.79%</td>
<td>111/122 = 90.98%</td>
</tr>
<tr>
<td>Overall</td>
<td>188/244 = 77.05%</td>
<td>220/244 = 90.16%</td>
</tr>
</tbody>
</table>

doi:10.1371/journal.pone.0024756.t003

Table 4. The predicted results by iDNA-Prot on the 100 DNA-binding hypothetical proteins from http://www.ncbi.nlm.nih.gov/protein/?term=DNA+binding+hypothetical.

<table>
<thead>
<tr>
<th>Gl code</th>
<th>Predicted result</th>
<th>Gl code</th>
<th>Predicted result</th>
<th>Gl code</th>
<th>Predicted result</th>
</tr>
</thead>
<tbody>
<tr>
<td>21960164</td>
<td>DBP</td>
<td>29122980</td>
<td>non DBP</td>
<td>26832636</td>
<td>DBP</td>
</tr>
<tr>
<td>21957418</td>
<td>DBP</td>
<td>21671920</td>
<td>DBP</td>
<td>26832400</td>
<td>DBP</td>
</tr>
<tr>
<td>21961058</td>
<td>DBP</td>
<td>32880245</td>
<td>DBP</td>
<td>21835917</td>
<td>DBP</td>
</tr>
<tr>
<td>21960858</td>
<td>DBP</td>
<td>90578605</td>
<td>DBP</td>
<td>21835539</td>
<td>DBP</td>
</tr>
<tr>
<td>21960204</td>
<td>DBP</td>
<td>90410315</td>
<td>DBP</td>
<td>21843120</td>
<td>DBP</td>
</tr>
<tr>
<td>21958545</td>
<td>DBP</td>
<td>89076244</td>
<td>non DBP</td>
<td>21836833</td>
<td>DBP</td>
</tr>
<tr>
<td>21958841</td>
<td>non DBP</td>
<td>23326729</td>
<td>non DBP</td>
<td>14627522</td>
<td>DBP</td>
</tr>
<tr>
<td>14828174</td>
<td>DBP</td>
<td>90439438</td>
<td>DBP</td>
<td>78363301</td>
<td>DBP</td>
</tr>
<tr>
<td>52629876</td>
<td>DBP</td>
<td>90328556</td>
<td>DBP</td>
<td>30116886</td>
<td>DBP</td>
</tr>
<tr>
<td>21958534</td>
<td>DBP</td>
<td>89048073</td>
<td>non DBP</td>
<td>20673954</td>
<td>DBP</td>
</tr>
<tr>
<td>21958313</td>
<td>DBP</td>
<td>30724697</td>
<td>DBP</td>
<td>88595361</td>
<td>DBP</td>
</tr>
<tr>
<td>21957779</td>
<td>non DBP</td>
<td>30726408</td>
<td>DBP</td>
<td>15769834</td>
<td>DBP</td>
</tr>
<tr>
<td>21958822</td>
<td>DBP</td>
<td>30726252</td>
<td>DBP</td>
<td>68057023</td>
<td>DBP</td>
</tr>
<tr>
<td>21957238</td>
<td>DBP</td>
<td>30725976</td>
<td>DBP</td>
<td>59480370</td>
<td>non DBP</td>
</tr>
<tr>
<td>15522778</td>
<td>DBP</td>
<td>30725598</td>
<td>DBP</td>
<td>52004347</td>
<td>DBP</td>
</tr>
<tr>
<td>21960397</td>
<td>DBP</td>
<td>30725306</td>
<td>DBP</td>
<td>11114707</td>
<td>DBP</td>
</tr>
<tr>
<td>21960196</td>
<td>DBP</td>
<td>30725067</td>
<td>DBP</td>
<td>22984379</td>
<td>DBP</td>
</tr>
<tr>
<td>21960777</td>
<td>DBP</td>
<td>30724845</td>
<td>DBP</td>
<td>22284549</td>
<td>DBP</td>
</tr>
<tr>
<td>21960121</td>
<td>DBP</td>
<td>16882676</td>
<td>DBP</td>
<td>14564202</td>
<td>DBP</td>
</tr>
<tr>
<td>21960008</td>
<td>DBP</td>
<td>30687056</td>
<td>DBP</td>
<td>14563738</td>
<td>non DBP</td>
</tr>
<tr>
<td>21959358</td>
<td>non DBP</td>
<td>30686776</td>
<td>DBP</td>
<td>14563550</td>
<td>DBP</td>
</tr>
<tr>
<td>21959322</td>
<td>DBP</td>
<td>30686615</td>
<td>DBP</td>
<td>90406920</td>
<td>DBP</td>
</tr>
<tr>
<td>21959035</td>
<td>DBP</td>
<td>30686107</td>
<td>DBP</td>
<td>14563638</td>
<td>DBP</td>
</tr>
<tr>
<td>21956991</td>
<td>DBP</td>
<td>30685947</td>
<td>DBP</td>
<td>29434364</td>
<td>DBP</td>
</tr>
<tr>
<td>21957969</td>
<td>DBP</td>
<td>30685727</td>
<td>DBP</td>
<td>23327083</td>
<td>non DBP</td>
</tr>
<tr>
<td>21957386</td>
<td>DBP</td>
<td>30685502</td>
<td>DBP</td>
<td>93211002</td>
<td>DBP</td>
</tr>
<tr>
<td>21956952</td>
<td>DBP</td>
<td>30685211</td>
<td>DBP</td>
<td>14498544</td>
<td>DBP</td>
</tr>
<tr>
<td>21877200</td>
<td>DBP</td>
<td>30977777</td>
<td>DBP</td>
<td>14526726</td>
<td>DBP</td>
</tr>
<tr>
<td>26832542</td>
<td>DBP</td>
<td>18859138</td>
<td>DBP</td>
<td>14527329</td>
<td>DBP</td>
</tr>
<tr>
<td>26832403</td>
<td>DBP</td>
<td>52002457</td>
<td>DBP</td>
<td>22981160</td>
<td>DBP</td>
</tr>
<tr>
<td>33989345</td>
<td>DBP</td>
<td>17093877</td>
<td>DBP</td>
<td>46913396</td>
<td>DBP</td>
</tr>
<tr>
<td>33989345</td>
<td>DBP</td>
<td>30891446</td>
<td>DBP</td>
<td>71342240</td>
<td>DBP</td>
</tr>
<tr>
<td>33875610</td>
<td>DBP</td>
<td>26832638</td>
<td>DBP</td>
<td>26832637</td>
<td>DBP</td>
</tr>
</tbody>
</table>

Table 3. A comparison of the predicted results by DNA-Prot [22] and iDNA-Prot on the independent dataset in the Information S3.

doi:10.1371/journal.pone.0024756.t003
in $\mathcal{S}^{\text{Ind}}$ (Information S4). From the table we can see that the success rates by \textbf{iDNA-Prot} in identifying both DNA-binding and non-DNA-binding proteins are remarkably higher than those by \textbf{DNA-Prot} [22], and that the overall success rate achieved by \textbf{iDNA-Prot} is about 13% higher than that by \textbf{DNA-Prot} [22]. In addition to yielding higher success rates than those by the relevant existing predictors, the computational time needed by \textbf{iDNA-Prot} to complete a prediction is also significantly shorter than any of its counterparts, and hence \textbf{iDNA-Prot} may become a useful high throughput tool for large-scale investigation of DNA-binding proteins. Moreover, as a demonstration to show the efficiency of the current method, the hypothetical proteins that are annotated as DNA-binding proteins were used to test \textbf{iDNA-Prot}. The information about this kind of hypothetical proteins can be obtained at http://www.ncbi.nlm.nih.gov/protein/?term=DNA-binding+hypothetical, from which we randomly picked 100 DNA-binding hypothetical proteins for test. The results predicted by \textbf{iDNA-Prot} on these proteins are given in Table 4, from which we can see the overall success rate is 90%.

Supporting Information

Information S1 The benchmark dataset $\mathcal{S}^{\text{Bench}}$ includes 424 proteins, classified into 212 DNA-binding proteins and 212 non-DNA-binding proteins. Both the accession identifier of PDB (Protein Data Bank) and sequences are given. None of the proteins has more than 25% sequence identity to any other in a same subset. See the text of the paper for further explanation.

Information S2 List of protein codes that occur in both the training and testing datasets for \textbf{DNA-Prot} (Kumar et al., 2009). See the main paper for further explanation.

Information S3 A step-by-step guide on how to use the web-server of \textbf{iDNA-Prot} to get the desired results.

Information S4 The independent dataset $\mathcal{S}^{\text{Ind}}$ includes 244 proteins, classified into 122 DNA-binding proteins and 122 non-DNA-binding proteins. Both the accession identifier of PDB (Protein Data Bank) and sequences are given. None of the proteins has more than 40% sequence identity to any other in a same subset. See the text of the paper for further explanation.

Author Contributions

Conceived and designed the experiments: WZL XX KCC. Performed the experiments: WZL JAF. Analyzed the data: WZL KCC. Contributed reagents/materials/analysis tools: XX. Wrote the paper: WZL KCC.

References