Bioinformatics, 2024, pp. 1–8

doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

PAPER

Transcription factor prediction using protein 3D structures

Fabian Neuhaus,[1,†] Jeanine Liebold,[1,2†] Jan Baumbach[1,3] and Khalique Newaz[1,4,*]

[1]Institute for Computational Systems Biology, Universität Hamburg, Albert-Einstein-Ring 8-10, 22761 Hamburg, Germany, [2]Faculty of Mathematics, Informatics and Natural Sciences, ZBH - Center for Bioinformatics, Universität Hamburg, Albert-Einstein-Ring 8-10, 22761 Hamburg, Germany, [3]Department of Mathematics and Computer Science, University of Southern Denmark, Campusvej 55, 5230 Odense, Denmark and [4]Center for Data and Computing in Natural Sciences, Universität Hamburg, Albert-Einstein-Ring 8-10, 22761 Hamburg, Germany

*Corresponding author, †Equal contribution

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Motivation: Transcription factors (TFs) are DNA-binding proteins that regulate expressions of genes in an organism. Hence, it is important to identify novel TFs. Traditionally, novel TFs have been identified by their sequence similarity to the DNA-binding domains (DBDs) of known TFs. However, this approach can miss to identify a novel TF that is not sequence similar to any of the known DBDs. Hence, computational methods have been developed for the TF prediction task that, instead of relying on known DBDs, use sequence features of proteins to train a machine learning model, in order to capture sequence patterns that distinguish TFs from other proteins. Because 3-dimensional (3D) structure of a protein captures more information than its sequence, using 3D protein structures can more correctly predict novel TFs.
Results: We propose the first deep learning-based TF prediction method (named *StrucTFactor*) based on 3D protein structures. We compare StrucTFactor with a recent state-of-the-art TF prediction method that relies only on protein sequences. We evaluate the considered methods on $\sim 550,000$ proteins across 12 datasets, capturing different aspects of data bias (including sequence redundancy and 3D protein structural quality) that can influence a method's performance. We find that StrucTFactor significantly (p-value < 0.001) outperforms the existing state-of-the-art TF prediction method, improving performance by up to 23% based on Matthews correlation coefficient. Our results show the importance of using 3D protein structures to predict novel TFs. We provide StrucTFactor as a computational pipeline.
Contact: khalique.newaz@uni-hamburg.de

Key words: transcription factors, transcription factor prediction, protein secondary structures

1. Introduction

1.1. Background and motivation

Transcription Factors (TFs) are DNA-binding proteins that influence the transcription regulations of genes (Latchman, 1997). Thus, knowing which proteins are TFs would help to better understand gene regulatory systems (Lambert et al., 2018). Because identification of TFs using wet-lab experiments is costly and time-consuming (Geertz and Maerkl, 2010), computational methods for TF predictions have been proposed.

Traditional computational methods predict a protein as a TF if the protein shows high sequence similarity to any of the DNA-binding domains (DBDs) of known TFs (Kummerfeld and Teichmann, 2006; Wilson et al., 2007). Because such traditional methods would fail to identify a TF that does not show sequence similarity to any of the known DBDs, alternative data-driven methods for TF predictions have been proposed that do not rely on known DBDs (Eichner et al., 2013; Zheng et al., 2008; Kim et al., 2021; Du et al., 2023). Such data-driven methods use features of proteins with known labels (i.e., whether a protein is a TF or not) to train a machine learning model in a supervised manner (i.e., by including known TF vs. non-TF labels of proteins in the training process) and then use features of a protein (not known to be a TF) as input into the trained model to predict whether the protein is a TF or not. Existing data-driven methods can be grouped into two categories based on how they extract features of proteins.

One category of data-driven methods relies on "handcrafted" protein features. An existing method in this category is TFpredict (Eichner et al., 2013). TFpredict aligns the sequence of a protein to known TFs and non-TFs using the basic local alignment search tool (Altschul et al., 1990), and then uses

© The Author 2024. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

the distribution of the corresponding alignment scores (i.e., bit scores) as features of the protein. Given such features of proteins, TFpredict trains a support vector machine to predict novel TFs. TFpredict outperformed previously proposed TF prediction methods (Eichner et al., 2013).

Another category of recently proposed data-driven methods, which is the focus of this study, extract features of proteins using deep learning (in contrast to handcrafted sequence features). The two recent methods in this category are DeepTFactor (Kim et al., 2021) and TFnet (Du et al., 2023). DeepTFactor first represents a protein sequence in a "one-hot" encoding matrix which, for each sequence position of the protein, indicates the type of amino acid. Given such one-hot encoding matrices of proteins with known TF vs. non-TF labels, DeepTFactor trains a convolutional neural network (CNN)-based deep learning model, which can be used to predict whether an input protein is a TF or not. DeepTFactor outperformed two TF prediction methods including TFpredict. While DeepTFactor uses a simple one-hot encoding matrix as the initial feature of a protein to be fed into deep learning, TFnet relies on a more sophisticated position-specific scoring matrix (Altschul et al., 1997) as initial features of a protein that then is fed into a deep learning model based on CNNs and recurrent neural networks (Du et al., 2023).

All of the existing data-driven TF prediction methods rely on sequence features of proteins. However, proteins (including TFs) perform functions in their 3-dimensional (3D) structural forms. Using 3D structural features of proteins has already been shown to outperform sequence features in several other tasks, including protein structural classification (Newaz et al., 2022; Guo et al., 2019; Newaz et al., 2020). Hence, in the light of an exponential increase in the number of predicted 3D protein structures (Jumper et al., 2021; Lin et al., 2023) of near experimental quality (Akdel et al., 2022), using 3D protein structural features of proteins can further improve upon the current state-of-the-art TF prediction methods, which is what we show in this study.

1.2. Our contributions

We propose a novel data-driven TF prediction method called *StrucTFactor*, which works as follows. First, similar to DeepTFactor, StrucTFactor represents a protein in a one-hot encoding matrix. While DeepTFactor's one-hot encoding of the protein indicates the type of amino acid, StrucTFactor's one-hot encoding of the protein indicates the type of amino acid along with the type of secondary structural elements (i.e., α-helix, β-sheet, or coil-turn), present at each sequence position of the protein. Second, given the secondary structure-based initial feature matrices of proteins with known TF vs. non-TF labels, StrucTFactor trains a CNN-based deep learning model, which is then used to predict novel TFs (Figure 1).

We compare StrucTFactor with DeepTFactor because DeepTFactor is the most recent TF prediction method (with publicly available software) that outperformed previously existing TF prediction methods (Kim et al., 2021). Thus, showing that StrucTFactor outperforms DeepTFactor would mean that StrucTFactor outperforms any other previously proposed TF prediction method. While a recently proposed TF prediction method called TFnet outperformed DeepTFactor (Du et al., 2023), the corresponding software is not publicly available, and hence we could not include TFnet in our study.

We use $\sim$550,000 proteins from UniProt (Consortium, 2019) with 3D structures from AlphaFold (Varadi et al., 2021). Given the $\sim$550,000 proteins, we create 12 datasets capturing different aspects of data bias, including sequence redundancy, number of TFs vs. non-TFs (i.e., class imbalance ratios), and 3D protein structural quality, that can influence the training of a machine learning model. Given a dataset, we evaluate the performance of a TF prediction method in a 5-fold cross-validation manner using four performance measures, including Matthews correlation coefficient (MCC).

We find that StrucTFactor significantly (p-value < 0.001) outperforms DeepTFactor, improving performance by up to 23% based on MCC. We use the concept of integrated gradients (Sundararajan et al., 2017) to investigate the parts of a protein that StrucTFactor focuses on, in order to classify the protein as a TF or not. We find that, in contrast to DeepTFactor, StrucTFactor puts more focus on the known DBDs of TFs, potentially explaining its superiority over DeepTFactor. We successfully perform literature validation of the two novel proteins uniquely predicted by StrucTFactor. We provide StrucTFactor as a computational pipeline (see *Availability*).

2. Data and method

Given 570,830 reviewed proteins from UniProt (Section 2.1), we create 12 datasets (Section 2.2) capturing different aspects of data bias, including sequence redundancy, number of TFs vs. non-TFs (i.e., class imbalance ratios) (Section 2.3), and 3D protein structural quality, that can influence the training of a machine learning model (Figure 2). Given a dataset, we extract

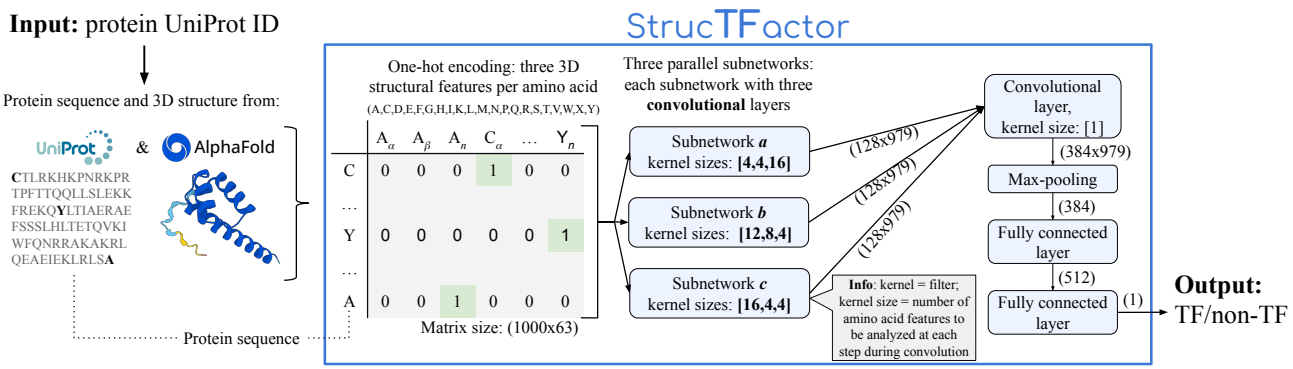

Figure 1. Overview of our TF prediction method: StrucTFactor. See Section 2 for details.

features of proteins based on their sequences or 3D structures, and then use the protein features (Section 2.4) along with a CNN-based deep learning architecture (Section 2.5) to train a model for TF prediction. We evaluate each trained model using four performance evaluation measures (Section 2.6).

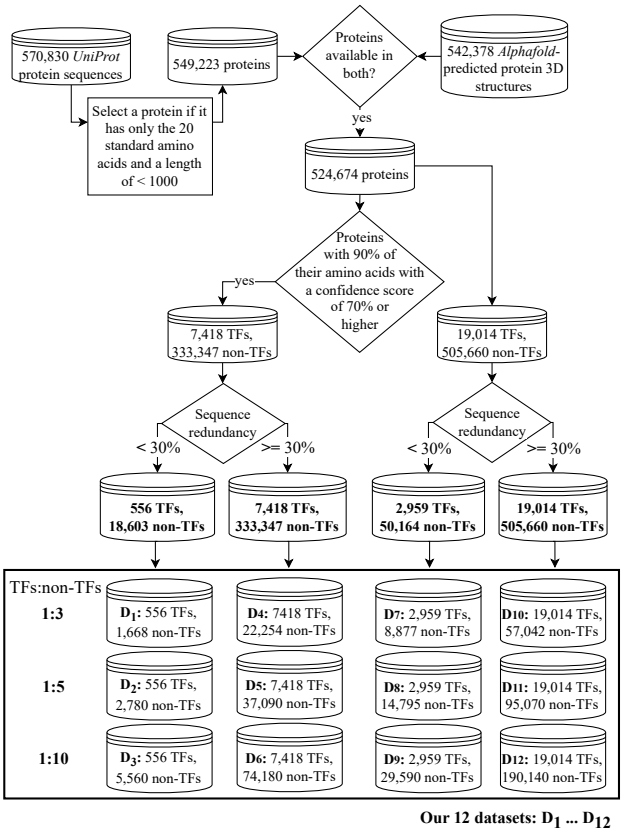

Figure 2. Overview of the creation of our 12 datasets (Section 2.2).

2.1. Collection of protein data

We use all 570,830 reviewed proteins from UniProt (last download on 22 Feb. 2024 from `https://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot.dat.gz`). We only consider a protein further if it i) has less than or equal to 1,000 amino acids, ii) only contains the 20 standard amino acids, and iii) is present in AlphaFold (Varadi et al., 2021) (downloaded on March 2023 from `https://ftp.ebi.ac.uk/pub/databases/alphafold/latest/swissprot_pdb_v4.tar`), resulting in 524,674 proteins.

2.2. Creation of datasets

Given the considered 524,674 proteins (Section 2.1), we use them in two ways depending on their predicted 3D structural quality. First, we consider only confidently predicted 3D protein structures (named *confidentAF*), based on an established criterion (Akdel et al., 2022) explained as follows. Given a protein, AlphaFold provides a "confidence score" for each amino acid in the protein. We include a protein in confidentAF if and only if the confidence scores of 90% of its amino acids are 70% or higher, resulting in 340,765 proteins. Second, we consider all 524,674 predicted 3D protein structures (named *allAF*).

Out of all 340,765 confidentAF proteins, there are 7,418 TFs and 333,347 non-TFs (Section 2.3). Because the number of TFs (7,418) is considerably less than the number of non-TFs (333,347), there is a class imbalance problem that can negatively impact the training of a machine learning model (Megahed et al., 2021). Hence, we take three random samples from all 333,347 non-TFs, such that the ratio between the numbers of TFs and non-TFs is 1:3, 1:5, and 1:10, resulting in three datasets named D_1, D_2, and D_3, respectively (Figure 2). Sequence redundancy between proteins within and across training and testing data could negatively impact the generalization ability of the trained model (Hou et al., 2022). Hence, we remove sequence redundancy among the $7,418 + 333,347 = 340,765$ proteins using the Many-against-Many sequence searching (MMSeqs2) tool (Steinegger and Söding, 2017) to only keep proteins with pairwise sequence similarities of less than 30% (Supplementary Section S2), resulting in 19,140 (with 556 TFs and 18,584 non-TFs) sequence non-redundant proteins. Then, we remove the class imbalance problem similarly as we do above. That is, we take three random samples from all 18,584 non-TFs, such that the ratio between the numbers of TFs and non-TFs is 1:3, 1:5, and 1:10, resulting in three datasets named D_4, D_5, and D_6, respectively (Figure 2). Thus, we create six datasets using confidentAF. Similarly, we create six datasets, i.e., D_7, D_8, D_9, D_{10}, D_{11}, and D_{12}, corresponding to allAF (see Figure 2 and Supplementary Section S1 for details).

2.3. Transcription factor annotation

To label a protein as a TF or not, we rely on existing criteria (Kim et al., 2021) using Gene Ontology (GO) (Consortium, 2015). We label a protein as a TF if it is associated with either i) a TF GO term or ii) a transcription regulation GO term in combination with a DNA binding GO term, else we label the protein as a non-TF. We extract protein-GO term associations from UniProt. As previously done (Kim et al., 2021), we consider 36 GO terms that contain 21, 11, and four GO terms in the TF, transcription regulation, and DNA-binding categories, respectively (Supplementary Table S1).

2.4. Protein features

2.4.1. Existing sequence-based features

DeepTFactor represents a protein as a one-hot encoding matrix M_{seq} of size $L \times |R|$, where L represents the length of the protein and R represents the ordered (alphabetically based on the one-letter-codes of amino acids) set of the standard 20 amino acids along with a placeholder (with the one-letter-code of "X") for possible non-standard amino acids. Thus, the size of R, i.e., $|R|$ is 21. A position $[i, j]$ in the matrix M_{seq} is 1 if the amino acid at the ith position in the protein sequence is the jth amino acid in R, else position $[i, j]$ in the matrix M_{seq} is 0.

2.4.2. Our proposed 3D structure-based features

We extract 3D structural features of proteins based on their secondary structural information using dictionary of protein secondary structure (DSSP) (Kabsch and Sander, 1983). DSSP annotates each amino acid of a protein structure as either α-helix (H), residue in isolated β-bridge (B), extended strand participating in β ladder (E), 3-helix (G), 5-helix (I), hydrogen-bonded turn (T), or bend (S). We categorize each amino acid of a protein into one of the three secondary structural groups, i.e., α-helix, β-sheet, or coil-turn based on

the DSSP output. That is, we consider an amino acid to be in the i) α-helix group if the corresponding DSSP output is either G, H, or I, ii) β-sheet group if the corresponding DSSP output is either B or E, or iii) coil-turn group if the corresponding DSSP output is either S or T. We represent a protein as a one-hot encoding matrix M_{str} of size $L \times |S|$, where L represents the length of the protein and S represents all possible combinations of the 21 amino acids (the 20 standard amino acids along with a placeholder for possible non-standard amino acids) and the above-defined three secondary structural groups, and thus the size of S, i.e., $|S|$ is $21 \times 3 = 63$. The set S is first ordered alphabetically based on the one-letter-codes of the amino acids and then, for a given amino acid, in the order of α-helix, β-sheet, and coil-turn. A position $[i, j]$ in the matrix M_{str} is 1 if the amino acid with certain secondary structural group annotation (i.e., either α-helix, β-sheet, or coil-turn) at the ith position in the protein sequence is the jth amino acid paired with the corresponding secondary structural group annotation in S, else position $[i, j]$ in the matrix M_{str} is 0.

2.5. Our deep learning model architecture

Our deep learning (DL) architecture is similar to DeepTFactor with some adaptations due to the different size of our proposed input protein feature matrix compared to the input protein feature matrix of DeepTFactor (Section 2.4). We keep our DL architecture as similar to DeepTFactor as possible to give the best-case advantage to DeepTFactor. Additionally, the aim of our study is to evaluate whether 3D structural features of proteins can better perform in the TF prediction task than sequence features. Hence, if we drastically change our DL architecture compared to the DL architecture of DeepTFactor, then, if our proposed 3D structure-based protein features outperform sequence features of DeepTFactor, it would not be clear whether the performance gain is due to our novel 3D structural features or the novel DL architecture.

Our DL architecture is based on convolutional neural networks (CNNs) with an input layer of size 1000×63. The CNN has three subnetworks (termed a, b, and c), each consisting of three convolutional layers with an attached batch normalization, a rectified linear unit (ReLU), and a dropout layer. The kernel sizes of the three convolutional layers within the three subnetworks are [4,63], [4,1], and [16,1] for subnetwork a, [12,63], [8,1], and [4,1] for subnetwork b and [16,63], [4,1], and [4,1] for subnetwork c. The output of each of the three subnetworks is concatenated into a single feature map, which goes into a convolutional layer followed by batch normalization, ReLU, max-pooling, and two fully connected layers with batch normalization. The first fully connected layer uses ReLU as an activation function, while the second fully connected layer outputs a single value using a Sigmoid function.

2.6. Model training and evaluation

Given a dataset, we evaluate a model in a 5-fold cross-validation manner. That is, given all proteins in a dataset, we divide the proteins into five non-overlapping subsets with stratification. By stratification, we mean that the ratio of TFs and non-TFs in each subset is the same as the ratio of TFs and non-TFs in the complete dataset. For each subset, we use its proteins as the test data and all of the proteins in the remaining four subsets as the training data. For training, we use the batch size of 120 proteins and train for 50 epochs. Over the two different features (see Section 2.4), 12 datasets (see Section 2.2), and five cross-validation folds, we train and test $2 \times 12 \times 5 = 120$ TF prediction models.

We evaluate each trained model with respect to the corresponding test data using four performance measures: Area under precision-recall curve (AU-PRC), Area under receiver operating characteristic curve (AU-ROC), precision, and MCC. For precision and MCC, we define a threshold of 0.5, meaning that we predict a protein to be a TF if its TF prediction score is > 0.5 and we predict a protein to be non-TF if its TF prediction score is ≤ 0.5. Precision is the percentage of all proteins with a prediction score of > 0.5 that are TFs. MCC is effective in evaluating model performance on imbalanced datasets because it gives a high score if and only if the prediction is good in all categories (i.e., true positives, false positives, true negatives, and false negatives) of the confusion matrix (Chicco and Jurman, 2020).

2.7. Interpretability of trained models

To gain insights into the features on which a trained model bases its predictions, we evaluate the interpretability of the model using the integrated gradient method (Sundararajan et al., 2017) from the captum library (Kokhlikyan et al., 2020) of PyTorch (Paszke et al., 2019). We use the models of StrucTFactor or DeepTFactor trained on the most confident dataset, i.e., D_1. Given a trained model and the input matrix of size $L \times Q$ (where L is the length of the protein and Q is the number of features with the value of either 21 for DeepTFactor or 63 for StrucTFactor, see Section 2.4) of a protein, the integrated gradient method outputs a matrix of the same size as the input matrix, with a score at each cell of the output matrix indicating the impact of the input feature corresponding to that cell on the predicted result. A score can either be positive or negative, where a negative score means that the input feature at the corresponding cell supports the model towards a non-TF prediction and a positive score means that the input feature at the corresponding cell supports the model towards a TF prediction. Given the output matrix with integrated gradient scores for each cell, we take the sum of each row to get one integrated gradient score per amino acid.

3. Results

To evaluate whether 3D structural information can more correctly distinguish TFs from non-TFs than sequences, we compare our proposed secondary structure-based TF prediction method (StrucTFactor) with the existing state-of-the-art sequence-based TF prediction method (DeepTFactor). First, we compare the performances of the considered methods on the most confident dataset with minimum class imbalance and sequence non-redundant proteins (Section 3.1). Second, to evaluate the effect of class imbalance ratio, we compare the performances of the methods across three (D_1, D_2, and D_3) datasets with different class imbalance ratios (Section 3.2). Third, to evaluate the effect of sequence redundancy, we compare the performances of the methods across six (D_1 to D_6) datasets with two sequence redundancy choices (Section 3.3). Until this point, we only consider confidently predicted 3D protein structures. Hence, fourth, to evaluate the effect of confidently vs. all predicted 3D protein structures, we compare the performances of the methods across all 12 (D_1 to D_{12}) datasets (Section 3.4). Fifth, to explain why StrucTFactor perform well on the TF prediction task, we dig deeper to look into the protein structural parts that contribute towards

StrucTFactor's good performance (Section 3.5). Sixth, we perform literature validation of top novel TF predictions of StrucTFactor (Section 3.6).

3.1. StrucTFactor outperforms DeepTFactor on the most confident dataset

We compare StrucTFactor and DeepTFactor based on the most confident dataset (D_1) that includes sequence non-redundant (< 30% pairwise sequence similarity) confidently predicted 3D protein structures with the least difference between the number of TFs and non-TFs (Section 2.2). We find that StrucTFactor outperforms DeepTFactor by a large margin (Figure 3).

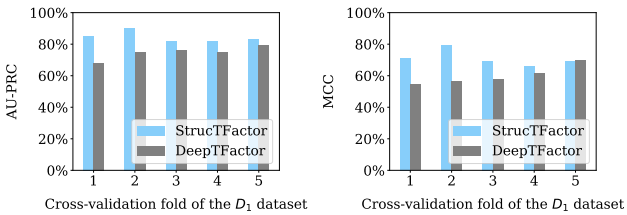

Figure 3. Performances of the StrucTFactor (blue) and DeepTFactor (gray) based on AU-PRC (left panel) and MCC (right panel) for the most confident dataset, i.e., D_1.

With respect to AU-PRC, StrucTFactor achieves ∼82% to ∼92% scores, outperforming DeepTFactor by a margin of at least ∼4% and going up to ∼17%, depending on the cross-validation fold (Figure 3 and Supplementary Table S2). With respect to MCC, StrucTFactor achieves ∼66% to ∼80% scores. For four out of the five cross-validation folds, StrucTFactor outperforms DeepTFactor by a margin of at least ∼4% and going up to ∼23%, while for the fifth cross-validation fold the performances of StrucTFactor and DeepTFactor are comparable with a difference of only ∼0.5% (Figure 3). Thus, using secondary structural information (as in StrucTFactor), instead of just the sequence information (as in DeepTFactor), can more accurately distinguish between TFs and non-TFs.

3.2. StrucTFactor outperforms DeepTFactor irrespective of class imbalance ratio

We evaluate the performances of the considered methods across different class imbalance ratios (i.e., number of TFs and non-TFs) using D_1 (with three times more non-TFs than TFs), D_2 (with five times more non-TFs than TFs), and D_3 (with ten times more non-TFs than TFs) datasets. In general, because a high class imbalance ratio can negatively impact the generalization capacity of a trained machine learning model (Megahed et al., 2021), we expect to see a decrease in performance for each of StrucTFactor and DeepTFactor with an increase in the class imbalance ratio.

Interestingly, we find that for both StrucTFactor and DeepTFactor, their performances improve when going from D_1 to D_2, while their performances are worst for the dataset with the highest class imbalance ratio (D_3) (Figure 4). For example, with respect to MCC, the performance values for StrucTFactor are ∼71.1%, ∼75.5%, and ∼70.8% for D_1, D_2, and D_3, respectively. The above results could be due to the trade-off between the class imbalance ratios and the number of proteins, both of which can affect the training of a machine learning model. While D_1 has the least class imbalance ratio, it also has the least number (876) of proteins on which a machine learning mode can be trained. On the other hand, while D_2 has a bit higher class imbalance ratio than D_1, there is a higher number (1,314) of proteins to train a machine learning model. Lastly, for the D_3 dataset, although there is a higher number (6,116) of proteins, the class imbalance ratio is relatively large which can negatively affect the training of a more generalized machine learning model.

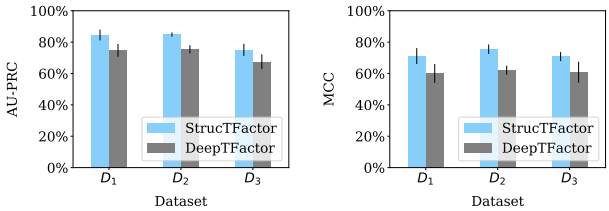

Figure 4. Performances (average over the five cross-validation folds) of the StrucTFactor (blue) and DeepTFactor (gray) based on AU-PRC (left panel) and MCC (right panel) for the D_1, D_2, and D_3 datasets.

Nevertheless, irrespective of the class imbalance ratio across the three datasets (D_1, D_2, and D_3), StrucTFactor consistently outperforms DeepTFactor in terms of both AU-PRC and MCC (Figure 4). With respect to AU-PRC, StrucTFactor outperforms DeepTFactor by a margin of at least ∼7.5% and going up to ∼10% depending on the dataset. With respect to MCC, StrucTFactor outperforms DeepTFactor by a margin of at least ∼10% and going up to ∼13% depending on the dataset. To quantify whether the performance of StrucTFactor is significantly better than DeepTFactor, we compare their performance values over all 15 combinations of the three datasets and five cross-validation folds, using the paired Wilcoxon signed-rank test (Woolson, 2008). We find that StrucTFactor significantly (p-value < 0.001) outperforms DeepTFactor in terms of all four performance measures.

3.3. StrucTFactor outperforms DeepTFactor irrespective of sequence redundancy

To see how the performances of each of StrucTFactor and DeepTFactor vary with different sequence redundancy, we compare them on datasets with sequence non-redundant (pairwise sequence similarity of <30%) proteins, i.e., D_1, D_2, or D_3, as well as on datasets with sequence redundant (pairwise sequence similarity of $\geq$ 30%) proteins, i.e., D_4, D_5, or D_6. In general, because high sequence similarities in a dataset can artificially boost the performances of trained machine learning models (Hou et al., 2022), we expect to see an increase in the performance for both StrucTFactor and DeepTFactor with an increase in sequence redundancy, which is exactly what we find (Table 1). For example, with respect to MCC, the performance for DeepTFactor is ∼60% for D_1 (a sequence non-redundant dataset), while it is ∼96.5% for the corresponding sequence redundant dataset D_4.

Even for the non-redundant datasets, where the scope of improvement is minimal (DeepTFactor already achieves stellar performances close to 100% as presented above), for each of the three sequence redundant datasets (D_4, D_5, and D_6), StrucTFactor consistently outperforms DeepTFactor in terms all considered performance measures (Table 1 and Supplementary Table S3).

Table 1. Variation in the performances of StrucTFactor and DeepTFactor with different sequence redundancies. The dataset D_1 has sequence redundancy of < 30%, while the dataset D_4 has sequence redundancy of ≥ 30%. Given a dataset and a performance measure, the results are shown as the average of the five cross-validation folds, and the best method performance is in bold font. Results are for the datasets with the 1:3 class imbalance ratio, i.e., three times more non-TFs than TFs. Results for the 1:5 and 1:10 ratios are qualitatively similar (Supplementary Table S3).

	AU-PRC		MCC	
	D_1	D_4	D_1	D_4
StrucTFactor	**0.847**	**0.988**	**0.711**	**0.975**
DeepTFactor	0.747	0.986	0.60	0.965

3.4. StrucTFactor is robust across different 3D protein structural quality

All of the above results correspond to confidently predicted 3D protein structures, i.e., confidentAF, which contains around 10 times less number of proteins than using all protein data, i.e., allAF (Section 2.2 and Supplementary Section S1). To investigate the impact of this reduced number of proteins in confidentAF than in allAF on the performance of a machine learning model, we evaluate the performances of the considered methods on allAF datasets (D_7 to D_{12}) and compare them against the performances of the considered methods on confidentAF datasets (D_1 to D_6). Although allAF datasets have many non-confident 3D protein structures, we find that the StrucTFactor performs better (or comparable) for allAF datasets than for confidentAF datasets (Table 2). Additionally, similar to confidentAF datasets (see previous subsections of results), StrucTFactor outperforms DeepTFactor on allAF datasets (Table 2), although with a smaller margin than on the confidentAF datasets.

Table 2. Variation in the performances of StrucTFactor and DeepTFactor across protein 3D structural quality. The dataset D_1 has only confidently predicted 3D structures, while the dataset D_7 has all predicted 3D structures, from AlphaFold. Given a dataset and a performance measure, the results are shown as the average of the five cross-validation folds, and the best method performance is in bold font. Results are for the datasets with the 1:3 class imbalance ratio. Results for the 1:5 and 1:10 ratios are qualitatively similar (Supplementary Table S4).

	AU-PRC		MCC	
	D_1	D_7	D_1	D_7
StrucTFactor	**0.847**	**0.865**	**0.711**	**0.712**
DeepTFactor	0.747	0.828	0.60	0.631

The above results could be due to the following reasons. The number of proteins in allAF datasets is larger than the number of proteins in the confidentAF datasets, which could help in a better training of a machine learning model. Thus, for both StrucTFactor and DeepTFactor, their performances for allAF datasets are, in general, higher than their performances for the corresponding confidentAF datasets. However, the additional proteins in allAF datasets are of lower 3D structural quality. Because StrucTFactor (unlike DeepTFactor), relies on 3D structural features of proteins, the increase in the number of proteins in allAF datasets than in confidentAF datasets does not boost the performance of StrucTFactor as much as it does for DeepTFactor. Nevertheless, to statistically quantify the superiority of StrucTFactor over DeepTFactor, we compare their performance values over all 30 combinations of the six allAF datasets and five cross-validation folds, using the paired Wilcoxon signed-rank test. We find that StrucTFactor significantly outperforms DeepTFactor in terms of AU-PRC (p-value < 0.001), MCC (p-value < 0.001), and precision (p-value ∼0.022), while StrucTFactor marginally significantly (p-value ∼0.052) outperforms DeepTFactor in terms of AU-ROC.

3.5. StrucTFactor captures domain binding regions of transcription factors

To get insights into why StrucTFactor performs well on the TF prediction task, we characterize the specific regions of a protein that StrucTFactor focuses on to make predictions. Given a protein, we use the concept of integrated gradients that gives a score to each sequence position of the TF (Section 2.7), where a positive score supports the prediction of the protein as a TF, while a negative score supports the prediction of the protein as a non-TF. We do this for each of the 376 TFs from the most confident dataset, i.e., D_1, for which there is DBD information in UniProt. Given the DBD information of a TF, we take the sum of the scores of each sequence position in the DBD region as the DBD-related score and the sum of the scores of each sequence position not present in the DBD region as the non-DBD-related score, of the protein. We find that the DBD-related scores of TFs for StrucTFactor are predominantly positive and are in general higher than the DBD-related scores of TFs for DeepTFactor (Figure 5). This signifies that StrucTFactor focuses on the DBD regions of known TFs more than DeepTFactor, in order to predict them as TFs. Additionally, the difference between the DBD-related scores and non-DBD-related scores are significantly more for StrucTFactor than for DeepTFactor (Figure 5). This signifies that the focus of StrucTFactor on the DBD regions is less noisy than the focus of DeepTFactor on the DBD regions (Figure 6).

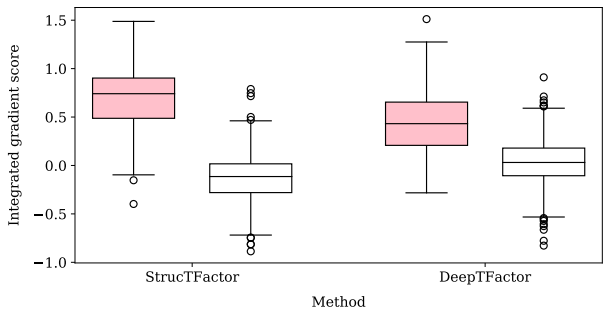

Figure 5. Distribution of DBD-related (pink color) and non-DBD-related (no color) integrated gradient scores of the 376 TFs from the D_1 dataset.

3.6. Literature validation

To explore whether StrucTFactor can potentially predict novel TFs not captured by DeepTFactor, we analyze the top five (based on the prediction score) false positive predictions (labeled as non-TFs but predicted as TFs) of StrucTFactor and compared them against the top five false positive predictions of DeepTFactor, from the most confident dataset, i.e., D_1. We find that two proteins (out of the chosen five false positives), i.e., *rstR2* (UniProt ID O34419) and *envR* (UniProt ID P0ACT3), are uniquely captured by StrucTFactor. We search these two

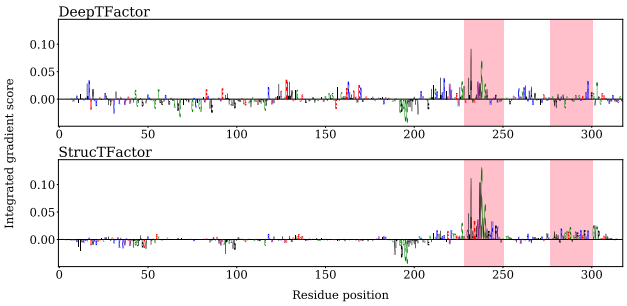

Figure 6. Integrated gradient score of each sequence position of the YqhC protein(UniProt ID *Q46855*). We highlight the known DBD region in pink. Each sequence position shows the corresponding amino acid with its size and orientation corresponding to its integrated gradient scores.

proteins in the literature. We find that *rstR2* is a DNA-binding transcription repressor in Vibrio cholarae (Krin et al., 2018). Similarly, *envR* is a DNA-binding transcription repressor in Escherichia coli with an entry in RegulonDB (Gama-Castro et al., 2010), as well as Salmonella enterica (Blair et al., 2022). Interestingly, while envR is ranked among the top five novel predictions by StrucTFactor, it is ranked relatively low (rank 141) in the list of novel predictions by DeepTFactor.

4. Conclusion

We propose a novel data-driven TF prediction method called StrucTFactor which is the first of its kind to use 3D structural features of proteins. Our comprehensive evaluation shows that StrucTFactor significantly outperforms the existing state-of-the-art TF prediction method that relies only on protein sequences. Thus, our results highlight the importance of considering 3D protein structures for the TF prediction task. More importantly, we perform our experiments with the predicted 3D structures of proteins, which shows that the recent advancements in 3D protein structural predictions can be positively utilized to model and understand TFs. In this work, we focus on the secondary structural characteristics of proteins to capture their 3D structures. Future research focusing on tertiary structural characteristics of proteins will potentially further improve the TF prediction task.

5. Competing interests

No competing interest is declared.

6. Author contributions statement

Conceived the study: F.N. and K.N. Collected and processed the data: F.N. and J.L. Designed the methodology: F.N. and K.N. Designed the experiments: F.N., J.L. and K.N. Performed the experiments: F.N. and J.L. Analysed the results: F.N., J.L. and K.N. Wrote the paper: F.N., J.L. and K.N. Read and approved the paper: F.N., J.L., J.B., and K.N. Supervised the study: K.N.

7. Funding

This work was supported by Universität Hamburg and HamburgX grant LFF-HHX-03 to the Center for Data and Computing in Natural Sciences (CDCS) from the Hamburg Ministry of Science, Research, Equalities and Districts. This work was supported by the German Federal Ministry of Education and Research (BMBF) within the framework of the *e:Med* research and funding concept (*grants 01ZX1910D and 01ZX2210D*). This work was developed as part of the PoSyMed project and is funded by the German Federal Ministry of Education and Research (BMBF) under grant number 031L0310A.

References

M. Akdel et al. A structural biology community assessment of alphafold2 applications. *Nature Structural & Molecular Biology*, 29(11):1056–1067, Nov 2022.

S. F. Altschul et al. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

S. F. Altschul et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 09 1997.

J. M. A. Blair et al. EnvR is a potent repressor of acrAB transcription in Salmonella. *Journal of Antimicrobial Chemotherapy*, 78(1):133–140, 10 2022.

D. Chicco and G. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6, Jan 2020.

G. O. Consortium. Gene ontology consortium: going forward. *Nucleic Acids Research*, 43(Database issue):D1049–D1056, 2015.

U. Consortium. Uniprot: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515, 2019.

Z. Du et al. Predicting tf proteins by incorporating evolution information through pssm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(2):1319–1326, 2023.

J. Eichner et al. Tfpredict and sabine: Sequence-based prediction of structural and functional characteristics of transcription factors. *PLOS ONE*, 8(12):1–12, 12 2013.

S. Gama-Castro et al. RegulonDB version 7.0: transcriptional regulation of Escherichia coli K-12 integrated within genetic sensory response units (Gensor Units) . *Nucleic Acids Research*, 39(suppl_1):D98–D105, 11 2010.

M. Geertz and S. J. Maerkl. Experimental strategies for studying transcription factor–DNA binding specificities. *Briefings in Functional Genomics*, 9(5-6):362–373, 09 2010.

H. Guo et al. Weighted graphlets and deep neural networks for protein structure classification, 2019.

Q. Hou et al. Ten quick tips for sequence-based prediction of protein properties using machine learning. *PLOS Computational Biology*, 18(12):1–15, 12 2022.

J. Jumper et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, August 2021.

W. Kabsch and C. Sander. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.

G. B. Kim et al. Deeptfactor: A deep learning-based tool for the prediction of transcription factors. *Proceedings of the National Academy of Sciences*, 118(2):e2021171118, 2021.

N. Kokhlikyan et al. Captum: A unified and generic model interpretability library for pytorch, 2020.

E. Krin et al. Expansion of the sos regulon of vibrio cholerae through extensive transcriptome analysis and experimental

validation. *BMC Genomics*, 19(1):373, May 2018.

S. K. Kummerfeld and S. A. Teichmann. DBD: a transcription factor prediction database. *Nucleic Acids Research*, 34 (suppl_1):D74–D81, 01 2006.

S. A. Lambert et al. The human transcription factors. *Cell*, 172(4):650–665, Feb 2018.

D. S. Latchman. Transcription factors: An overview. *The International Journal of Biochemistry & Cell Biology*, 29 (12):1305–1312, 1997.

Z. Lin et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

F. M. Megahed et al. The class imbalance problem. *Nature Methods*, 18(11):1270–1272, Nov 2021.

K. Newaz et al. Network-based protein structural classification. *Royal Society Open Science*, 7(6):191461, 2020.

K. Newaz et al. Multi-layer sequential network analysis improves protein 3d structural classification. *Proteins: Structure, Function, and Bioinformatics*, 90(9):1721–1731, 2022.

A. Paszke et al. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

M. Steinegger and J. Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017.

M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 06–11 Aug 2017.

M. Varadi et al. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, 50(D1):D439–D444, 11 2021.

D. Wilson et al. DBD—taxonomically broad transcription factor predictions: new content and functionality. *Nucleic Acids Research*, 36(suppl_1):D88–D92, 12 2007.

R. F. Woolson. *Wiley Encyclopedia of Clinical Trials*, chapter Wilcoxon Signed-Rank Test. John Wiley & Sons, Inc., Hoboken, NJ, USA, Sept. 2008.

G. Zheng et al. The combination approach of svm and ecoc for powerful identification and classification of transcription factor. *BMC Bioinformatics*, 9(1):282, Jun 2008.