

SOFTWARE

Open Access



SNPeBoT: a tool for predicting transcription factor allele specific binding

Patrick Gohl¹ and Baldo Oliva^{1*}

*Correspondence:
baldo.oliva@upf.edu

¹ Department of Medicine
and Life Sciences, SBI-GRIB,
Universitat Pompeu Fabra,
08003 Barcelona, Catalonia, Spain

Abstract

Background: Mutations in non-coding regulatory regions of DNA may lead to disease through the disruption of transcription factor binding. However, our understanding of binding patterns of transcription factors and the effects that changes to their binding sites have on their action remains limited.

Summary: To address this issue we trained a Deep learning model to predict the effects of Single Nucleotide Polymorphisms (SNP) on transcription factor binding. Allele specific binding (ASB) data from Chromatin Immunoprecipitation sequencing (ChIP-seq) experiments were paired with high sequence-identity DNA binding Domains assessed in Protein Binding Microarray (PBM) experiments. For each transcription factor a paired DNA binding Domain was selected from which we derived E-score profiles for reference and alternate DNA sequences of ASB events. A Convolutional Neural Network (CNN) was trained to predict whether these profiles were indicative of ASB gain/loss or no change in binding. 18211 E-score profiles from 113 transcription factors were split into train, validation and test data. We compared the performance of the trained model with other available platforms for predicting the effect of SNP on transcription factor binding. Our model demonstrated increased accuracy and ASB recall in comparison to the best scoring benchmark tools.

Conclusion: In this paper we present our model SNPeBoT (Single Nucleotide Polymorphism effect on Binding of Transcription Factors) in its standalone and web server form. The increased recovery and prediction accuracy of allele specific binding events could prove useful in discovering non-coding mutations relevant to disease.

Keywords: Transcription factor, Gene regulation, Neural network

Background

Regulation of gene expression is mediated in part by transcription factors (TF), these proteins bind to sequence specific strands of DNA, thereby acting in concert with enhancer, silencer or insulator elements to modulate the expression of associated genes. The presence or absence of these TF lead to much of the differential gene expression across an organism's developmental stages [1], cell cycle [2] and cell types [3]. However, SNP within TF binding sites can lead to local increase or decrease in TF binding affecting gene expression and consequently causing disease [4, 5]. Therefore, the prediction



© The Author(s) 2025. **Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

of ASB may prove useful in elucidating disease etiology and processing Genome Wide Association Study (GWAS) data. Tools already exist for predicting the effects of SNPs on TF binding [6–8] including Deep learning based models [9]. However, many tools, while relatively simple for experienced users, retain prohibitive installation or implementation requirements for the general user. They may also require, in the case of deep learning models, fine tuning before and therefore necessitate the generation of large amounts of training data which isn't always readily available to the user. Additionally, we show that existing tools leave many ASB events unidentified. Increasing the number of SNPs that can be identified as potentially altering gene expression regulation may lead to the discovery of etiologically significant mutations. We have trained a PBM E-score based Convolutional Neural Network on ASB data to predict when a SNP results in TF gain, loss or no-change in binding. Our model demonstrated an improved recovery of SNPs resulting in changes to TF binding as well as accuracy with regards to MotifbreakR [6] and atSNP [7]. We have made SNPeBoT available on a web server where it is freely available for use, as well as provided a standalone version on github.

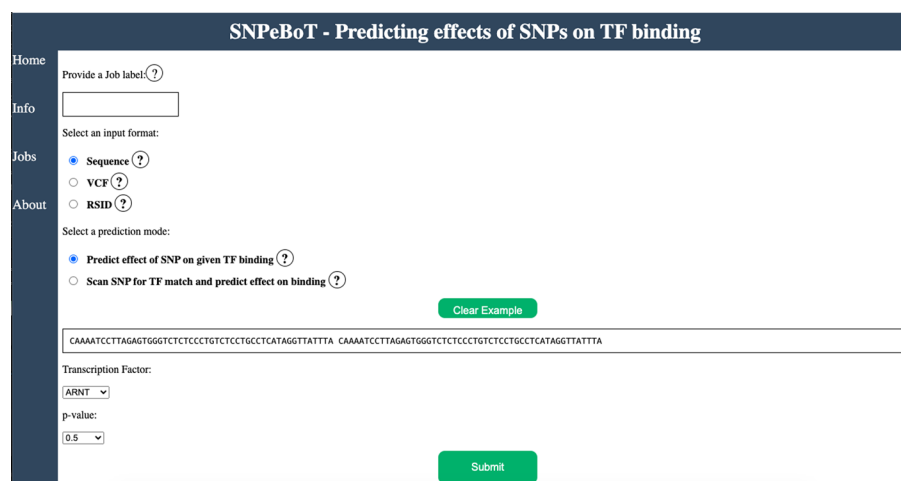
Implementation

Input/output

SNPeBoT supports 2 methods for retrieving predictions: a Webserver and a Standalone package. We suggest that users install the standalone packages in cases where more than 15 SNPs are to be evaluated, for all other cases the web server provides the simplest application of our predictor.

Web server

In order to make a prediction using the web server the user is required to submit an input following the prompts of the website (Fig. 1). First, a unique job identifier will be submitted followed by a selection of the format for the input SNPs. Three formats are supported: (1) A reference and alternate DNA strand, tab separated, of 51 nucleotides long each. If the sequence information beyond 7 bases up and downstream of the



The screenshot shows the SNPeBoT web application interface. The title bar reads "SNPeBoT - Predicting effects of SNPs on TF binding". On the left is a navigation menu with links: Home, Info, Jobs, and About. The main content area contains the following fields and options:

- Provide a Job label:** A text input field with a help icon.
- Select an input format:** Three radio button options:
 - ☒ Sequence (with a help icon)
 - ☐ VCF (with a help icon)
 - ☐ RSID (with a help icon)
- Select a prediction mode:** Two radio button options:
 - ☒ Predict effect of SNP on given TF binding (with a help icon)
 - ☐ Scan SNP for TF match and predict effect on binding (with a help icon)
- Clear Example:** A green button.
- Sequence Input:** A text area containing the example sequence: "CAAAATCCTTAGAGTGGGCTCTCCCTGCTCTCCCTCATAGGTATTATTA CAAAATCCTTAGAGTGGGCTCTCCCTGCTCTCCCTCATAGGTATTATTA".
- Transcription Factor:** A dropdown menu currently showing "ARNT".
- p-value:** A dropdown menu currently showing "0.5".
- Submit:** A green button.

Fig. 1 Screenshot of the SNPeBoT website

sequence midpoint is missing (a 15 bp sub-sequence at the center of each strand) padding may be added without affecting the prediction. (2) Variant call format (VCF) SNP data where columns (in order) contain the following: chromosome number (eg. chr1), Allele location (in hg38), an identifier for the SNP, reference allele, alternate allele. (3) Reference Single nucleotide polymorphism IDs (RSIDs). Next, the user will select from two prediction options: (1) Making a prediction of the effect of the SNP on a particular TF. In this case, the user will be able to input multiple SNPs, up to a limit of 15, and additionally make a selection of the supported TFs in the webservice that must be tested. SNPeBoT processes all information in the sequence format, thus vcf and rsid mutations are first converted to sequences. Due to server constraints, extra memory and computation allotted to conversion means that users are limited to 15 and 1 mutation for vcf and rsid inputs respectively. Or 2) Submitting a single SNP to predict the effect on multiple TFs from a range selected out of a drop down menu. Finally, in both cases the user is asked to select a p-value threshold for Find Individual Motif Occurrences (FIMO) [10] within the post-hoc prediction filtration process and submit the job. Once the job is done, the predictions will either appear on screen as “loss”, “gain”, or “no change” where the SNP was predicted to cause the given TF to either decrease, increase or have no effect on the binding of the TF. The user can also access the job by navigating to the jobs window and searching for the jobs ids to retrieve the results.

Standalone

For the standalone version there are two required input files. The first file contains the SNPs. This file should have three tab separated columns with the TF in the first column followed by the reference and alternate sequences (each sequence is 51 bases long with the allele of interest at the center). The second file should contain a single column with all of the TFs to be tested (these must correspond with testable TFs). Once the bash script is executed the program will run and write all predictions to a file (Results/{TF}_Output.txt). The results will be displayed in two columns: the SNP Line column which shows what line of the input file the predicted SNP came from and the prediction column which yields predictions in the same format as the webserver. In the standalone package a python script is available for the conversion of rsids and vcf SNPs to sequences. We have provided more detailed instructions on the github regarding installation instructions and dependencies as well as the retrieval of E-scores data from Catalog of Inferred Sequence Binding Preferences (CISBP) [11] or other available PBM experiments.

Model architecture and training data

Data

ASB training data was retrieved from chip-seq experiments compiled in ADAstra [12]. This dataset is composed of 327,642 ASB events. After filtering out ASB events that may be occurring due to off-site effects (Non Concordant data) 34,374 data events remain. Further filtering out of data that could not be assigned to a PBM TF DNA binding Domain (TF-DBD) as outlined in the Data Processing section. Additionally all duplicated ASB points were removed to prevent train-test crossover and overfitting. This resulted in 9215 ASB events, 4987 and 4228 ASB events for loss and gain respectively. Control SNPs resulting in no change to TF binding were retrieved from the GTRD data

used by ADAstra. Thus, 8996 SNPs without significant ASB events were added to the total training set, which then consisted of 18,211 data points (Fig. 2). E-score values and TF Position Weight Matrix (PWM) were retrieved from the CISBP database.

Data processing

A single SNP prediction must undergo multiple levels of processing to yield the input features upon which the prediction is made. First the amino acid sequence of the TF acting upon the SNP is retrieved. Then, the TF sequence is compared using BLAST [13] against a database of TF-DBD sequences for which PBM experiments have been performed. Any TF-DBD with 70% or more sequence identity with the query TF is selected as a match. All TFs that did not pass the 70% sequence identity threshold for any of the TF-DBD included in PBM experiments could not be included in the training or testing of our model. These were excluded as no E-score profile would be generatable, hence SNPeBoT will also not be able to make a prediction on any such TF. This preprocessing step was handled by SBILib [14]. Next, reference and alternate sequences are retrieved from the UCSC genome browser [15] or genopyc [16] for VCF or RSID input format respectively. Subsequently, the PWMs for these TF-DBDs are scanned against both the alternate and reference sequences using FIMO. For each DNA sequence the PWM with the best scoring FIMO hit incorporating the mutated position is selected as the binding PWM and has its top score stored for CNN input. PBM experiments included in the CISBP database have measured the binding affinity of tested motifs against all permutations of 8 base length DNA sequences and assigned these an E-score, this means that in an ASB, for every 8-mer that includes the SNP we can derive an E-score as long as the relevant TF has a corresponding motif with PBM data. Therefore, the relevant TF-DBD associated E-scores for all octamers incorporating the mutated position are retrieved,

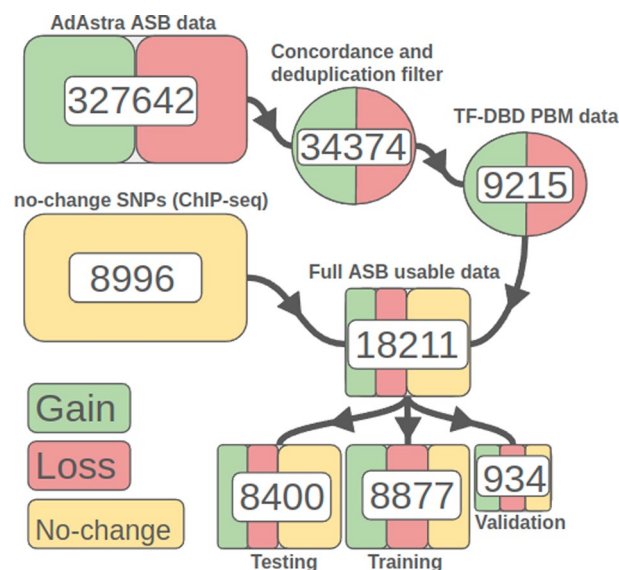


Fig. 2 Preprocessing stages of data. Each stage is labelled with the number of data points therein and color coded for the presence of classes (gain, loss and no-change). The distribution of classes for train, validation and testing data sets are (gain:2959, loss:2959, no-change:2959), (gain:129, loss:206, no-change:599), (gain:1140, loss:1822, no-change:5438) respectively

yielding a total of 16 E-scores. Next, a sliding window averages the E-scores assigned to each Nucleotide part of these octamers, 7 bases up and downstream of the mutated position, yielding a total of 30 E-score averages (15 for the reference and alternate sequences, respectively). These 30 scores and 2 Fimo hit scores were then transformed into a 8×4 matrix for CNN input.

CNN model

We trained a CNN using Tensorflow [17]. The model takes as input a (8,4,1) feature matrix. This is passed on to two initial 2D convolution layers and a pooling layer, followed by a second set of 2 convolution and 1 pooling layer. Finally the data is flattened, batch normalized and passed to 2 Dense layers with a final “softmax” activation function producing a 3 class output. The output corresponds to the probability of each prediction as “gain”, “loss”, and “no-change” in binding (see Fig. 3). The model underwent hyperparameter tuning using keras tuner [18] with the optimization objective being set to validation accuracy. The target parameters were our 2D convolutional layer filters as well as the dense layer units. Additionally, kernel regularizers were applied to each layer. To train the model, the dataset was split by selecting 70% of the smallest class count (gain) and matching these with data from the remaining two classes to yield a balanced training data set. From the test set a further split was made yielding a 10% 934 sample validation set and a final test set of 8400 samples. All data was shuffled prior to the splits to ensure unbiased class representation. The CNN model was applied to predict the outcome of 8400 test SNP consisting of 1140, 1822, 5438 ASBs for gain, loss and no change in transcription factor binding respectively. In this testing data SNPs were allowed even when they occurred in the training data set if the TF applied was unique (the TF-SNP pairing did not occur in training). To measure performance on completely unseen SNP we removed all seen SNPs regardless of TF and reran the prediction with our same model, additionally performance was measured on SNPs bound by TF not included in the training data.

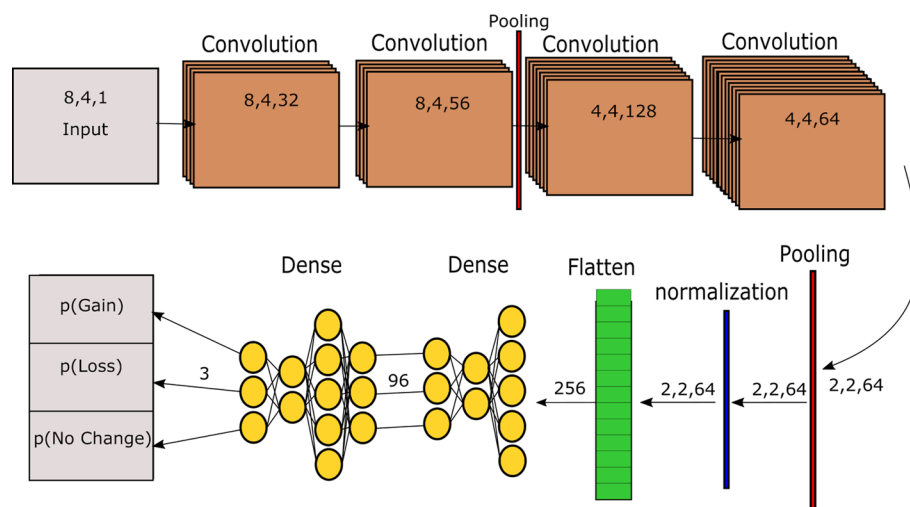


Fig. 3 Diagram of SNPeBot's CNN model architecture. Each layer in the model is labeled. The data shape at the output of each layer is shown either within the layer figure or immediately after

Post-hoc test

To distinguish between ASB directly influenced by DNA sequence variations and ASB arising from indirect factors, such as chromatin accessibility or linkage disequilibrium, we applied a post-hoc test. For each SNP the selected TF-DBD for both reference and alternate sequence had their PWMs scanned with FIMO along the sequence at the site of the mutation. SNPs without hits under a given p-value threshold for either the reference or the alternate sequence were labelled as no-change in binding. All remaining ASB predictions were kept as predicted by the CNN. To find out how much of the information from the final prediction came from the FIMO hits and how much came from SNPeBoT we conducted a comparative study. The new predictor was applied to the same testing data as before. Predictions were run first using SNPeBoT. Then FIMO was used to scan for hits using several p-value thresholds along the same ASB sequences. The hit count for both alternate and reference sequences were summed and compared. A prediction was made based on the relative hit count between the 2 sequences. A higher number of hits for the alternate sequence received a prediction of “gain” whereas lower count resulted in a “loss” prediction and the same number of hits meant that there was no-change in binding. The prediction accuracy and coverage for both methods was compared across all p-value thresholds used in this test. The best performing threshold was then used to test prediction accuracy and ASB recall when test data was grouped along transcription factor families. Another test was conducted on Enhanced Yeast 1 Hybrid (eY1H) experimental data [19] to measure prediction performance in the absence of off-site effects and on in vitro data. And finally a supplementary test was conducted at a p-value threshold of 0.001 on ASB events with testable TFs from the AlleleDB database [20] to test for generalizability of predictions across different data sources (Supplementary Information).

Benchmarking

The same ASB chip-seq data used for the post-hoc tests was then applied to two established tools for prediction of ASBs: MotifbreakR [6] and atSNP [7]. Both tools are available as R packages and were installed with all necessary dependencies. For both MotifbreakR and atSNP we followed the provided vignette tutorials. In MotifbreakR each transcription factor had its PWMs queried from MotifDB and predictions on the SNPs were made for these PWMs scoring with the relative entropy method at a threshold of 10^{-4} . Because MotifbreakR doesn't select a PWM to apply to each SNP we frequently retrieve multiple predictions per SNP, corresponding to the various PWMs for the given TF. To overcome this ambiguity, we summed all the allele differences for each prediction per SNP. Thus we were left with a single prediction of either “loss” or “gain” depending on whether the sum was positive or negative. All SNPs that did not yield a significant prediction were assigned as “no change” predictions. For atSNP, TFs publicly available PWMs from the CISBP database were downloaded. For each SNP-PWM pair the change in affinity scores were calculated and assigned a p-value by atSNP. The p-values underwent multiple test adjustments using Storey's q-value with the qvalue package [21]. We applied a qval_rank lower than 0.05 to retrieve effect predictions. This produced multiple predictions per SNP as it did with MotifbreakR, reflecting the number of PWMs per TF, for which we employed the same approach as before to retrieve a single

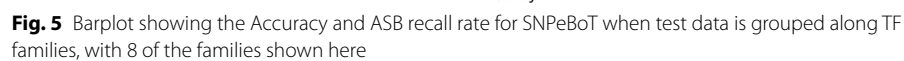
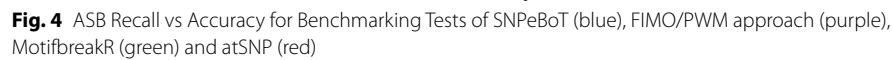
prediction. Prediction accuracy and coverage were calculated for both MotifbreakR and atSNP as well as SNPeBoT on a 3-class basis (gain, loss or no change). We then proceeded to measure performance for all three predictors with varying parameters altering the FIMO threshold for SNPeBoT and the PWM only method as well as the `qual_rank` for atSNP and motifbreakR thresholds. Next the area under the receiver operating characteristic curve was calculated for all three classes of predictions at their best performing threshold (based on accuracy of predictions). Due to the nature of Receiver Operating Characteristic ROC curve generation we were limited to classes that received a probability measure, since no-change classifications for atSNP and motifbreakR were any cases that did not return significant predictions. To generate class probability measures we converted prediction outputs to a range between 0 and 1 and attributed the complementary probability to the non predicted ASB class. We used resulting binary class probabilities to generate ROC curves for all benchmarking predictors, for full class ROC curves and further methodology information please refer to the SNPeBoT web server info page.

Results and discussion

We subjected SNPeBoT to multiple rounds of testing and benchmarking beginning with the initially trained CNN. Holdout testing of the CNN model prior to inclusion of the post-Hoc filter achieved 81% accuracy at 62% ASB recall with no distinction applied between the TFs for which predictions were made. Next we wanted to tighten our stringency on controlling for any possible information leakage between training and testing. When predictions were run on unseen SNPs regardless of the TF, the model achieved an accuracy of 80% with a 61% ASB recall rate. Accuracy and ASB recall when predicting on TFs not included in training for any SNPs was 83%. The maintenance of prediction performance even on this data seems to indicate that the model is able to generalize quite well from the training data and make accurate predictions even on completely new data. The performance of the CNN was improved by the implementation of a post-hoc filter. The highest accuracy SNPeBoT implementation (85%), while outperforming the CNN only based predictions on that measure, yielded a lower ASB recall rate (55%) than it. However, five SNPeBoT thresholds managed to produce results either matching or exceeding both metrics of prediction (Fig. 4). We believe that the inclusion of the post-hoc filter is justified by this increase in performance.

Benchmarking results

Next, we measured how SNPeBoT performed compared to other available predictors at 9 different thresholds. Our goal in measuring performance across multiple thresholds was to broaden the scope of testing to ensure that tools were compared at their highest capability. To test how much additional information SNPeBoT returned on top of PWMs we added a fourth predictor which based its classification of ASBs solely on comparison of binding scores of PWMs. SNPeBoT consistently outperformed this PWM-only method regardless of threshold in accuracy while simultaneously yielding higher ASB recall rates. This indicates that there is some information added by SNPeBoT which remains hidden from a scanning of affinity changes by FIMO. When grouping the TF by family, SNPeBoT yields a range of results (Fig. 5) with the best results produced by bZIP and HLH families. For a full list of families and



performances please refer to the Supplementary material (S. Figure 7). Benchmarking results using atSNP and motifbreakR were such that for any threshold result of these existing tools a threshold can be found for SNPeBoT that produced higher accuracy and ASB recall values (Fig. 4). To compare results at peak performance we selected

Table 1 Table of prediction accuracy of all three predictors at their respective best performing (accuracy) threshold levels. Rows represent the expected (true) labels for each point and the columns the predicted labels

Expected	SNPeBoT			motifbreakR			atSNP		
gain	709	88	370	603	9	541	391	187	578
lost	209	944	594	13	842	978	208	598	1027
no-change	319	301	5151	239	274	4945	143	156	5143
	gain	lost	no-change	gain	lost	no-change	gain	lost	no-change

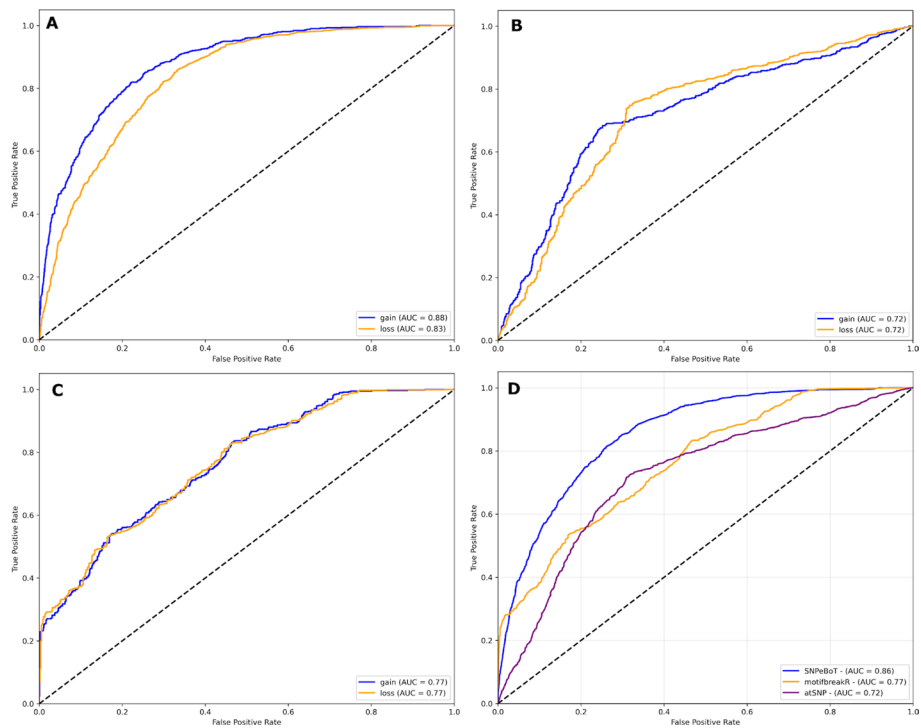


Fig. 6 Binary Receiver Operating Characteristic Curves for best scoring thresholds of all three tested tools. A) SNPeBoT B) atSNP C) motifbreakR D) The averaged curves all A-C displayed together

for each tool the results for the threshold at which it was most accurate. When these were compared, SNPeBoT yielded both the highest accuracy as well as ASB recall rate (Table 1). On a binary basis, considering only predictions of gain and loss, SNPeBoT has a higher Area Under the Curve than either atSNP or motifbreakR both on an individual class basis and when averaged across classes (Fig. 6). We believe that there are several possible explanations for the improvement of our model upon the baseline models compared against. These models represent the state of the art approach for predicting SNP effects on TF binding which employ PWMs. PWMs themselves may be calculated from binding sequences measured from in vivo experiments or from in vitro experiments such as PBM. When a PWM is measured from these experiments the nucleotide relevance at each position is calculated while assuming statistical independence from all other positions within the binding site. This will lead to

loss of information concerning TF binding effect linkage of nucleotides at varying positions. We believe that by applying a deep learning method directly to the Raw binding data, (E-scores), SNPeBoT is able to rescue some of this information which is lost during PWM calculation, and therefore sees an improvement in the accuracy of prediction. Secondly, SNPeBoT applies a rudimentary method for selecting the motif whose PBM data is to be used for prediction. Since each TF can have multiple different motifs, tools that do not discriminate between each, such as the baseline models measured, may be introducing noise into their predictions. Should the user desire to investigate any given ASB binding prediction further, we have provided instruction in the Supplementary Information on how they might do so. We believe that the greater coverage of ASB events predicted by SNPeBoT with small gains in accuracy may be a significant improvement in the field. For example, SNPeBoT would be notably useful in applications where an emphasis is placed on the large number of SNPs retrieved and the predictions can be supplemented with additional experimental data controlling for false positive predictions. Additionally, the availability of SNPeBoT in a web server provides an ease of use facilitating quick filtration of potential ASB that may then be followed up with additional experimental confirmation.

Generalizability

Finally, we wanted to test SNPeBoT's predictive ability when applied to a novel data type. SNPeBoT achieved a prediction accuracy on eY1H data of 70% with an ASB Recall of 56%, when considering only cases where the model predicted ASB (gain or loss) it achieved 88% accuracy. This decrease in performance for SNPeBoT when predicting eY1H as compared to ChIP-seq data was unexpected as off-site effects such as chromatin accessibility and protein–protein interactions do not contribute to TF binding in these experiments. However, low accuracy for this test was caused by a large number of false negatives. It is possible that SNPeBoT's training on ChIP-seq data and overcompensation for negatives therein is causing this heavy overprediction of negatives in the in vitro data set which contained only SNPs resulting in change of TF binding.

Implications and potential extensions

We believe that SNPeBoT opens the door to further development by incorporating information important to TF binding other than the target sequence such as the sequence environment and protein interaction landscape [22]. Furthermore, while tools such as SNPeBoT relying on data from PBM, PWM, ChIP-seq, or Systematic Evolution of Ligands by Exponential enrichment (SELEX) information for training or use are limited by data availability, recently our lab has shown with ModCRE that statistical potentials calculated from modeled TF-DNA interactions are valid approximations for PBM derived E-scores [23]. Thereby we intend to apply the methods of SNPeBoT to ModCRE where information of Protein interaction can be incorporated and the reliance on experimental data for the generation of predictions can be shed. Finally, SNPeBoT provides a perfect opportunity to test the significance of GWAS associations to disease etiology.

Conclusion

SNPeBoT improves upon the predictive accuracy of existing tools analysing the effects of SNPs on TF binding when applied to TF with PBM data. The increase in Allele specific Binding events recovered may yield insight into previously undiscovered regulatory mechanisms underlying human disease. Furthermore, the ease of implementation ensures that SNPeBoT is usable to as many researchers as may be interested.

Availability and requirements

Project name

SNPeBoT.

Project home page

Webserver: (<https://snpebot.upf.edu/>), Standalone: (<https://github.com/structuralbioinformatics/SNPeBoT>).

Operating system(s)

Platform independent.

Programming language

Python3, shell, html.

Other requirements

(Standalone) SBILib, fimo (MEME package), dssp, genopyc 2.6.3, tensorflow 2.13.0, matplotlib 3.8.2, jsonpickle 3.0.2.

License

MIT.

Any restrictions to use by non-academics

None.

Abbreviations

ASB	Allele specific binding
AUC	Area under the curve
ChIP-seq	Chromatin immunoprecipitation sequencing
CISBP	Catalog of inferred sequence binding preferences
CNN	Convolutional neural network
eY1H	Enhanced yeast 1 hybrid
FIMO	Find individual motif occurrences
GWAS	Genome wide association study
PBM	Protein binding microarray
PWM	Position weight matrix
ROC	Receiver operating characteristic
RSID	Reference single nucleotide polymorphism ID
SELEX	Systematic evolution of ligands by exponential enrichment
SNP	Single nucleotide polymorphism
SNPeBoT	Single nucleotide polymorphism effect on binding of transcription factors
TF-DBD	Transcription factor DNA binding domain

TF Transcription factor

Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s12859-025-06094-4>.

Additional file 1 (DOCX 330 KB)

Acknowledgements

We would like to thank all members of the SBI lab that advised in the design and development of this project and the system administrators of GRIB and MELIS.

Author contributions

PG: conducted data preprocessing, CNN Training and Testing, data Analysis and manuscript writing. B.O. conducted the project supervision and planning and manuscript writing and editing. All authors reviewed the manuscript.

Funding

The work was supported by grants PID2020-113203RB-I00, PID2023-150068OB-I00 and “Unidad de Excelencia María de Maeztu” (ref: CEX2018-000792-M), funded by the MCIN and the AEI <https://doi.org/10.13039/501100011033>, MCIUN/AEI/10.13039/501100011033/FEDER, UE as well as an FPU scholarship (ref: FPU22/02303) and an SGR from the Generalitat de Catalunya (ref: 4413015318- J.SELENT/SGR-22).

Availability of data and materials

The datasets analysed during the current study are available in the AdAstra repository, <https://adastra.autosome.org/mabel/downloads>, processed data and source code are available in the Github repository (<https://github.com/structuralbioinformatics/SNPeBot>).

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Conflict Of Interest

The authors declare no competing interests.

Received: 24 December 2024 Accepted: 21 February 2025

Published online: 10 March 2025

References

1. Lettice LA, Williamson I, Devenney PS, Kilanowski F, Dorin J, Hill RE. Development of five digits is controlled by a bipartite long-range cis-regulator. *Development*. 2014;141(8):1715–25. <https://doi.org/10.1242/dev.095430>.
2. Engeland K. Cell cycle regulation: p53–p21–RB signaling. *Cell Death Differ*. 2022;29(5):946–60. <https://doi.org/10.1038/s41418-022-00988-z>.
3. Gertz J, Savic D, Varley KE, Partridge EC, Safi A, Jain P, Cooper GM, Reddy TE, Crawford GE, Myers RM. Distinct properties of cell-type-specific and shared transcription factor binding sites. *Mol Cell*. 2013;52(1):25–36. <https://doi.org/10.1016/j.molcel.2013.08.037>.
4. Cavalli M, Pan G, Nord H, Wallerman O, Wallén Artzt E, Berggren O, Elvén ML, Rönneblom L, Lindblad Toh K, Wadelius C. Allele-specific transcription factor binding to common and rare variants associated with disease and gene expression. *Hum Genet*. 2016;135(5):485–97. <https://doi.org/10.1007/s00439-016-1654-x>.
5. Rheinbay E, Nielsen MM, Abascal F, Wala JA, Shapira O, Tiao G, Hornshøj H, Hess JM, Juul RL, Lin Z, Feuerbach L. Analyses of non-coding somatic drivers in 2,658 cancer whole genomes. *Nature*. 2020;578(7793):102–11.
6. Coetzee SG, Coetzee GA, Hazelett DJ. motifbreakR: an R/Bioconductor package for predicting variant effects at transcription factor binding sites. *Bioinformatics*. 2015;31(23):3847–9. <https://doi.org/10.1093/bioinformatics/btv470>.
7. Zuo C, Shin S, Keleş S. atSNP: transcription factor binding affinity testing for regulatory SNP detection. *Bioinformatics*. 2015;31(20):3353–5. <https://doi.org/10.1093/bioinformatics/btv328>.
8. Nishizaki SS, Ng N, Dong S, Porter RS, Morterud C, Williams C, Asman C, Switzenberg JA, Boyle AP. Predicting the effects of SNPs on transcription factor binding affinity. *Bioinformatics*. 2020;36(2):364–72. <https://doi.org/10.1093/bioinformatics/btz612>.
9. Zhou Z, Ji Y, Li W, Dutta P, Davuluri R, & Liu H. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. (2023) arXiv preprint [arXiv:2306.15006](https://arxiv.org/abs/2306.15006).
10. Grant CE, Bailey TL, Noble WS. FIMO: scanning for occurrences of a given motif. *Bioinformatics*. 2011;27(7):1017–8. <https://doi.org/10.1093/bioinformatics/btr064>.
11. Weirauch MT, Yang A, Albu M, Cote AG, Montenegro-Montero A, Drewe P, Najafabadi HS, et al. Determination and inference of eukaryotic transcription factor sequence specificity. *Cell*. 2014;158(6):1431–43.

12. Abramov S, Boytsov A, Bykova D, Penzar DD, Yevshin I, Kolmykov SK, Fridman MV, Favorov AV, Vorontsov IE, Baulin E, Kolpakov F, Makeev VJ, Kulakovskiy IV. Landscape of allele-specific transcription factor binding in the human genome. *Nat Commun*. 2021;12(1):2751. <https://doi.org/10.1038/s41467-021-23007-0>.
13. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. *J Mol Biol*. 1990;215(3):403–10. [https://doi.org/10.1016/S0022-2836\(05\)80360-2](https://doi.org/10.1016/S0022-2836(05)80360-2).
14. Gohl P, Bonet J, Fornes O, Planas-Iglesias J, Fernandez-Fuentes N, Oliva B. SBLib: a handle for protein modeling and engineering. *Bioinformatics*. 2023. <https://doi.org/10.1093/bioinformatics/btad613>.
15. Raney BJ, Barber GP, Benet-Pagès A, Casper J, Clawson H, Cline MS, Diekhans M, et al. The UCSC genome browser database: 2024 update. *Nucleic Acids Res*. 2024;52:D1082–8.
16. Gualdi F, Oliva B, Piñero J. Genopyc: a python library for investigating the functional effects of genomic variants associated to complex diseases. *Bioinformatics*. 2024. <https://doi.org/10.1093/bioinformatics/btae379>.
17. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, Kudlur M, Levenberg J, Monga R, Moore S, Murray DG, Steiner B, Tucker P, Vasudevan V, Warden P, Wicke M, Yu Y, and Zheng X. TensorFlow: A system for Large-Scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283, Savannah, GA, November 2016a. USENIX Association. ISBN 978–1- 931971–33–1.
18. Chollet F, & others. (2015). Keras. GitHub. Retrieved from <https://github.com/fchollet/keras>
19. Fuxman Bass JI, Sahni N, Shrestha S, Garcia-Gonzalez A, Mori A, Bhat N, Yi S, Hill DE, Vidal M, Walhout AJM. Human gene-centered transcription factor networks for enhancers and disease variants. *Cell*. 2015;161(3):661–73. <https://doi.org/10.1016/j.cell.2015.03.003>.
20. Chen J, Rozowsky J, Galeev TR, Harmanici A, Kitchen R, Bedford J, Abyzov A, Kong Y, Regan L, Gerstein M. A uniform survey of allele-specific binding and expression over 1000-Genomes-Project individuals. *Nat Commun*. 2016. <https://doi.org/10.1038/ncomms11101>.
21. Storey JD, Bass AJ, Dabney A, Robinson D (2023). qvalue: Q-value estimation for false discovery rate control. <https://doi.org/10.18129/B9.bioc.qvalue>, R package version 2.34.0, <https://bioconductor.org/packages/qvalue>.
22. Deplancke B, Alperin D, Gardeux V. The genetics of transcription factor DNA binding variation. *Cell*. 2016;166(3):538–54. <https://doi.org/10.1016/j.cell.2016.07.012>.
23. Fornes O, Meseguer A, Aguirre-Plans J, Gohl P, Bota PM, Molina-Fernández R, Bonet J, et al. Structure-based learning to predict and model protein–DNA interactions and transcription-factor co-operativity in cis-regulatory elements. *NAR Genom Bioinform*. 2024;6(2):lqae068.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.