Promoter Motif And Conserved Module Analysis In DNA And RNA Using Almodules

Muharrem Aydinli

University of Würzburg

Chunguang Liang

University of Würzburg

Thomas Dandekar (✉ dandekar@biozentrum.uni-wuerzburg.de)

University of Würzburg

Research Article

Keywords: Motif, TF, TFBS, Transcription factor, Matrix, Poly(A)-tail, Modules

Posted Date: January 27th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1284479/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License. [Read Full License](#)

Abstract

Identification of transcription factor binding sites (TFBSs) elucidates regulation and control of transcription and gene expression. Overprediction of TFBS due to the high abundance of the small binding motifs is a notorious problem. For this comparative evaluation of TFBS is important.

The tool AModules presented in this paper presents an integrated solution for TFBS analysis as Web service and downloadable software. It works for other DNA and RNA motifs as well.

Several sequences (DNA and RNA) are searched for TFBSs using predefined matrices from the JASPAR DB or by using own matrices for DNA or RNA motif discovery. Furthermore, AModules can find TFBS common to two or more sequences. Demanding high or low conservation AModules outperforms other solutions in speed and finds more modules (specific combinations of TFBS) than alternatives. The web application searches also RNA motifs (e.g. polyadenylation motif) as well as other DNA motifs (e.g. enhancers, silencers) and motif combinations. The application is free and open source to be used on-site or locally.

These functionalities allow a number of biological applications, in particular analysis of promoters, including conserved modules and shared TFBSs in conserved gene families, presence of modules, motifs or regulatory elements in mRNA.

Introduction

During transcription of DNA to RNA the transcription preinitiation complex is build up by transcription factors, that bind specific regions of the DNA. These regions are called the transcription factor binding sites and on the one side make up the promoter itself and on the other are conserved throughout species as well as gene families. E. g., TFs make up about 8% of all human genes (1). Due to mutation the sequences of TFBSs differ in various species and hence to search for the equivalent transcription factors (TFs) among them, techniques were developed to estimate with high confidence not only the corresponding TFs but also coupled TFs, which are called modules. For that, databases such as GenBank, ProDoric, MotifMap, etc. were developed as well as TRANSFAC, which offers a publicly free but outdated version from the year 2005, or the up-to-date JASPAR(3) database, which uses high quality matrices generated from SELEX, protein binding microarray (PBM), ChIP-based assays, etc.(1) (see Supplementary Table S9 for more on matrix generation). Via these techniques TFs could be predicted with high confidence. However, due to not knowing the TF-TF protein interactions in the preinitiation complex or other details during the DNA annealing of the TF, hence only relying on the one DNA motif for predicting a TFBS, false positives were generated. That's why it is important to search for conserved TFBSs (transcription factor binding sites) and consider cell type specific as well as functional modules. Hence, a useful software "Genomatix" was created, which did exactly that (TRANSFAC, as another tool for module discovery, is not part of this paper and mentioned in the Supplementary Section "TRANSFAC"). However, the tool was commercialized and an open source alternative is currently not available. Thus, we present a free and open source tool which fills this gap and we present the different biological uses of it (e.g. motif and module search). In addition, this allowed to generalize the tool so that also other DNA motifs can be searched (silencers, enhancers) as well as more complex combinations of such. Moreover, also RNA motifs and conserved bindings sites in RNA can be efficiently searched. This is demonstrated for a poly(A)-tail search in RNA.

Results

Architecture

We used a three-layered architecture (Figure 1): front-end, back-end, database. The searches for TFBSs are done on the backend, whereas the module search as well as the rendering of the result in the front-end. Furthermore, we prepared a Docker based solution (Supplementary section "Build and Deploy").

Remark to editor: Insert Figure 1 here!

Motif searches

At the time of writing this manuscript only two commercial products (Genomatix database tool and TRANSFAC database) were available, which could find modules (coupled motifs) on a stretch of DNA. Therefore, we sought to make available an open source tool that allows the user to insert own DNA stretches as well as own matrices, but also provide a comprehensive collection of matrices from Jaspar. The tool is a web service and therefore allows for easy including as well as applying new features. Motif search is done on the server-side, whereas module discovery on the client to decrease the load on the server.

The User Interface is designed to support the user, so that a quick start is possible (see Supplementary section “User Interface”). After module calculation, the result for the module search is depicted visually and can be downloaded as an Excel file as well (Figure 2 and for details see Supplementary Fig. S 6).

AIModules offers three types of searches: TFBS, RNA motif and module search.

TFBS search

The most basic search AIModules offers is the search for TFBSs. For that the user inserts DNA sequences for analysis, select the thresholds (L_a and L_d) and the matrices. These can either be selected from the database or individual ones can be inserted. Matrices for AIModules can also be generated in the tool itself. An example on the TFBS search is depicted in Figure 2a.

RNA Motif search

Furthermore, our tool AIModules also allows for the analysis of RNA motifs. Therefore, the RNA sequence of choice can be analyzed by the user. Uracils are internally converted to thymine. After selecting the parameters L_a and L_d as well as selecting the matrix, the result is obtained. L_a is the actual log-odds ratio of the match, whereas L_m is the maximum possible log-odds ratio for a match, i.e. the information content of the consensus sequence. L_d on the other hand is the maximum log likelihood deficit ($L_a - L_d$) – or put simply L_a can be understood as the lower and L_d as the top score threshold between which a TFBS is valid. The score is calculated via position weight matrices (PWMs) (5) (7). For matrices the user may select Jaspar matrices from a drop down menu within the web application or insert own ones. We included not TRANSFAC database as this is commercial, however, this alternative can of course easily be uploaded in older public versions (8). To illustrate the capabilities for RNA searches, a suitable RNA motif search matrix of poly(A) motif sequences from (9) was generated in AIModules itself and used for the analysis. A result of a Poly-(A) search is depicted in Figure 2b. The result can also be downloaded as an Excel table (see Table 1).

Table 1

Results of poly(A)-tail search. AlModules allows to search for Poly-(A) tails. Generally, with AlModules one may search for functional RNA sites as well.

Sequence Name	Sequence Length	Matrix Length	Hit Sense	Hit Start	Hit Stop	Hit Score (La)	Hit Max log likelihood ratio score (Lm)	Difference (Ld) (maxscore(Lm) - score(La))	Hit Oligo
NM_000600.5 Homo sapiens interleukin 6 (IL6), transcript variant 1, mRNA	1127	6	N	1025	1030	9.889837	9.889837	0.000000	AATAAA
			N	1103	1108	9.889837	9.889837	0.000000	AATAAA
			R	842	847	9.889837	9.889837	0.000000	TTTATT
NM_000594.4 Homo sapiens tumor necrosis factor (TNF), mRNA	1678	6	N	1655	1660	9.889837	9.889837	0.000000	AATAAA
			R	928	933	9.889837	9.889837	0.000000	TTTATT
			R	1345	1350	9.889837	9.889837	0.000000	TTTATT
			R	1352	1357	9.889837	9.889837	0.000000	TTTATT
			R	1356	1361	9.889837	9.889837	0.000000	TTTATT
			R	1363	1368	9.889837	9.889837	0.000000	TTTATT
			R	1367	1372	9.889837	9.889837	0.000000	TTTATT
			R	1387	1392	9.889837	9.889837	0.000000	TTTATT
NM_020525.5 Homo sapiens interleukin 22 (IL22), mRNA	1165	6	N	1144	1149	9.889837	9.889837	0.000000	AATAAA
			R	824	829	9.889837	9.889837	0.000000	TTTATT
			R	873	878	9.889837	9.889837	0.000000	TTTATT
			R	991	996	9.889837	9.889837	0.000000	TTTATT
			R	1015	1020	9.889837	9.889837	0.000000	TTTATT
			R	1029	1034	9.889837	9.889837	0.000000	TTTATT
			R	1043	1048	9.889837	9.889837	0.000000	TTTATT
			R	1131	1136	9.889837	9.889837	0.000000	TTTATT

Remark to editor: Insert Table 1 here!

The table is truncated. The full list of analyzed mRNAs can be found in Supplementary section “Used sequences” and the complete table of poly-(A) results in Supplementary Table S 12.

Module Search

Moreover, we implemented the module search. For that, the user may insert own sequences, select the parameters for La and Ld, and activate the checkbox for module filtering. The conservation of the TF can be selected by the user using a stepper menu: The user then may select either Jaspar matrices or insert own matrices as well. Results from module searches are shown in Figure 2c and Figure 2d.

Remark to editor: Insert Figure 2 here!

The promoter sequences we used can be found in Supplementary section “Used sequences”. Furthermore, found TFBSs as well as modules can be downloaded as Excel files from the web application. Moreover, the user may select and deselect TFs from the result in order to manually search for modules (see Supplementary Fig. S 2).

Performance comparison to other tools (TFBS search)

We compared our tool to the web application conTraV3 (10). Whereas the conTraV3 solution was in production state, we tested our tool in a Virtualbox virtual machine with Debian, 4.5 GB of RAM and two CPUs (Intel i7-3520m 2 x 2,9GHz). Although run locally, our tool performed well compared to conTraV3. We analyzed one sequence (AJ223836.1 *Chionodraco hamatus* mRNA for cathepsin; length 1332 bp; for sequence see Supplementary section “conTraV3 comparison”). For conTraV3 we selected one matrix (PWM1id: TF1name: HOME0) with default thresholds (Stringency: core = 0.95, similarity matrix = 0.85) and in case of AIModules we selected all 1920 matrices with default thresholds (La=6, Ld=8) as well. Our solution took about one second to receive the result from the backend and ca. two seconds to render the found TFBSs into the result. On the other side, we cancelled the analysis from the tool conTraV3 after one hour without a result.

Comparison to commercial products (TFBS and Module search)

To analyze how well our solution performs, we took our results and compared our results to a commercial tool, Genomatix. Hence, we began to compare the results of both – *AIModules* and *Genomatix* (11). Genomatix is a commercially available tool with a one-week free trial period (available tools in the field of promotor analysis are found in the *Discussion* section). We analyzed and compared the results from AIModules and Genomatix by looking at an example set of selected genes. The homologous promoters are taken from GenBank and are in the first example cathepsins (Homo_sapiens_cathepsin_V_transcript_variant-1_promoter, Bos_taurus_cathepsin-Z_promoter, Mus_musculus_cathepsin-F-transcript-variant-X1_promoter_576-1076) and as second example IL-10 (X73536.1_H.sapiens_promoter_region_of_human_IL-10_gene, AY486432.1_Macaca-mulatta_interleukin-10-(IL-10)_gene_promoter_region, AF121965.1_Mus-musculus_interleukin-10-(IL10)_gene_promoter_partial_sequence). The sequences are depicted Supplementary section “Used sequences”.

The resulting modules from AIModules and Genomatix were compared by hand, as the modules from Genomatix were low in number. On the other side, namely the results from crude TFBS search, were handled differently. Due to the high number of the findings and the differences between the naming of the TFBSs from both systems, we chose a semi-automatic approach. Each result set was put into python arrays and the Genomatix result sets were copied unchanged into another array. The corresponding arrays from AIModules and Genomatix then were compared for string equality by a python script (see Supplements.zip). The matches were put in separate arrays with the syntax

AIModules_TFBS_name::Genomatix_TFBS_name, where *Genomatix_TFBS_name* can consist of multiple hits, which are separated by commas. The resulting arrays then were printed to *standard output* and refined manually in *libreoffice Spreadsheet*. We saw that AIModules found more TFBSs than Genomatix and that some motifs are common to both systems. Regarding modules, AIModules found much more (ten-fold) modules than Genomatix. The parameters used for both solutions and the statistics of the found TFBSs as well as modules are depicted in Supplementary Table S1 – S8.

The amount of found TFBSs from both methods differ in number. This is due to differences in available matrices and the setting of search parameters which are for AIModules La and Ld, and for Genomatix 0.75 and *Optimized*. As explained in methods, we give considerations how these parameters can be compared and where they differ. Regarding the module search the differences between the system parameters are similar to the ones for TFBSs. In AIModules the parameters are La, Ld and the activated checkbox for module search, whereas Genomatix uses a *Threshold for number of elements* and a *Maximum number of matches*.

As these parameters are difficult to directly compare and normalize to each other, the found matches have only a small overlap. Additionally, some of the TFBSs are specific to the used system.

We show that our tool is the only one, that can detect common modules within the analyzed sequences. Moreover, we combine this feature with a TFBS search as well as RNA motif discovery. Our tool allows the user to insert not only own sequences but also own matrices.

Each of the motif discovery tools mentioned in Table 2 and Table 3 are the results of beautiful work and they have their own use cases. For specific uses however, our tool has shown to find much more modules than even a commercial product. We have a detailed comparison of the tools from Table 2 and Table 3 in the Supplementary section “An Overview of the tools for TFBS discovery”.

Table 2
Comparison of promoter analysis tools.

Name	Search for TFBSs	Saved in Databases	Precalculated results	Insert own matrices	Insert own sequences	Show common modules of own sequences
<i>AIModules</i>	Yes	Matrices	No	Yes	Yes	Yes
<i>ContraV3</i> (10)	Yes	Genomes and Matrices	No	Yes	Yes	No
<i>TRANSFAC</i> (25) (26) (free version and commercial version)	Yes	Genomes, promoters, matrices and modules	No	Yes	Yes	No; modules are calculated for each sequence
<i>Genomatix</i> (11) (commercial)	Yes	Genomes, promoters, matrices and modules (vertebrates and plants)	No	Yes	Yes	No; for each sequence own modules are depicted
<i>Motifmap</i> (27) (28) (29)	Yes	Genomes and Matrices	No	No	No	No
<i>Promo</i> (30) (31)	Yes	Matrices	No	No	Yes	No, only common TFBSs
<i>ModuleMaster</i> (13)	Could not be run under Linux Mint 64bit and Windows 10 64bit. Not supported anymore, as the originating Lab does not do bioinformatics anymore.					
<i>Prodoric</i> (32) (33)	Yes	Genomes and Matrices	Yes	No	Yes, but only one	No

Table 3
Comparison of promotor analysis tools: Further tools

<i>Softberry</i> (34) (35) (36)	Yes	Matrices	No	No	Yes	No; only one sequence can be input
<i>TAIR</i> (37)	Yes; if TFBS is common to at least 3 sequences	Matrices	No	No	Yes	No; shows TFBSs common to at least 3 sequences; only for some plants; only for 6-mer TFBSs
<i>PlantPan 3.0</i> (38)	Yes	Genomes	No	No; but consensus sequence as IUPAC code	Yes	No; only common TFBSs
<i>CisBP</i> (21; 39)	Yes	Genomes and Matrices	No	No	Yes	No
<i>UniPROBE</i> (22; 40)	Yes	Matrices	No	No	Yes	No
<i>HOCOMOCO</i> (41; 42)	Yes	Matrices	No	No	Yes	No
<i>FlyFactorSurvey</i> (43)	Yes	Genomes and Matrices	No	No	No	No
<i>MEME Suite</i> (44; 45)	Yes	Matrices	No	Yes	Yes	No
<i>YeTFaSCo</i> (46; 47)	Yes	Matrices	No	No	Yes	No
<i>TESS</i> (5) (6; 7)	Not available any more as a webservice. The code of parts of the backend is available.					

Discussion

AIModules allows efficient module searches on more than one sequence and filters the common modules to render them beautifully on the website. The application is enriched by TFBS searches and RNA motif discovery as well. This is possible due to our general approach algorithm. Therefore, we believe that this is a powerful package to find modules *in silico*. We have prepared the application in a way, that allows for extensions without much effort, not only due to the architecture but also due to the chosen free and open-source licensing agreement (GPLv2). Moreover, the application is provided on our own server so that the user does not have to use complicated scripts or even commercial software.

Modules analysis follows in AIModules a strict algorithm to try and find shared modules between the input sequences. For that the TFBSs have to have a match in N input sequences to be valid and hence be included in the module search. The number N can be defined by the user via a stepper control and defines the conservation of the TF. The strand orientation for TFBSs is relevant in this step. A module consists of two TFBSs with a fixed offset of ± 200 bp. Every permutation of every TFBS is tested for validity (AB, AC, AD, ..., BC, BD, ...). The module is a valid one, when it is shared at least between two input sequences. In Genomatix all sequences are analyzed for known modules independently. I.e., that Genomatix does not show common modules. These differences in the module finding process lead to different amounts of found modules. Where AIModules finds all possible modules algorithmically, Genomatix relies on known co-citations. This means that AIModules may over represent modules, whereas Genomatix only shows co-citations and may miss modules, that are included in AIModules. However, the number of found modules can be refined in AIModules by increasing L_a or decreasing L_d or only using user input matrices, or a combination of the aforementioned points. The analyzed sequences for cathepsin and IL-10 showed no overlap in AIModules and Genomatix regarding modules. For each of these systems *in silico* search will not make experimental validation obsolete.

The other commercial product *TRANSFAC* is available as a free version after registration. However, the matrices are from 2005, hence outdated, and limited in number (398 matrices). Furthermore, this free version is functionally constrained (12) and the professional version is only available after licensing. *AlModules* offers 1920 matrices from the Jaspar DB, whereas the public version of *TRANSFAC* contains only 398. Therefore, we decided for the more up-to-date and sensitive matrices from the Jaspar DB, which also provides a REST-API and the JASPAR R/Bioconductor package (3). Moreover, the matrices from *Transfac* cannot be downloaded, but have to be extracted from the website manually, which is time consuming as well as error-prone. Since the application is open source, *Transfac* matrices can be added when needed.

Compared to those two most popular databases (*Genomatix* and *Transfac* public), *AlModules* offers the possibility to find much more patterns. Meanwhile *AlModules* can search multiple sequences and obtain comprehensive visualization and statistic results. Moreover, *AlModules* allows the user to deselect and select each of the found TFBSs and assemble TF patterns manually (for more see Supplementary Fig. S 2).

By the time this manuscript is assembled only two products were on the market that could predict modules. These tools are *Genomatix* (from Intrexon Bioinformatics Germany GmbH) as well as *TRANSFAC*. *ModuleMaster* (13) is another tool that could predict modules, but we were unable to start the WebStart Application on different operating systems. The originating lab could not provide any assistance as the Bioinformatics research is no more part of it. Furthermore, for the end user it is easier to use a website than a Java WebStart application that is firstly not up to date and secondly shipped without a valid certificate which may pose a security hazard. All the other tools in Table 2 and Table 3 had no common module search functionality, but they are the result of beautiful work and have their own uses. Moreover, *AlModules* is not only available as a web application, but can be deployed as an on-site application or locally on a PC or notebook as well.

Remark to editor: Insert Table 2 here!

Remark to editor: Insert Table 3 here!

Table 2 and Table 3 are discussed in more detail in Supplementary section “An Overview of the tools for TFBS discovery”.

Furthermore, for TFBS analyses R packages from Bioconductor (14) are also available (e.g. *TFBSTools* (15), *RcisTarget* (16), *enrichTF* (17)). These, however, must be packaged into new code to be able to be used on sequences for TFBS identification, in particular if you want to determine conserved TFBS between different promotor regions and DNA sequences or for establishing a web server and visualization of TFBS found.

Compared to the tool *conTraV3* we have shown that our tool outperforms it as *AlModules* needs for the same sequence with more matrices to analyze only seconds, whereas the analysis from *conTraV3* was cancelled after one hour without results. Additionally, we have shown that *AlModules* is able to detect polyadenylation sites (see Table 1) which were previously described in (9). Our tool *AlModules* is not only faster but also presents features such as module search and RNA motif discovery, whereas the sequences as well as the matrices can be inserted by the user individually. An overview of tools and features can be found in Table 2 and Table 3.

The resulting picture (see Figure 2 or Supplementary Fig. S 5) shows that binding site matches frequently overlap. These matches are filtered beforehand by the backend via the parameters *La* and *Ld*, and therefore strong bindings and high score matches are shown. However, it has to be considered that these results mean, that a TF would bind the binding site *in vitro*, but this has to be validated through experimentation. I.e., that even if the TF binds *in vitro* this must not mean that this TF plays a role in gene regulation *in vivo*.

The other side of the coin is, that if a TFBS is not shown in the result of an input sequence this must not mean that there is none. It could mean that the match was excluded by a high *La* or low *Ld*. Furthermore, the JASPAR database of matrices, while being more current and more thorough than e.g. the public version of *TRANSFAC*, is not exhaustive and is enriched and

optimized over time. I.e., that the TF may not be available and therefore may not be included in the result. These restrictions also apply to commercial products.

Predictions should be treated as such. A match means that in vitro the corresponding TF is very likely to bind the TFBS. In vivo there are factors like interactions of the TF with chromatin (conformation) which play a crucial role. Furthermore, the quantity of available TF relative to its TFBS and the quantity of cofactors contribute greatly to TF-TFBS interactions.

The endeavor of trying and involving machine learning should be assessed. Since there is an abundance of data it should be possible to train a model to recognize different TFBSs and modules in promoters. Important work on that has already been done using deep learning(18) but this approach can be improved. This could be done by machine learning frameworks such as Tensor Flow or PyTorch, etc. Moreover, models based on hidden Markov models such as Transcription Factor Flexible Models (TFFMs), that can model positional interdependence within the TFBSs and variable length motifs (19; 20), can be included. The high quality matrices from JASPAR can also be enriched with matrices from CisBP(21) or UniPROBE(22) as well.

Furthermore, the filtering for modules can be improved in quality by saving co-cited TFBSs (modules) in a fast database with its' offsets and applying these to the found TFBSs.

Since the architecture is also available as a Docker based solution (and a *docker swarm*), it should not be difficult to deploy this system onto a Kubernetes provider with high throughput. In this environment add-ons should be considered to be deployed as micro services or server-less components to increase the capacity for load balancing and failover functionality.

We present AIModules as a stand-alone promotor analysis tool. The user does not have to use complex scripts or commercial tools. Our solution provides a general approach to analyse protein factors binding to DNA as well as RNA. We do not only predict TFBSs on DNA but the tool can also be applied to proteins recognizing RNA motifs and to analyze RNA stretches. It is also a means to search for modules (conserved TFBS combinations) and thus fills the gap that arose regarding academic software when Genomatix was commercialized. As the only free and open source tool that can predict common modules on two or more sequences we also include matrices of high quality by the use of the JASPAR 2022 DB. This database is more recent than e.g. the public TRANSFAC version, which is from the year 2005 (8). Moreover, within the web application we include a functionality to calculate matrices to be used in AIModules, so that any user can compile own matrices. Additionally, we included a matrix for the search of poly(A)-tails which again can be extended by further RNA motifs. We show, that our solution finds more TFBSs, and a lot more modules than public available alternatives. Furthermore, our application is much faster than the solution conTraV3 (few seconds vs at least 1 hour). When searching for modules the degree of conservation of TFBSs can be determined by the user so that less conserved modules can be found as well as individually conserved TFBSs. Moreover, found TFBSs can be selected manually to compile custom modules manually. With our solution we not only provide a free service that can be deployed in a containerized environment for load balanced high availability environments but also provide a framework for TF and module searches, that can be extended easily due to the chosen design as well.

Conclusions

AIModules is a nice and versatile software package which allows to identify transcription factor binding sites (TFBSs) as well as combinations of them comparing their conservation in several genes. The software performs better than current alternatives and is generic, i.e. it can also be applied to look for other motifs in DNA such as enhancers or silencers or regulatory motifs in RNA. The user can also look into conserved motifs of his own choice. We provide both a web-server as well as the stand-alone software for installation. We have thus a particular flexible and easy to use solution for the interested researcher. Most notably, it is completely free, non-commercial and open source – filling here a gap for such a free solution in the academic community.

AIModules functionalities allow a number of biological applications, in particular analysis of promoters, including conserved modules and shared TFBSs in conserved gene families, presence of modules, motifs or regulatory elements in mRNA.

Methods

Architecture and parameters:

The backend calls *tessWms* to search for TFBSs with the parameters selected in the front-end. These are *La* and *Ld*. *La* is the log odds ratio of the match from a PWM, whereas *Ld* is the maximum log likelihood deficit, i.e. the difference between the maximum ratio score of a PWM (*Lm*), which is the consensus sequence, and the log odds ratio of the match (*Lm-La*) (5) (7). Each position in a binding site can contribute up to the value of two to the score. Thus, the best *La* corresponds to the consensus sequence and hence this means that *Ld* defines how much worse the *La* of a TFBS is compared to *Lm*.

The parameters for *MatInspector* in the tool *Genomatix* on the other hand are *Optimized* and 0.75. A perfect match to the matrix means, that the binding site is equal to the consensus sequence and hence gets a score of 1.00. *Optimized* in the context of matrix similarity for *Genomatix* means, that the binding site is valid if the score is greater than 0,75 (23).

Implementation:

Our tool *AIModules* (3) uses position frequency matrices (PFMs), introduced in 1982 (4), to predict TFBSs either from the Jaspar 2022 Database or user input PFMs. We decided against including Transfac matrices, as these are from 2005, hence old, and the 398 matrices cannot be downloaded but have to be extracted manually from their website one by one from HTML, which is time consuming and error-prone. The result of the analyses is presented to the user in graphical and file form. Apart from a conservative three layered architecture (see Figure 1), *AIModules* is also implemented and prepared as a Docker container based solution (see Supplementary section “Build and Deploy”).

The architecture of *AIModules* is the product of three layers which are loosely coupled. For the view we chose the single page application framework (SPA) *angularJS* to reduce calls to the backend. The SPA is offered on an apache webserver. The frontend communicates with the backend via *JSON*, the backend itself is a JAXRS Rest service running on apache tomcat 9.0.0.M1 and *JavaEE*. In this layer the executable *tessWMS* (5; 6; 7) is called to find TFBSs. We modified *tessWms* to allow for a JSON interface, hence communication from *tessWMS* to the Java backend is done via JSON. The user can select TFBS classes in the frontend. Those are read from the REST backend, which communicates with the third layer – a postgres 11 DB with the Jaspar2022 TFBSs – and presents them via specific URLs (see Supplementary section “Build and Deploy”).

Furthermore, we offer a completely docker packaged environment (see Supplementary section “Build and Deploy”).

Moreover, all mentioned databases and tools of the paper are described in Supplementary section “Development”.

Abbreviations

TF: Transcription factor, TFBS: Transcription factor binding site

Declarations

Acknowledgements

Not applicable.

Author contributions

MA developed the web application in its three-layered form and as a Docker based solution. MA generated the results. CL provided software expertise and checked the software. MA wrote the manuscript with the assistance of TD. All authors read and approved the final version of the manuscript.

Data availability statement

All data and materials are fully available from the paper and its supplementary materials. The program sources are available via <https://github.com/muharrem-aydinli/AIModules.git> or <https://zenodo.org/badge/latest/doi/363702392>. Sources are available and uploaded as soon as the paper is accepted. Further project and software information:

Project name: AIModules

Project home page: <https://bioinfo-wuerz.de/aimodules/>

Operating system(s): Platform independent, Web application

Programming language: Java, Javascript, C, Python

Other requirements: docker, Java 1.8, Tomcat, postgres, or use our web application with a web browser

License: GNU GPL v2

Any restrictions to use by non-academics: none

Competing interests

The authors declare that they have no competing interests.

Funding

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project number 3 74031971 – TRR 240/INF

Author information

1 Biocenter, Department of Bioinformatics, Am Hubland, 97074 University of Würzburg, Germany

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

References

1. Lambert, S. A. et al. The Human Transcription Factors. *Cell*. Feb 8, 2018, 172(4), pp. 650-665. Erratum in: *Cell*. 2018 Oct 4;175(2):598-599. PMID: 29425488.
2. Fornes, O. et al. JASPAR 2020: update of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*. Volume 48, Januar 8, 2020, Vol. Issue D1, pp. D87–D92.
3. AIModules. [Online] [Cited: 10 31, 2021.] <https://bioinfo-wuerz.de/aimodules/>.

4. Stormo, G. D., Schneider, T. D., Gold, L., Ehrenfeucht, A. Use of the 'Perceptron' algorithm to distinguish translational initiation sites in *E. coli*. *Nucleic Acids Research*. 1982, 10(9), pp. 2997–3011.
5. Schug, J. Using TESS to predict transcription factor binding sites in DNA sequence. *Curr Protoc Bioinformatics*. Chapter 2, 2008, Vol. Unit 2.6.
6. Schug, Jo. TESS. *Transcription Element Search System*. [Online] Computational Biology and Informatics Laboratory. [Cited: 07 30, 2021.] <https://www.cbil.upenn.edu/tess>.
7. Overton, G. C., Schug, J. TESS: Transcription Element Search Software on the WWW. [ed.] School of Medicine, University of Pennsylvania Laboratory. 02 16, 1998.
8. Public Transfac Database. [Online] [Cited: 10 22, 2020.] <http://gene-regulation.com/pub/databases.html>.
9. Xie, B., Jankovic, B. R., Bajic, V.B., Song, L., Gao, X. Poly(A) motif prediction using spectral latent features from human DNA sequences. *Bioinformatics*. 7 1, 2013, 29(13), pp. i316-25.
10. Kreft, L. et al. ConTra v3: a tool to identify transcription factor binding sites across species, update 2017. *Nucleic Acids Research*. Volume 45, July 3, 2017, Vol. Issue W1, pp. W490–W494. <https://doi.org/10.1093/nar/gkx376>.
11. Genomatix. [Online] [Cited: 07 30, 2021.] <http://www.genomatix.de/>.
12. Transfac Profession vs. Public. [Online] [Cited: 11 12, 2020.] https://portal.genexplain.com/archive/documents/transfac_comparison.pdf.
13. Wrzodek, C. et al. ModuleMaster: A new tool to decipher transcriptional regulatory networks. *Biosystems*. 2010, 99(1), pp. 79–81.
14. Bioconductor. Bioconductor. [Online] [Cited: 01 01, 2022.] <https://bioconductor.org>.
15. Tan, G., Lenhard, B. TFBSTools: an R/Bioconductor package for transcription factor binding site analysis. *Bioinformatics*. 32, 2016, pp. 1555–1556.
16. Eibar, S., Aerts, S. RcisTarget. [Online] [Cited: 01 03, 2022.] <https://bioconductor.org/packages/release/bioc/html/RcisTarget.html>.
17. Wei, Z., Duren, Z., Ma, S. enrichTF. [Online] [Cited: 01 03, 2022.] <https://www.bioconductor.org/packages/release/bioc/html/enrichTF.html>.
18. Alipanahi, B. et al. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat Biotechnol*. 2015, 33, pp. 831–838.
19. Kiesel, A. et al. The BaMM web server for de-novo motif discovery and regulatory sequence analysis. *Nucleic Acids Res*. July 2, 2018, 46(W1), pp. W215-W220.
20. Mathelier, A., Wasserman, W. W. The Next Generation of Transcription Factor Binding Site Prediction. *PLoS Comput Biol*. 2013, 9(9), p. e1003214.
21. Weirauch, M. T. et al. Determination and inference of eukaryotic transcription factor sequence specificity. *Cell*. Sep 11, 2014, 158(6), pp. 1431–43.
22. Hume, M. A., Barrera, L. A., Gisselbrecht, S. S., Bulyk, M. L. UniPROBE, update 2015: new tools and content for the online database of protein-binding microarray data on protein-DNA interactions. *Nucleic Acids Research*. 2014.
23. Genomatix. *Explanation of scores from Genomatix programs*. [Online] [Cited: May 14, 2021.] https://www.genomatix.de/online_help/help/scores.html.
24. Chow, C. W., Rincón, M., Davis, R. J. Requirement for transcription factor NFAT in interleukin-2 expression. *Mol Cell Biol*. 19(3), 1999, pp. 2300–2307. doi:10.1128/mcb.19.3.2300.
25. Kel, A. E. et al. MATCH: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res*. 31(13), July 1, 2003, pp. 3576-9. PMID: 12824369; PMCID: PMC169193.
26. Matys, V. et al. TRANSFAC: transcriptional regulation, from patterns to profiles. *Nucleic Acids Res*. 31(1), Jan 1, 2003, pp. 374–8. PMID: 12520026; PMCID: PMC165555.

27. Kenneth, D., Vishal, R. P., Paul, R., Xiaohui, X., Pierre, B. MotifMap: integrative genome-wide maps of regulatory motif sites for model species. *BMC Bioinformatics*. 2011, 12, p. 495. doi:10.1186/1471-2105-12-495.
28. Xie, X., Rigor, P., Baldi, P. MotifMap: a human genome-wide map of candidate regulatory motif sites. *Bioinformatics*. 2009, Vol. 25, pp. 167–174.
29. Motifmap. [Online] University of California, 2018. [Cited: August 31, 2018.] <http://motifmap.ics.uci.edu/>.
30. Messeguer, X. et al. PROMO: detection of known transcription regulatory elements using species-tailored searches. *Bioinformatics*. 2002, 18(2), pp. 333–334.
31. Farré, D. et al. Identification of patterns in biological sequences at the ALGGEN server: PROMO and MALGEN. *Nucleic Acids Res*. 2003, 31(13), pp. 3651–3. http://alggen.lsi.upc.es/cgi-bin/promo_v3/promo/promoinit.cgi?dirDB=TF_8.3.
32. Münch, R. et al. PRODORIC: prokaryotic database of gene regulation. *Nucleic Acids Res*. 2003, 31(1), pp. 266–9.
33. Münch, R., et al. Virtual Footprint and PRODORIC: an integrative framework for regulon prediction in prokaryotes. *Bioinformatics*. 2005, 21(22), pp. 4187–4189.
34. Solovyev, V. V., Shahmuradov, I. A. PromH: Promoters identification using orthologous genomic sequences. *Nucleic Acids Res*. 2003, Vol. 31(13), pp. 3540–3545.
35. Solovyev, V. V., Shahmuradov, I. A., Salamov, A. A. Identification of promoter regions and regulatory sites. *Methods Mol Biol*. 2010, Vol. 674, pp. 57–83.
36. Shahmuradov, I., Solovyev, V. Nsite, NsiteH and NsiteM Computer Tools for Studying Transcription Regulatory Elements. *Bioinformatics*. 2015, Vol. 31(21), pp. 3544–5.
37. TAIR. [Online] 2019. [Cited: 07 30, 2021.] <https://www.arabidopsis.org/tools/bulk/motiffinder/index.jsp>.
38. Chow, C. N. et al. PlantPAN3.0: a new and updated resource for reconstructing transcriptional regulatory networks from ChIP-seq experiments in plants. *Nucleic Acids Res*. 2019, 47(D1), pp. D1155-D1163.
39. CisBP. [Online] [Cited: 11 06, 2021.] <http://cisbp.cabr.utoronto.ca/TFTools.php>.
40. UniPROBE. UniPROBE. [Online] [Cited: 11 06, 2021.] http://the_brain.bwh.harvard.edu/uniprobe/index.php?ncsg=1.
41. HOCOMOCO. [Online] [Cited: 11 06, 2021.] <https://hocomoco11.autosome.ru>.
42. Kulakovskiy, I. V. et al. HOCOMOCO: towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis. *Nucl. Acids Res*. 11 11, 2017, Database issue, p. gkx1106.
43. FlyFactorSurvey. [Online] [Cited: 11 06, 2021.] <https://pgfe.umassmed.edu/ffs/>.
44. MEME Suite. [Online] [Cited: 11 06, 2021.] <https://meme-suite.org/meme/tools/meme>.
45. Bailey, T. L., Johnson, J., Grant, C. E., Noble, W. S. The MEME Suite. *Nucleic Acids Research*. 2015, 43(W1), pp. W39-W49.
46. YeTFaSCo. [Online] [Cited: 11 06, 2021.] <http://yetfasco.cabr.utoronto.ca/>.
47. Boer, C.G., Hughes, T.R. YeTFaSCo: a database of evaluated yeast transcription factor sequence specificities. *Nucleic Acids Research*. 01 01, 2012, Vol. Issue D1, Volume 40, pp. D169–D179.
48. Stormo, G. D. Dna binding sites: representation and discovery. *Bioinformatics*. 2000, 16(1), pp. 16–23.
49. Wikipedia. *Position weight matrix*. [Online] [Cited: 07 30, 2021.] https://en.wikipedia.org/wiki/Position_weight_matrix.
50. Chen, X., Guo, L., Fan, Z., Jiang, T. Learning position weight matrices from sequence and expression data. *Comput Syst Bioinformatics Conf*. 2007, 6, pp. 249–60.
51. Tang, D. Transcription factor binding site prediction. [Online] [Cited: 12 23, 2019.] <https://davetang.org/muse/2013/10/02/transcription-factor-binding-site-prediction/>.
52. Gribskov, M., McLachlan, A. D., Eisenberg, D. Profile analysis: detection of distantly related proteins. *Proc Natl Acad Sci U S A*. 1987, 84(13), pp. 4355–8.
53. Guigo, R. An introduction to position specific scoring matrices. [Online] [Cited: 07 30, 2021.] <http://bioinformatica.upf.edu/T12/MakeProfile.html>.

54. Stormo, G. D., Schneider, T. D., Gold, L., Ehrenfeucht, A. Use of the 'perceptron' algorithm to distinguish translational initiation sites in E. coli. *Nucleic Acids Res.* 1982, 10(9), pp. 2997–3011.

55. Chen, X., Guo, L., Fan, Z., Jiang, T. Learning position weight matrices from sequence and expression data. *Comput Syst Bioinformatics Conf.* 2007, 6, pp. 249–60.

56. Kreft, Ł. et al. Web: ConTra v3: a tool to identify transcription factor binding sites across species, update 2017. [Online] [Cited: 10 4, 2020.] <http://bioit2.irc.ugent.be/contra/v3/#/step/1>.

57. Guigo, R. An introduction to position specific scoring matrices. 2003.

Figures

Figure 1

Architecture of AIModules.

The three layers (frontend, backend, DB) and the flow of information are illustrated.

Figure 2

Transcription factor binding and module searches using AIModules.

Subfigures **a-d** are only excerpts; for full results visit <https://bioinfo-wuerz.de/aimodules/>; the black line represents the sequence itself. Above the line you will find the motifs of the (+) strand and below the line are the motifs from the (-) strand. The results can be downloaded from the web application as an Excel file as well; the application is described in detail in the Supplementary section "User Interface"; the sequences can be found in the Supplementary section "Used sequences"; **a**

TFBS search for the promoters X73536.1_H.sapiens_promoter_region_of_human_IL-10_gene, AY486432.1_Macaca-mulatta_interleukin-10-(IL-10)_gene_promoter_region and AF121965.1_Mus-musculus_interleukin-10-(IL10)_gene_promoter_partial_sequence (configuration: La 6, Ld 8, vertebrates matrices from Jaspar DB) (excerpt contains base pairs 250-400; results can be found in the file Supplements.zip); **b** Poly(A) TFBS search for the promoters NM_000600.5 Homo sapiens interleukin 6 (IL6)-transcript variant 1, NM_000594.4 Homo sapiens tumor necrosis factor (TNF) and NM_020525.5 Homo sapiens interleukin 22 (configuration: La 9, Ld 8, own Poly-(A)-matrix) (excerpt contains base pairs 990-1150; results can also be found in the Supplementary section “Poly-(A) results”);

c Module search for promoters Homo_sapiens_cathepsin_V, Bos_taurus_cathepsin-Z and Mus_musculus_cathepsin-F (configuration: threshold 3, La 6, Ld 8, vertebrates matrices from Jaspar DB) (excerpt contains base pairs 1-200; results can also be found in the file Supplements.zip);

d Module search for promoters Mus musculus interleukin 10, Mus musculus interleukin 15 and Mus musculus interleukin 13 (configuration: threshold 3, La 8, Ld 8, vertebrates matrices from Jaspar DB) (excerpt contains base pairs 300-550; NFAT for all sequences was found as previously described in (24); results can also be found in the Supplementary Table S 10)

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [DataSupplements.zip](#)
- [Supplements.docx](#)