Efficient Online Transcription Factor Binding Site Adjustment by Integrating Transitive Graph Projection with MoRAine 2.0

Tobias Wittkop[*1], Sven Rahmann[2], Jan Baumbach[3,4,5]

[1]Genome Informatics, Bielefeld University, Bielefeld, Germany

[2]Bioinformatics for High-Throughput Technologies, Computer Science XI, TU Dortmund, Dortmund, Germany

[3]International Computer Science Institute, Berkeley, USA

[4]Max Planck Institute for Informatics, Germany

[5]Saarland University, Cluster of Excellence for Multimodal Computing and Interaction, Germany

Summary

We investigated the problem of imprecisely determined prokaryotic transcription factor (TF) binding sites (TFBSs). We found that the identification and reinvestigation of questionable binding motifs may result in improved models of these motifs. Subsequent model-based predictions of gene regulatory interactions may be performed with increased accuracy when the TFBSs annotation underlying these models has been re-adjusted.
We present MoRAine 2.0, a significantly improved version of MoRAine. It can automatically identify cases of unfavorable TFBS strand annotations and imprecisely determined TFBS positions. With release 2.0, we close the gap between reasonable running time and high accuracy. Furthermore, it requires only minimal input from the user: (1) the input TFBS sequences and (2) the length of the flanking sequences.
Conclusions: MoRAine 2.0 is an easy-to-use, integrated, and publicly available web tool for the re-annotation of questionable TFBSs. It can be used online or downloaded as a stand-alone version from `http://moraine.cebitec.uni-bielefeld.de`.

1 Background

We recently considered the problem of imprecisely determined transcription factor (TF) binding sites (TFBSs), which are stored in public databases that are dedicated to prokaryotic gene regulatory networks [19]. We found that the identification and reinvestigation of questionable binding motifs may result in improved *in silico* models of these motifs. Hence, subsequent model-based bioinformatics predictions of gene regulatory interactions may be performed with increased accuracy [10]. We developed two tools that tackle the problem of TFBS re-adjustment: (1) MotifAdjuster, a stand-alone software and (2) MoRAine 1.0, a fast heuristic. While the MotifAdjuster algorithm is based on expectation maximization and highly accurate,

*Corresponding author: Tobias.Wittkop@CeBiTec.Uni-Bielefeld.DE.

Copyright 2010 The Author(s). Published by Journal of Integrative Bioinformatics.
This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License (http://creativecommons.org/licenses/by-nc-nd/3.0/).

MoRAine is optimized to provide a fast, online-available web solution but asks the user for various parameters to trade between running time and solution quality.

Here, we present a significantly improved version of MoRAine, which now closes the gap between reasonable running time and high accuracy. As release 1.0, MoRAine 2.0 is an online tool but it requires only minimal input from the user: (1) the input TFBS sequences and (2) the length of the flanking sequences, i.e. the expected length of the TF binding motif (TFBM). In our study, we focus on prokaryotic TFBSs, i.e. those of *Escherichia coli.*

In the following, we introduce the biological background, explain why annotation problems may occur, and show why this may result in a poor TFBS prediction performance.

Transcription factor binding site annotation - A difficult and error-prone task

Transcription factors are important components of the cell's regulatory machinery. They are DNA-binding proteins that are able to detect intra- and extracellular signals. By binding to so-called transcription factor binding sites they control the expression of their target genes and thereby decisively influence genetic programs like growth, reproduction, and defense [7,1,2,24, 22]. Given a set of known TFBSs for a certain regulator, we can build mathematical models to perform *in silico* predictions of further TFBSs in order to predict regulatory networks. This task is generally complicated by the relatively low level of TFBS conservation. The most widely used model for TFBSs are so-called position frequency matrices (PFMs) [25]. PFMs can be converted to position specific scoring matrices (PSSMs) by calculating log-odds scores. These matrices are used in turn to predict TFBSs in the upstream sequences of putative target genes for a certain TF. Various software tools are available: PoSSuMsearch [11], Virtual Footprint [21], MATCH [20], and P-MATCH [14], to name a few.

Nowadays, TFBS wet lab determination is done by electrophoretic mobility shift assays (EMSA) [17], DNAse footprinting [16], ChIP-chip [26], ChIP-seq [18], or mutations of putative TFBMs and subsequent expression studies. All of these methods lack a precise binding sequence identification that is accurate to one base pair [8]. Another problem occurs: Since TFs bind the double-stranded DNA, it is a matter of interpretation which strand of the TF-binding sequence is annotated. Clearly, both issues directly affect and complicate TFBS modeling as position frequency matrices and hence, the subsequent PSSM-based binding site prediction. This problem occurs when a TFBS from either strand based on approximate knowledge of its position is entered in a reference database and subsequently used blindly for PSSM-based predictions. This does happen in practice for regulatory databases that integrate information from other sources [19, 10], for instance, in CoryneRegNet [3, 4, 5, 8, 9]. Here, the data is added manually to the data repository by curation teams. They scan the literature and transfer the published knowledge into a formal data structure. This task is difficult, error-prone and, hence, further supports putative mistakes with the TFBS annotation process.

For mis-annotated TFBSs, we may observe a poor information content of the subsequently computed PFM, which consequently leads to a decreased binding motif prediction for the PSSM that was constructed from that PFM. We attack this problem by re-annotating the TFBSs by possibly switching their strands and/or shifting them a few positions, in order to maximize the information content of the resulting PFM.

Therefore, MoRAine 1.0 provided a combination of two clustering approaches, a variant of

Copyright 2010 The Author(s). Published by Journal of Integrative Bioinformatics.
This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License (http://creativecommons.org/licenses/by-nc-nd/3.0/).

k-means (km) and cluster growing (cg), applied to two similarity measures, cluster similarity ($simC$) and seed node similarity ($simS$). In the following, we describe why and how we replaced these methods in MoRAine 2.0 in order to provide better results within decreased running times. Afterwards, we compare both releases and demonstrate the increased TFBS prediction performance. Subsequently, we show that the PFMs resulting from MoRAine-adjusted TFBSs significantly improve the prediction performance of further binding sites in practice, since the adjusted TFBSs lead to PFMs with higher information content.

2 Methods

We need the following definitions to explain how MoRAine works and to compare the readjustment performances of MoRAine 1.0 and 2.0.

Let $\Sigma := \{\text{A,T,C,G}\}$ be the DNA alphabet. In accordance with [10], a position frequency matrix $F = (f_{\sigma j})$ for a set of n TFBSs of length m over the alphabet Σ is defined as a $|\Sigma| \times m$ matrix, where $f_{\sigma j}$ is the relative frequency of letter σ at position j.

Crooks *et al.* introduced in [15] the information content as quality measure for PFMs. The information content I_j for column j of F is defined as

$$I_j := \log_2 |\Sigma| + \sum_{\sigma \in \Sigma} f_{\sigma j} \cdot \log_2 f_{\sigma j} \quad \text{[bits]}.$$

If all symbols at position j agree, I_j reaches its maximum with maximal value 2 bits for a 4-letter alphabet Σ. The mean information content $I(F)$ for a given PFM F is defined as the average I_j over all positions j:

$$I(F) := \frac{1}{m} \sum_{j=1}^{m} I_j.$$

In what follows, we use the mean information content $I(F)$ as a quality measure for a given PFM F and denote it shortly with I if F is fixed. We will use the information content to compare the quality of two different PFMs F_1 and F_2 by comparing $I(F_1)$ with $I(F_2)$. If F_2 is the PFM of the MoRAine-adjusted TFBSs, while F_1 is the PFM computed from the input TFBSs, with $I(F_1) \leq I(F_2)$, we can calculate the percentage improvement performance P with $P = 100 \cdot \frac{I(F_2)}{I(F_1)}$.

MoRAine now works as follows: The input is a set of n annotated length-m TFBS sequences that extend l bp to the left and r bp to the right. Hence, the length of the given input sequences is $m^+ := m + l + r$. First, MoRAine computes the set M of every possible motif of length $m = m^+ - l - r$ derived by the operations *shift* and *switch* applied to each of the n input sequences. The operation *shift* provides every substring of length m for a given motif of length m^+, and the operation *switch* its reverse complement sequence. We obtain a set S_i of $M := |S_i| = 2 \cdot (l + r + 1)$ potential TFBS sequences of length m for each input sequence i, with $i = 1, \ldots, n$.

So far MoRAine 1.0 and 2.0 work in a similar way. For both, the goal is to find a set C of TFBSs that contains exactly one TFBS from each S_i and maximizes the mean information content of the corresponding PFM F_C. As mentioned earlier, MoRAine 1.0 offers two heuristic clustering algorithms, (cg) and (km), both working on either of two similarity functions, ($simC$)

Copyright 2010 The Author(s). Published by Journal of Integrative Bioinformatics.
This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License (http://creativecommons.org/licenses/by-nc-nd/3.0/).

and ($simS$). Table 1 summarizes the running times and TFBS annotation improvement performance of MoRAine 1.0 for all four combinations. One can see a trade-off between accuracy and running time: (cg/$simS$) provides best results but (cg/$simC$) is much faster (see section Results and Discussion for more details).

With MoRAine 2.0, we close this gap and provide a powerful tool that now provides better results than MoRAine 1.0 with (cg/$simS$) at running times equal to (cg/$simC$). The goal can be cast as follows: We partition the set of input TFBSs into $M = 2 \cdot (l + r + 1)$ clusters, where each cluster contains exactly n motifs, one of each S_i ($i = 1, \ldots, n$) and thus is a putative solution. In the following, we describe how we adapted and integrated Transitivity Clustering with MoRAine 2.0 to find such a set C.

Transitivity Clustering is a clustering method based on the weighted transitive graph projection (WTGP) problem. By solving this NP-complete graph modification problem, objects are partitioned into groups of similar elements. We briefly describe the underlying WTGP problem and how it has been modified to fulfill the needs for this specific task. Given a set of objects V and a pairwise similarity function $s : \binom{V}{2} \to \mathbb{R}$ a similarity graph $G = (V, E)$ is constructed, where V is the set of objects and E is the set of undirected edges between these objects. An edge is present in G if the similarity between the adjacent nodes exceeds a user defined threshold t. The goal is to find a *transitive* graph $G' = (V, E')$ with smallest distance to G. Transitivity means that for all triples $\{u, v, w\} \in \binom{V}{3}$: $\{u, v\} \in E$ and $\{v, w\} \in E$ implies $\{u, w\} \in E$. A transitive graph is a disjoint union of cliques, also called a cluster graph. Therefore a cost function for adding/deleting edges is defined as $c(u, v) = |t - s(u, v)|$ serving as optimization function. Thus, the distance between two edge sets is the cost of transforming one into the other. The resulting disjoint cliques of the minimum cost transitive G' represent the clusters we are looking for.

Transitivity Clustering is flexible and offers the possiblity to integrate additional knowledge. As similarity function, instead of the functions from MoRAine 1.0, (cg) and (km), we now use the difference between the motif length $\ell = |p| = |q|$ and the hamming distance $h(p, q)$ between two TFBSs p, q, hence $s(p, q) := \ell - h(p, q)$. To ensure that each cluster of TFBSs contains only one motif from each set S_i, we set the similarity function s to $-\infty$ if $p \in S_i$ and $q \in S_i$ for some S_i, i.e. if both potential solutions (the TFBSs p and q) originate from the same input TFBS. The threshold t is set to zero, which guaranties that each cluster contains exactly one TFBS from each set S_i. Transitivity Clustering has successfully been applied to protein family detection using the layout based heuristic FORCE to solve the NP-complete WTGP problem [27]. Together with an exact fixed parameter algorithm developed by Böcker et al. [13] and the fast but less accurate heuristic CAST (Cluster Affinity Search Technique) by Ben-Dor et al. [12], this layout-based approach has been integrated into the clustering framework TransClust. The TransClust software combines the different methods to provide very accurate results in reasonable time. Its integration with MoRAine is mainly responsible for the increased performance of MoRAine 2.0, as we will demonstrate in the following section. Further information about Transitivity Clustering is available at the web site `http://transclust.cebitec.uni-bielefeld.de`.

Copyright 2010 The Author(s). Published by Journal of Integrative Bioinformatics.
This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License (http://creativecommons.org/licenses/by-nc-nd/3.0/).

3 Results and Discussion

3.1 Implementation

MoRAine 2.0 is an open source JAVA 6 program. It can accessed and downloaded at `http://moraine.cebitec.uni-bielefeld.de`. As shown for MoRAine 1.0 in [6], release 2.0 of MoRAine may be included into a database back-end as quality assurance tool or to provide a bioinformatics workflow with adjusted position weight matrices for TFBS predictions.

We emphasize that the main advantage of MoRAine is it's easy-to-use web interface. The user may copy and paste binding sequences in FASTA format at the MoRAine web site to calculate the adjusted motifs as well as the corresponding sequence logos by using the Berkeley web logo software [15]. Just as MoRAine 1.0, the second release is an easy-to-use alternative for the computation of sequence logos and the adjustment of transcription factor binding sites, but it now provides increased accuracy at decreased running times and an eased user-interface with less parameters to adjust.

3.2 Increased information content improvement with MoRAine 2.0

In Figure 1 we exemplarily illustrate the output of the MoRAine online service for the binding sites of the transcription factor RamB of *Corynebacterium glutamicum*. The TFBSs have been taken from CoryneRegNet release 5.0. As in most databases, in CoryneRegNet [8], each binding site is annotated in $5' \rightarrow 3'$ direction relative to the regulated target gene. By using MoRAine 2.0 we improved the average information content from 0.64 (original database TFBSs) to 1.15 (MoRAine-adjusted TFBSs) by switching the strands for 15 of the 38 input sequences. The computation time was less than 2 seconds.

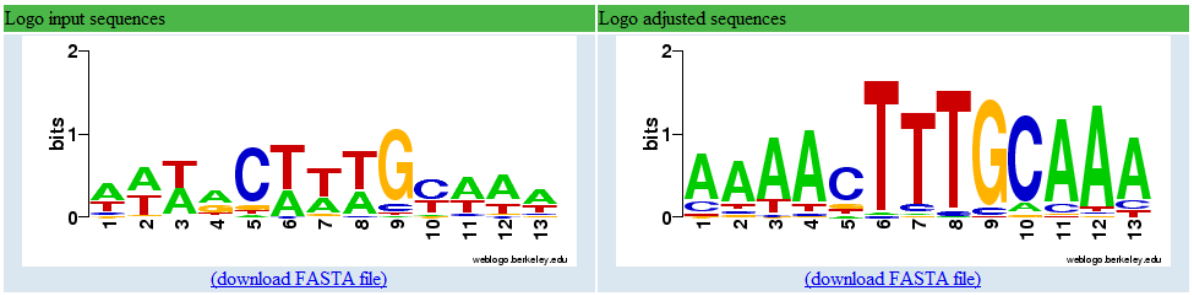

Figure 1: A screenshot from the MoRAine 2.0 web site. A comparison of the sequence logos constructed from the original TFBSs (left side) for the transcription factor RamB of *Corynebacterium glutamicum* and the adjusted TFBSs by using MoRAine 2.0 (right side).

To demonstrate the performance, i.e. decreased running time and increased information content improvement, of MoRAine 2.0, we used the same datasets as in [10]: 1165 binding sites of 85 transcription factors of *Escherichia coli*. We compare the average runtime and the mean information content improvement of MoRAine 2.0 with the four methods implemented in MoRAine 1.0 for different lengths of the flanking sequences (l and r, respectively). As shown in Table 1, with MoRAine 1.0 the combination ($cg/simC$) had the best runtime, but to gain the best information content improvement, we used the combination ($cg/simS$) [10]. With the work

Copyright 2010 The Author(s). Published by Journal of Integrative Bioinformatics.
This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License (http://creativecommons.org/licenses/by-nc-nd/3.0/).

presented in this paper, we closed the gap between running time and accuracy. In Table 2, we compare MoRAine 2.0 with release 1.0 using the most accurate combination ($cg/simS$) and the fastest combination ($cg/simC$), respectively. For a fair running time comparison, we re-evaluated MoRAine 1.0 ($cg/simC$) and MoRAine 2.0 on the same standard desktop PC. Table 2 shows that MoRAine 2.0 outperforms the previous release in terms of information content improvement with running times almost as fast as those of ($cg/simC$). Furthermore, MoRAine 2.0 does not require the user to choose various input parameters to optimize its results.

	Difference (%)				Time (s)			
$l = r$	$cg/simC$	$cg/simS$	$km/simC$	$km/simS$	$cg/simC$	$cg/simS$	$km/simC$	$km/simS$
0	26.1	**27.0**	26.5	26.8	**0.6**	0.7	1.2	1.1
1	50.9	**54.4**	50.1	52.3	**0.7**	2.3	7.2	4.0
2	57.5	**63.6**	57.6	62.4	**0.8**	4.2	45.9	8.3
3	60.0	**69.5**	64.6	64.7	**1.0**	8.4	128.0	12.8
4	65.3	**70.1**	65.0	69.3	**1.1**	11.9	198.3	19.5
5	66.3	73.0	68.8	**73.3**	**1.3**	16.8	298.3	30.5
6	66.6	73.1	74.3	**74.9**	**1.8**	23.9	427.0	34.4
7	68.0	**78.7**	73.5	78.4	**2.0**	30.1	505.4	42.6

Table 1: This table was taken from [10] and summarizes the average information content improvements and the mean running times of MoRAine 1.0 for different l- and r-values and the four search method/similarity function combinations over all TFBSs of 85 transcriptional regulators of *E. coli.*

	Difference (%)		Time (s)	
$l = r$	MoRAine 1.0 ($cg/simS$)	MoRAine 2.0	MoRAine 1.0 ($cg/simC$)	MoRAine 2.0
0	27.0	27.2	0.21	0.23
1	54.4	54.7	0.26	0.29
2	63.6	66.5	0.32	0.36
3	69.5	72.2	0.38	0.42
4	70.1	75.5	0.46	0.50
5	73.0	75.7	0.55	0.59
6	73.1	77.8	0.60	0.66
7	78.7	79.1	0.71	0.77

Table 2: In this table we compare the average information content improvement and the mean running time of MoRAine 1.0 with MoRAine 2.0 for different l- and r-values over all TFBSs of 85 transcriptional regulators of *E. coli.* We compare MoRAine 2.0 with the most accurate combination of similarity function and search method of MoRAine 1.0 (left side) and with the fastest combination (right side).

3.3 Improved binding site prediction performance with MoRAine-adjusted sequences

As mentioned earlier, positions specific scoring matrices (PSSMs) are used for the prediction of TFBSs in sequences upstream of putatively regulated target genes or operons for a specific regulator. A PSSM allows us to assign a score to any length-m DNA sequence window. A PSSM matches such a window if the score exceeds a user-given threshold. Such a matching binding site is considered to be a good candidate for a real TFBS if we properly choose the score (generally as the log-odds score between the nucleotide distribution of true binding sites

Copyright 2010 The Author(s). Published by Journal of Integrative Bioinformatics.
This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License (http://creativecommons.org/licenses/by-nc-nd/3.0/).

and a background distribution) and the threshold (ideally based on statistical considerations; see e.g. [23]). As in [10], we use the PSSM-based DNA matching tool PoSSuMsearch [11] for the evaluation of the prediction performance of PSSMs computed from both the original TFBSs and the MoRAine-adjusted PSSMs. The threshold is computed efficiently by PoSSuMsearch based on the tolerable frequency of hits in random sequences (p-value) generated from a background model (the nucleotide frequencies in the upstream sequences); for more details refer to [11]. We show that by using MoRAine 2.0 as preprocessing for the TFBSs that are used for PSSM calculation, the classification performance is significantly increased.

Again, we use the same datasets as in [10]: the same 1165 binding sites for the 85 transcription factors from *E. coli* and 3341 upstream sequences of all transcription units (genes or operons, respectively). Here, an upstream region is defined as the DNA sequence from -560 to $+20$ bps upstream to the start codon of a transcription unit. For each PSSM, both forward and reverse strand of upstream sequences are used to predict TFBSs with PoSSuMsearch for varying p-value thresholds. For each threshold and PSSM, we compute precision, recall and F-measure over all putative upstream sequences, with:

$$\text{Precision} := \frac{TP}{TP + FN}$$

$$\text{Recall} := \frac{TP}{TP + FP}$$

$$\text{F-measure} := 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

where $FP :=$ number of incorrectly predicted motifs, $FN :=$ number of wrongly not predicted motifs, and $TP :=$ number of correctly predicted motifs. We extracted the correct motifs from the CoryneRegNet database and compared them with the predicted binding sites to compute the number of FP, FN, and TP.

In Figure 2, we plot precision vs. recall for varying p-value thresholds for all PSSMs readjusted with MoRAine 2.0 for $l = r = 4$ (blue curve) in comparison to the prediction performance obtained with the original PSSMs (red curve) and to the performance when using MoRAine 1.0 adjusted PSSMs (green curve). Note that MoRAine 1.0 was used in its most accurate mode ($cg/simS$). For a fixed recall, the MoRAine-adjusted precision is always higher than with original, not adjusted TFBSs/PSSMs. Figure 3 plots the F-measure against different p-value thresholds. The plot show that predictions based on adjusted PSSMs outperform those based on original PSSMs for all thresholds. Both plots illustrate that although MoRAine 2.0 is as fast as the previous release in its fastest mode, it is as helpful as MoRAine 1.0 in its most accurate mode. In additional material (`http://moraine.cebitec.uni-bielefeld.de/download/additional_file1.pdf`), we also provide a brief comparison of MoRAine with six de-novo motif discovery tools by using data from [19].

4 Conclusions

In [19, 10], we showed that imprecisely determined prokaryotic transcription factor binding sites stored in public databases may cause problems with the prediction of further binding sites. The exact determination of the TFBS positions down to one basepair is difficult, the strand

Copyright 2010 The Author(s). Published by Journal of Integrative Bioinformatics.
This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License (http://creativecommons.org/licenses/by-nc-nd/3.0/).

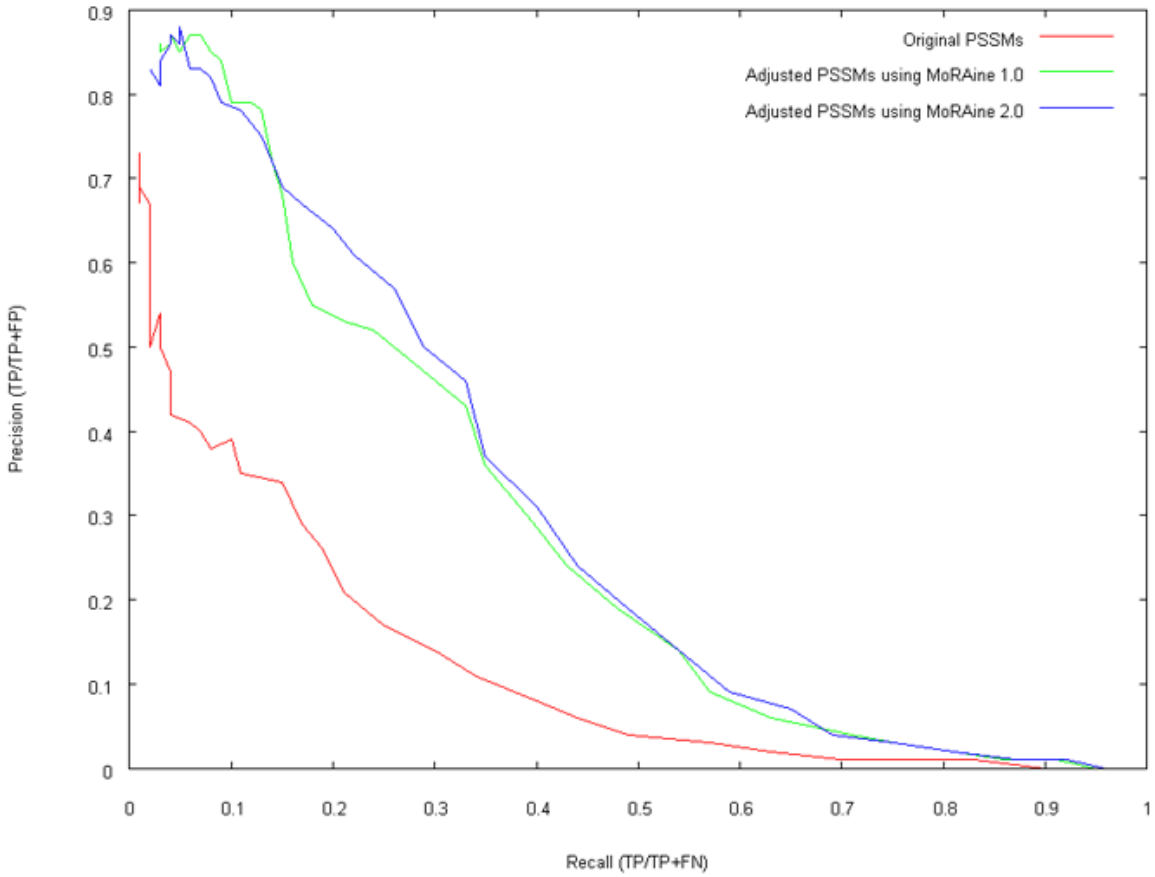

Figure 2: Prediction performance comparison of PoSSuMsearch by means of precision and recall. All values are measured for varying p-value thresholds based on PSSMs learned from the original TFBSs (red line) compared to those of readjusted TFBSs with MoRAine 2.0 (blue line) and readjusted TFBSs with MoRAine 1.0 (green line).

annotation is sometimes neglected, and TF binding sequences are often stored $5' \rightarrow 3'$ relative to the target gene. We demonstrated that the identification and reinvestigation of questionable TFBSs in *E. coli* may result in improved *in silico* models of these motifs and improve the subsequent prediction performance.

To address this problem, we presented an extended version of MoRAine, which now integrates weighted transitive graph projection (by means of Transitivity Clustering) as data partitioning method. MoRAine 2.0 now provides increased accuracy together with decreased running times, if compared to MoRAine 1.0. Now, it is as fast as the previous release in its fastest mode but the optimization performance is even better. Furthermore, MoRAine 2.0 does not require the user to adjust various parameters to achieve these results. It only requires the necessary input to solve the readjustment problem, i.e. the input sequences themselves and the length of the flanking sequences. However, we see the main advantage of MoRAine in its integrative web interface, which runs on a non-dedicated web server. Biologists may visit the MoRAine web site, copy and paste their TFBS sequences and obtain readjusted sequences for download along with the sequence logos.

In summary, this article introduces an improved version of MoRAine, an online tool that supports the re-annotation of transcription factor binding sites. We provide a web server to fa-

Copyright 2010 The Author(s). Published by Journal of Integrative Bioinformatics.
This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License (http://creativecommons.org/licenses/by-nc-nd/3.0/).

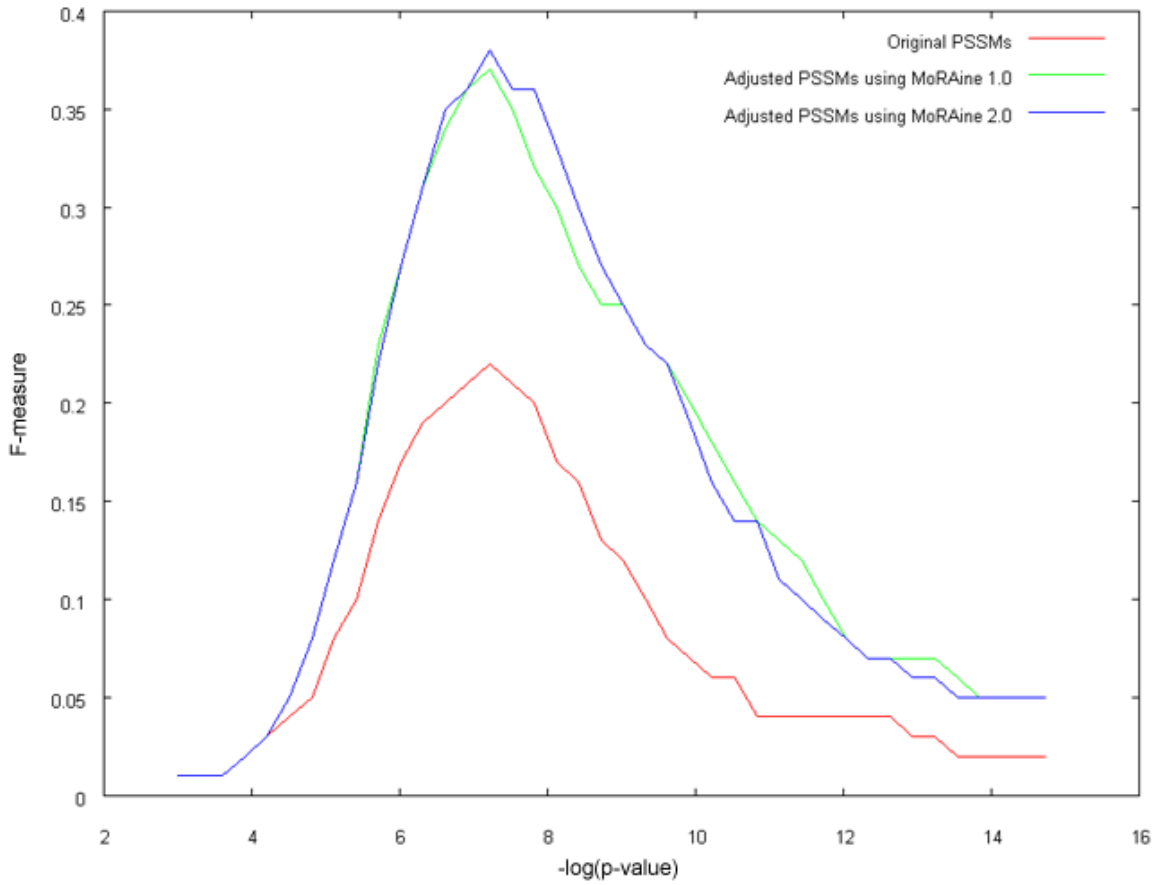

Figure 3: Prediction performance comparison by means of plotting the F-measure for varying PoSSuMsearch p-value thresholds for the original TFBSs (red line), the MoRAine 1.0-adjusted TFBSs (green line), and the MoRAine 2.0-adjusted TFBSs (blue line) allowing 4 shifts to the left and right (l=r=4).

cilitate using MoRAine and to compute sequence logos. We further demonstrated that the re-annotation of TFBSs may be necessary for some prokaryotic databases and helps to improve the PSSM-based prediction performance. MoRAine may also be downloaded as stand-alone tool and integrated in any data processing pipeline.

Acknowledgements

This work was partially supported by the Cluster of Excellence for Multimodal Computing and Interaction and by the German Academic Exchange Service (DAAD).

References

[1] Babu MM, Luscombe NM, Aravind L, Gerstein M, and Teichmann SA. Structure and evolution of transcriptional regulatory networks. *Curr Opin Struct Biol*, 14(3):283–291, Jun 2004.

Copyright 2010 The Author(s). Published by Journal of Integrative Bioinformatics.
This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License (http://creativecommons.org/licenses/by-nc-nd/3.0/).

[2] Babu MM, Teichmann SA, and Aravind L. Evolutionary dynamics of prokaryotic transcriptional regulatory networks. *J Mol Biol*, 358(2):614–633, Apr 2006.

[3] Baumbach J. CoryneRegNet 4.0 - A reference database for corynebacterial gene regulatory networks. *BMC Bioinformatics*, 8:429, 2007.

[4] Baumbach J, Brinkrolf K, Czaja L, Rahmann S, and Tauch A. CoryneRegNet: An ontology-based data warehouse of corynebacterial transcription factors and regulatory networks. *BMC Genomics*, 7(1):24, Feb 2006.

[5] Baumbach J, Brinkrolf K, Wittkop T, Tauch A, and Rahmann S. CoryneRegNet 2: An Integrative Bioinformatics Approach for Reconstruction and Comparison of Transcriptional Regulatory Networks in Prokaryotes. *Journal of Integrative Bioinformatics*, 3(2):24, 2006.

[6] Baumbach J, Rahmann S, and Tauch A. Reliable transfer of transcriptional gene regulatory networks between taxonomically related organisms. *BMC Syst Biol*, 3:8, 2009.

[7] Baumbach J, Tauch A, and Rahmann S. Towards the integrated analysis, visualization and reconstruction of microbial gene regulatory networks. *Brief Bioinform*, 10(1):75–83, Jan 2009.

[8] Baumbach J, Wittkop T, Kleindt CK, and Tauch A. Integrated analysis and reconstruction of microbial transcriptional gene regulatory networks using CoryneRegNet. *Nat Protoc*, 4(6):992–1005, 2009.

[9] Baumbach J, Wittkop T, Rademacher K, Rahmann S, Brinkrolf K, and Tauch A. CoryneRegNet 3.0-An interactive systems biology platform for the analysis of gene regulatory networks in corynebacteria and Escherichia coli. *J Biotechnol*, 129(2):279–289, Apr 2007.

[10] Baumbach J, Wittkop T, Weile J, Kohl T, and Rahmann S. MoRAine - A web server for fast computational transcription factor binding motif re-annotation. *Journal of Integrative Bioinformatics*, 5(2):91, 2008.

[11] Beckstette M, Homann R, Giegerich R, and Kurtz S. Fast index based algorithms and software for matching position specific scoring matrices. *BMC Bioinformatics*, 7:389, 2006.

[12] Ben-Dor A, Shamir R, and Yakhini Z. Clustering gene expression patterns. *J Comput Biol*, 6(3-4):281–297, 1999.

[13] Böcker S, Briesemeister S, Bui QBA, and Truss A. Going Weighted: Parameterized Algorithms for Cluster Editing. *Theor. Comput. Sci.*, 2009. Doi:10.1016/j.tcs.2009.05.006.

[14] Chekmenev DS, Haid C, and Kel AE. P-Match: transcription factor binding site search by combining patterns and weight matrices. *Nucleic Acids Research*, 33(Web Server issue):W432–W437, Jul 2005.

[15] Crooks GE, Hon G, Chandonia JM, and Brenner SE. WebLogo: a sequence logo generator. *Genome Res*, 14(6):1188–1190, Jun 2004.

Copyright 2010 The Author(s). Published by Journal of Integrative Bioinformatics.
This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License (http://creativecommons.org/licenses/by-nc-nd/3.0/).

[16] Galas DJ and Schmitz A. DNAse footprinting: a simple method for the detection of protein-DNA binding specificity. *Nucleic Acids Research*, 5(9):3157–3170, Sep 1978.

[17] Hellman LM and Fried MG. Electrophoretic mobility shift assay (EMSA) for detecting protein-nucleic acid interactions. *Nat Protoc*, 2(8):1849–1861, 2007.

[18] Jothi R, Cuddapah S, Barski A, Cui K, and Zhao K. Genome-wide identification of in vivo protein-DNA binding sites from ChIP-Seq data. *Nucleic Acids Res*, 36(16):5221–5231, Sep 2008.

[19] Keilwagen J, Baumbach J, Kohl TA, and Grosse I. MotifAdjuster: a tool for computational reassessment of transcription factor binding site annotations. *Genome Biol*, 10(5):R46, 2009.

[20] Kel AE, Gössling E, Reuter I, Cheremushkin E, Kel-Margoulis OV, and Wingender E. MATCH: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Research*, 31(13):3576–3579, Jul 2003.

[21] Münch R, Hiller K, Grote A, Scheer M, Klein J, Schobert M, and Jahn D. Virtual Footprint and PRODORIC: an integrative framework for regulon prediction in prokaryotes. *Bioinformatics*, 21(22):4187–4189, Nov 2005.

[22] Pabo CO and Sauer RT. Transcription factors: structural families and principles of DNA recognition. *Annu Rev Biochem*, 61:1053–1095, 1992.

[23] Rahmann S, Müller T, and Vingron M. On the power of profiles for transcription factor binding site detection. *Stat Appl Genet Mol Biol*, 2:Article7, 2003.

[24] Resendis-Antonio O, Freyre-González JA, Menchaca-Méndez R, Gutiérrez-Ríos RM, Martínez-Antonio A, Avila-Sánchez C, and Collado-Vides J. Modular analysis of the transcriptional regulatory network of E. coli. *Trends Genet*, 21(1):16–20, Jan 2005.

[25] Stormo GD. DNA binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23, Jan 2000.

[26] Sun LV, Chen L, Greil F, Negre N, Li TR, Cavalli G, Zhao H, Steensel BV, and White KP. Protein-DNA interaction mapping using genomic tiling path microarrays in Drosophila. *Proc Natl Acad Sci U S A*, 100(16):9428–9433, Aug 2003.

[27] Wittkop T, Baumbach J, Lobo F, and Rahmann S. Large scale clustering of protein sequences with FORCE – A layout based heuristic for weighted cluster editing. *BMC Bioinformatics*, 8(1):396, Oct 2007.

Copyright 2010 The Author(s). Published by Journal of Integrative Bioinformatics.
This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License (http://creativecommons.org/licenses/by-nc-nd/3.0/).