MoRAine - A web server for fast computational transcription factor binding motif re-annotation

Jan Baumbach[*1,2,3], Tobias Wittkop[1,2,4], Jochen Weile[1,2], Thomas Kohl[3,5], Sven Rahmann[1,6]

[1]Computational Methods for Emerging Technologies, Bielefeld University, Bielefeld, Germany

[2]Genome informatics, Bielefeld University, Bielefeld, Germany

[3]International Graduate School in Bioinformatics and Genome Research, Center for Biotechnology, Bielefeld, Germany

[4]DFG Graduiertenkolleg Bioinformatik, Bielefeld University, Bielefeld, Germany

[5]Lehrstul für Genetik, Bielefeld University, Bielefeld, Germany

[6]Bioinformatics for High-Throughput Technologies, TU Dortmund, Dortmund, Germany

Summary

Background: A precise experimental identification of transcription factor binding motifs (TFBMs), accurate to a single base pair, is time-consuming and difficult. For several databases, TFBM annotations are extracted from the literature and stored $5' \rightarrow 3'$ relative to the target gene. Mixing the two possible orientations of a motif results in poor information content of subsequently computed position frequency matrices (PFMs) and sequence logos. Since these PFMs are used to predict further TFBMs, we address the question if the TFBMs underlying a PFM can be re-annotated automatically to improve both the information content of the PFM and subsequent classification performance.
Results: We present MoRAine, an algorithm that re-annotates transcription factor binding motifs. Each motif with experimental evidence underlying a PFM is compared against each other such motif. The goal is to re-annotate TFBMs by possibly switching their strands and shifting them a few positions in order to maximize the information content of the resulting adjusted PFM. We present two heuristic strategies to perform this optimization and subsequently show that MoRAine significantly improves the corresponding sequence logos. Furthermore, we justify the method by evaluating specificity, sensitivity, true positive, and false positive rates of PFM-based TFBM predictions for *E. coli* using the original database motifs and the MoRAine-adjusted motifs. The classification performance is considerably increased if MoRAine is used as a preprocessing step.
Conclusions: MoRAine is integrated into a publicly available web server and can be used online or downloaded as a stand-alone version from `http://moraine.cebitec.uni-bielefeld.de`.

[*]Corresponding author: Jan.Baumbach@CeBiTec.Uni-Bielefeld.DE.

1 Background

DNA-binding transcription factors (TFs) are important components of transcriptional regulatory networks that act on environmental and intracellular signals and control cellular reproduction, growth, and defense [3, 4, 5]. Depending on the environmental and internal conditions of a microorganism, a certain fraction of the total set of transcription factors is active at any given time [23]. Some of them only control the expression of a single gene, whereas others organize the activation or repression of numerous target genes [27]. TFs contain DNA-binding domains that recognize the operator sequences of controlled target genes [21]. These DNA sequences are more or less conserved, and we refer to them as transcription factor binding motifs (TFBMs).

Given a set of TFBMs, one can construct models to perform *in silico* predictions of cognate operator sequences in order to predict the regulatory network. Generally, this is complicated by the relatively low level of TFBM conservation. The by far most widely used model for TFBMs are position frequency matrices (PFMs) as introduced in [25]. PFMs can be converted to position weight matrices (PWMs), also called position-specific score matrices (PSSMs) by taking log-odds; in turn they may be used to scan the upstream sequences of putative target genes by utilizing programs like PoSSuMsearch [10] in CoryneRegNet [8], Virtual Footprint [20] in PRODORIC [19], or MATCH [17], and P-MATCH [11] in TRANSFAC [28, 18], to name a few.

Obviously, an important prerequisite for the construction of PFMs is an accurate annotation of TFBMs. The motifs are subsequently stored in gene regulatory databases such as the aforementioned ones, and additionally used for the genome-wide computational prediction of gene regulatory interactions. The determination of TFBMs in wet lab experiments is time-consuming and error-prone. Often, the position within the double-stranded DNA sequence to which a TF binds can be determined by electrophoretic mobility shift assays (EMSA) [15], DNAse footprinting [13], ChIP-chip [26], or mutations of putative TFBMs and subsequent expression studies. All of these methods lack a precise identification that is accurate to one base pair (bp). Generally, TFs bind the double-stranded DNA and it is a matter of interpretation which strand of the DNA sequence is annotated (for example, the binding sequence AGGCAT on the forward strand is equivalent to the sequence ATGCCT on the reverse strand). Conceptually, this poses no problem, since given either motif, its reverse complement is easily computed. However, a practical problem occurs, especially for PFM construction, when a motif from either strand based on approximate knowledge of its position is entered in a database and subsequently used blindly for PFM construction. This does happen in practice, especially for regulatory databases that integrate information from other sources, e.g., in RegulonDB [24], or CoryneRegNet [6, 9, 7]). Here all TFBMs are given $5' \rightarrow 3'$ (forward) relative to the target gene.

Since the stored motif is essentially chosen from a random strand, subsequently constructed PFMs may show a poor information content (e.g., a mixture of AGGCAT and ATGCCT instead of either motif) that consequently leads to bad binding motif predictions from the PFM.

In this article, we introduce MoRAine, an algorithm and software that assists with automatic TFBM re-annotation. Each motif with experimental evidence underlying a PFM is compared against each other such motif. The goal is to re-annotate the TFBMs by possibly switching their strands and/or shifting them a few positions, in order to maximize the information content of the resulting PFM. In a previous manuscript the idea of re-annotating TFBMs by switching

their strands and shifting them a few positions was introduced in [16]. While the MotifAdjuster algorithm in [16] is based on expectation maximization, MoRAine is optimized to provide a fast, online-available web solution.

The remainder of this article is structured as follows. First, we give the necessary definitions. Next, we show that both methods implemented in MoRAine significantly increase the matrix quality by means of two examples calculated with the MoRAine web server version. In one example, we adjust the TFBMs of the regulator NarL of *Escherichia coli*. We show that the corresponding sequence logo looks very similar to the manually re-annotated one, which is stored in the PRODORIC database. We discuss the same for the regulator MalT. Subsequently, we show that the PFMs resulting from the adjusted TFBMs significantly improve the prediction performance of further binding sites. MoRAine-adjusted PFMs increases the sensitivity at fixed specificity and decrease both the false positive and the false negative rates. Finally, in the Methods section, we present details about the algorithms for motif re-annotation.

1.1 Definitions

In what follows, let $\Sigma := \{\text{A,T,C,G}\}$ be the DNA alphabet.

One of the most widely used models to describe a set of TFBMs for a given TF is a position frequency matrix (PFM), defined as follows: Given a set of TFBMs of length m over the alphabet Σ, a *position frequency matrix* $F = (f_{\sigma j})$ for a set of n TFBMs of length m is a $|\Sigma| \times m$ matrix, where $f_{\sigma j}$ is the frequency of symbol σ at position j.

Information content based sequence logos can be used to judge the PFM quality [12]. The *information content* I_j for column j of a PFM F is defined as

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

I_j reaches its maximum if and only if all symbols at position j agree; for $|\Sigma| = 4$, the maximal value is 2 bits. The *mean information content* $I(F)$ for a whole frequency matrix F is

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

We use the mean information content as a quality measure and denote it shortly with I if the matrix F is fixed.

1.2 Information content maximization

We start with a set of DNA sequences that extend l bp to the left and r bp to the right of the annotated TFBMs and set $m^+ := m + l + r$ to the length of the given sequences. Given a set of n sequences of length m^+, we first calculate the set M of every possible motif of length $m = m^+ - l - r$ derived by the operations *shift* and *switch* applied to every sequence. The operation *shift* provides every substring of length m for a given motif of length m^+, and the operation *switch* its reverse complements. This leads to a set S_i of $M := |S_i| = 2 \cdot (l + r + 1)$ motifs of length m for each input sequence i, with $i = 1, \ldots, n$.

The goal of this work is to find a set C of motifs that contains exactly one motif from each S_i and maximizes the information content of the corresponding frequency matrix F_C. We propose two heuristic algorithms (cluster growing (cg) and a k-means variation (km)) to find such a motif set C. Both algorithms use of two similarity funtions ($simC$ or $simS$). More details can be found in the Methods section.

2 Results and Discussion

2.1 Software and web server

MoRAine is written in JAVA. It is open source and can be downloaded at the project web site (`http://moraine.cebitec.uni-bielefeld.de`). The stand-alone version of MoRAine can easily be included into a database back-end (i) as quality assurance and (ii) to additionally provide adjusted PFMs for subsequent TFBM predictions.

Furthermore, MoRAine can be used as a web application. The user can copy and paste lists of TFBMs in FASTA format. Using such a list as input, the MoRAine web server calculates (i) the adjusted TFBMs, and (ii) the corresponding sequence logos using the Berkley web logo library [12]. MoRAine thus becomes an easy-to-use alternative for the computation of sequence logos, since it directly integrates PFM quality improvement. The adjusted TFBMs can be downloaded in FASTA format and used to build adjusted PFMs.

2.2 Information content improvement

Figure 1 illustrates two example outputs of the MoRAine web server for the transcriptional regulators NarL and MalT of *Escherichia coli*. One can see that the average information content is significantly improved. For NarL, we allowed to shift the motifs by at most one position to the left or to the right ($l = r = 1$). Therefore, we added one base pair to the left and to the right of the annotated TFBMs as flanking sequences. We provide both examples as application cases at the MoRAine web site.

The manually curated database of prokaryotic transcriptional regulations PRODORIC [19, 20] also provides TFBMs and sequence logos for NarL at [2] and MalT at [1]. As in most databases, in CoryneRegNet [6, 7] and RegulonDB [24], each TFBM is annotated in $5' \rightarrow 3'$ direction relative to the regulated target gene. Similar to our automated approach, the database annotaters of PRODORIC improved the TFBMs annotations manually. They utilized the same operations to the TFBMs as MoRAine, namely *shift* and *switch*. Additionally, they removed or shortened TFBMs if necessary and beneficial. In the case of NarL both adjusted sequence logos look very similar. In the case of MalT, the PRODORIC annotators chose (i) to shorten the motifs from 10 bps to 6 bps and (ii) to use the reverse complement TFBM sequences. We can reproduce this annotation by using the 10 bps TFBMs of MalT as input for MoRAine and set the user defined parameters $l = r = 2$.

An impression of how the running time scales with the number of input sequences is illustrated in Figure 2 (for $l = r = 2$). The fastest combination of search algorithm and similarity function is cg/$simC$. In order to decide which combination of search strategy and similarity function performs best in general, we compared the runtimes and the average improvement

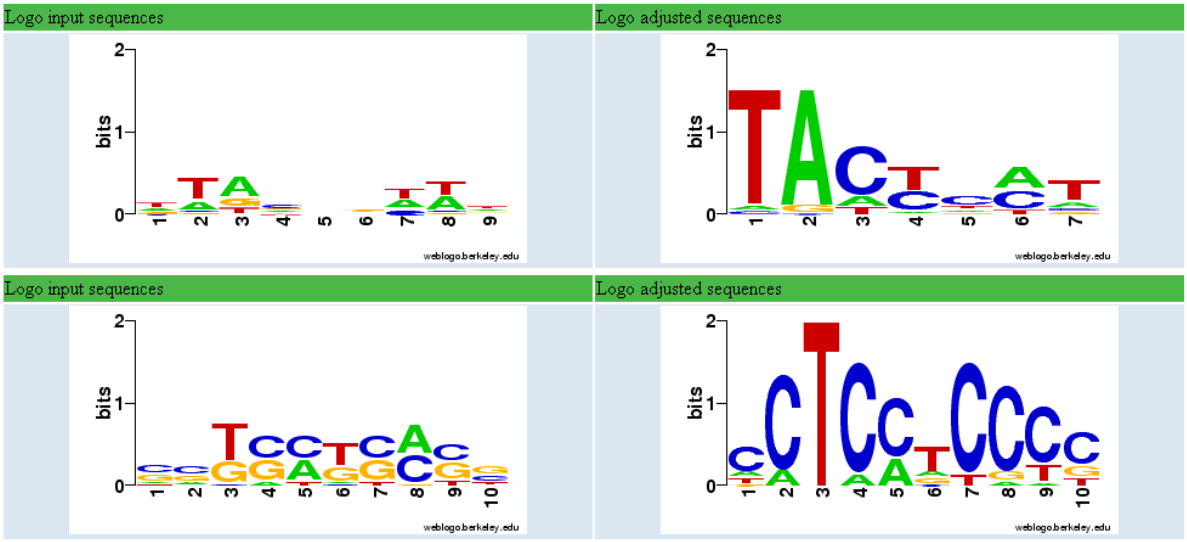

Figure 1: A comparison of the sequence logos constructed from the original database TFBMs (left side) and the adjusted TFBMs by using MoRAine (right side). The corresponding transcription factors are NarL (top) and MalT (bottom), both from *E. coli*. The 75 TFBMs for NarL and the 20 TFBMs for MalT have been extracted from RegulonDB. For NarL, we allowed to shift the motifs by at most one position to the left or to the right. The figure was taken as a screenshot from the MoRAine website.

of the mean information content for several values of l and r. We used 1165 TFBMs of 85 transcription factors of *Escherichia coli* obtained from RegulonDB. The results are summarized in Table 1. The combination ($cg/simC$) has the best runtime, but to gain the best information content improvement, one should use the combinations ($cg/simS$) or ($km/simS$). Additional file 1 (see Appendix) illustrates the relation between runtime and quality improvement for all combinations. In order to find good solutions within acceptable runtime, we recommend to use the combination $cg/simS$, which often provides the best improvement and still has an appropriate runtime.

	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 summarizes the information content improvement and the running times of MoRAine for different l- and r-values and all four search method/similarity function combinations.

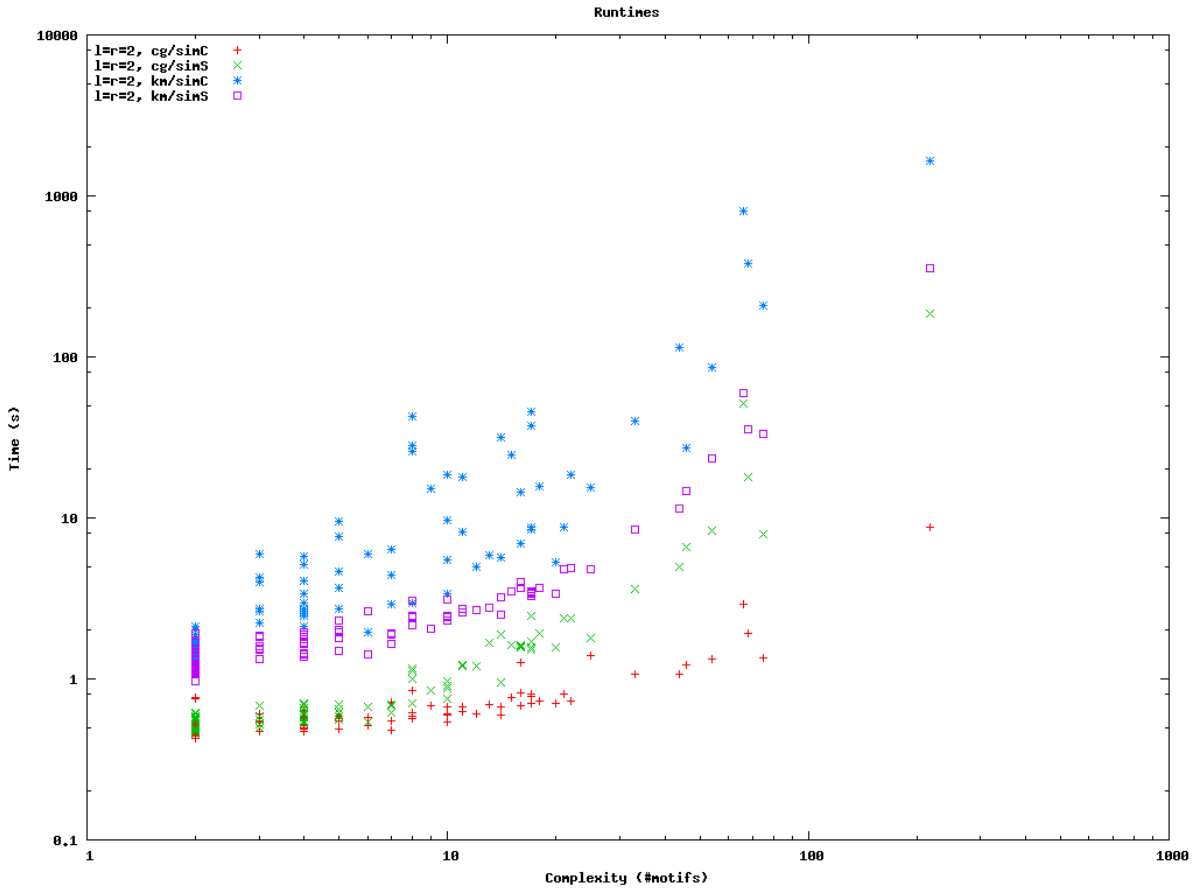

Figure 2: This plot illustrates the running times of MoRAine for different numbers of input TFBMs for $l = r = 2$. Note that both axes are log-scaled.

2.3 Adjusted TFBMs lead to better binding site predictions

The PFMs and PWMs derived from TFBMs are often used to predict further TFBMs in a given set of DNA sequences, generally in sequences upstream of putatively regulated target genes or operons. A PWM allows to assign a score to any length-m DNA sequence window. We say that a PWM matches such a window if the score exceeds a given threshold. A match is considered to be a good candidate for a real TFBM if we properly choose the score (generally as the log-odds score between the nucleotide distribution of true binding sites on the one hand and a background distribution on the other hand) and the threshold (ideally based on statistical considerations; see e.g. [22]). Different algorithms and implementations exist to perform these searches. Here we use the tool PoSSuMsearch [10] for further analyses. It uses lookahead scoring and it is based on efficiently searching an enhanced suffix array that previously has been created from upstream sequences of *E. coli*. The threshold for a match is automatically computed based on the tolerable frequency of hits in random sequences (p-value) by an efficient and exact lazy-evaluation method (for more details refer to [10]).

In the following, PoSSuMsearch is used to evaluate the prediction performance of (i) PWMs constructed from the original TFBMs extracted from the RegulonDB database and (ii) the MoRAine-adjusted PWMs. We show that by using MoRAine for preprocessing, the classification performance is significantly increased.

Datasets. We use the aforementioned 1165 extracted TFBMs for 85 transcription factors (TFs) from RegulonDB and construct 85 PWMs. Additionally, we obtained 3341 upstream sequences of all transcription units (TUs) of *E. coli* from CoryneRegNet. In CoryneRegNet, an upstream region is defined as that DNA sequence -560 to $+20$ bps upstream to the start codon of a TU (a gene, or an operon respectively). For every PWM we split these sequences into two sets: those with a known TFBM for the corresponding regulator (true positive) and those without a known TFBM, which we assume to be true negatives. This may be a debatable decision; however, the *E. coli* genome is one of the most well annotated genomes currently available.

Classification performance. For each PWM, both forward and reverse strand of upstream sequences are used to predict TFBMs with PoSSuMsearch, using different p-value thresholds. For each threshold, we measure the fraction of false positives (FP := number of incorrectly predicted motifs in relation to all predicted motifs), false negatives (FN := number of not predicted motifs in relation to the number of all motifs in the reference list), and the accuracy (ACC := number of correctly predicted motifs in relation to all motifs in the reference list).

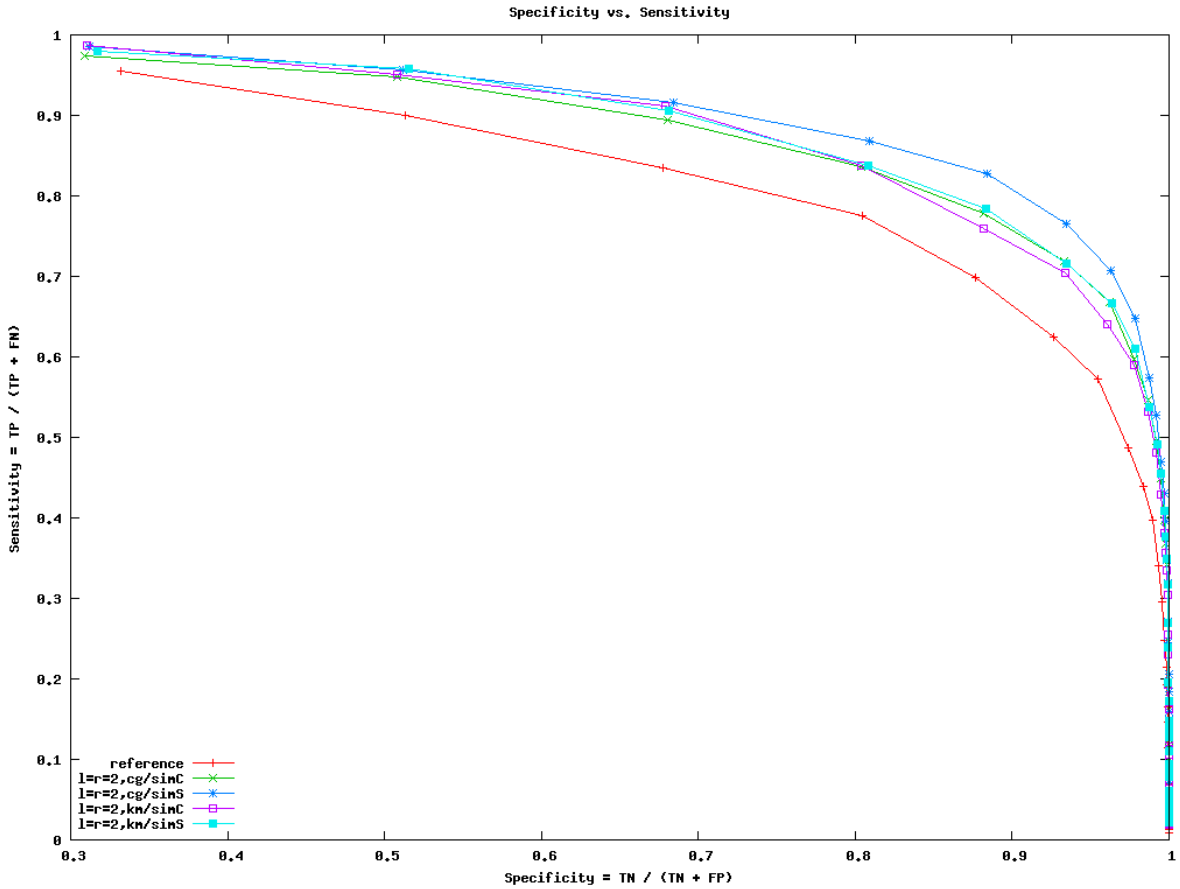

Figure 3: Specificity vs. sensitivity for varying p-value thresholds, for $l = r = 2$. For other values of l and r refer to Additional File 2 (see Appendix). For the reference curve we used original PFMs learned from original database TFBMs.

Figure 3 plots specificity vs. sensitivity for all PWMs adjusted with MoRAine for $l = r = 2$, for the four combinations of search algorithms and similarity functions, in comparison to specificity/sensitivity obtained with the original PWMs built from the original database TFBMs

(the reference curve). The measured values are plotted at different p-value thresholds. For a fixed specificity, the sensitivity obtained with adjusted PWMs is always higher than with original PWMs. The point where the sensitivity equals specificity is at ≈ 0.89 (adjusted PWMs) and ≈ 0.80 (original PWMs). Additional file 2 (see Appendix) plots the specificity/sensitivity for adjusted PWMs versus original PWMs for $0 \leq l = r \leq 7$. The prediction performance using adjusted PWMs always outperforms the reference. Generally the combination $(cg/simS)$ performs best. There is a visible jump in quality between $l = r = 0$ and $l = r = 1$. Increasing l and r further has smaller effects.

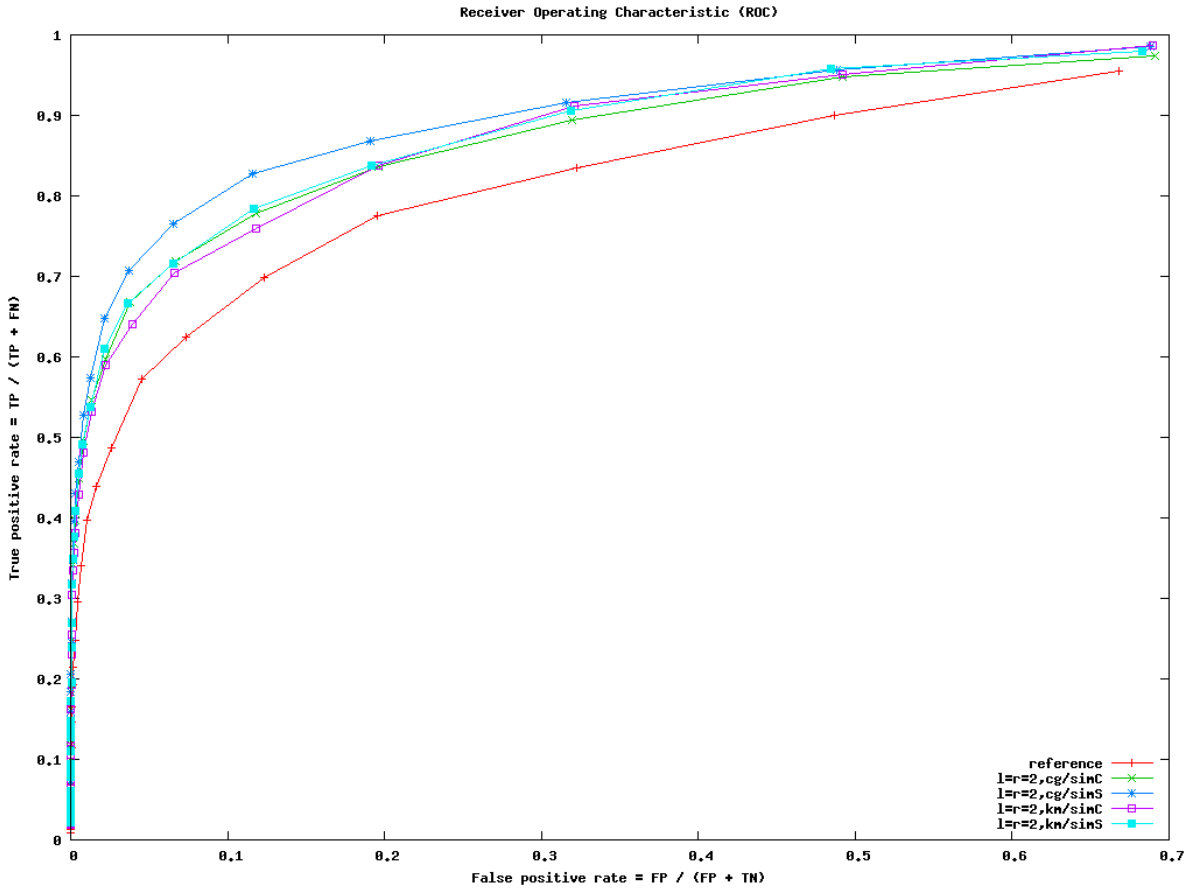

Figure 4: True positive (TP) vs. false positive (FP) rates for different p-value thresholds for $l = r = 2$. For other values of l and r refer to Additional File 3 (see Appendix). For the reference curve we used original PFMs learned from original database TFBMs.

Figure 4 shows a ROC plot: TP rate versus FP rate for $(l = r = 2)$. The plots show that predictions based on adjusted PWMs outperform those based on original PWMs. Again, one can see that the combination $(cg/simS)$ performs best. Additional file 3 (see Appendix) contains TP versus FP rates for $0 \leq l = r \leq 7$.

3 Conclusions

Gene regulatory protein-DNA interactions are stored in databases, such as RegulonDB, CoryneRegNet, PRODORIC, or TRANSFAC, along with annotated transcription factor binding sites, extracted from the scientific literature. Since the exact determination of the TFBM positions down

to one basepair is difficult and the annotation of the TFBM strands is sometimes neglected, and binding sequences are often stored $5' \rightarrow 3'$ relative to the target gene, TFBMs are frequently manually re-annotated (e.g. in PRODORIC for the regulators NarL and MalT in *E. coli*). This is both time-consuming and error-prone. For example, for $l = r = 0$ in ≈35% of all cases MoRAine suggests to switch the strand annotation.

It should be mentioned that the presented algorithms are heuristics selected for their good running time performance and scalability and do not guarantee an optimal solution in all cases. The observed increase in information content, however, suggests that we generally get useful re-annotations, and the speed of the algorithm allows it to be run on a non-dedicated web server.

Summarizing, this article introduces MoRAine, a software that supports the automatic re-annotation of TFBMs to increase the mean information content of a corresponding PFM. We provide a web server to facilitate using MoRAine and to compute sequence logos from transcription factor binding sites. We have demonstrated that a reliable strand annotation is necessary and helps to improve the PWM-based prediction performance. MoRAine-adjusted PWMs provide significantly more accurate classifications.

4 Methods

Recall that the goal is to find a set C of motifs that contains exactly one motif from each S_i and maximizes the information content of the corresponding frequency matrix F_C. Here, we describe two heuristic algorithms based on clustering to find such a motif set C.

4.1 Similarity Measures

The algorithms need a similarity function sim that measures the similarity between one motif and an existing cluster and thus helps to evaluate to which cluster of TFBMs a new TFBM is assigned. We use two different functions.

Motif-cluster similarity ($simC$) To measure the similarity between a single TFBM s and an existing non-empty cluster C', we calculate the man information content I for the frequency matrix constructed from all TFBMs of C' and s itself. We call this value the motif-cluster similarity $simC(s, C')$.

Motif-seed similarity ($simS$) Following another strategy, each cluster is represented by a single seed motif s'. Here we calculate I for the frequency matrix built from only the seed motif and the new TFBM s. We call this value the motif-seed similarity $\mathrm{simS}(s, s')$; it is faster to evaluate, but less accurate than $simC$.

These definitions apply only if the cluster C' to which a new motif s from a set S_i is to be assigned does not yet contain another motif from S_i. Otherwise, the similarity is set to $-\infty$; this ensures that each cluster contains only one motif from every set S_i.

4.2 Clustering strategies

The goal is to partition the set of motifs 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. We describe two clustering strategies.

Variant of k-means with random seeds (km) In this particular application, the number M of clusters is known; so we use a variation of the k-means algorithm [14]. In the end, we pick the cluster with the highest mean information content I:

We start with a random set of M (out of nM) motifs (the *seeds*) that form the initial clusters. Then, the following procedure is iterated until convergence: Each motif, in arbitrary but fixed order, is assigned to the cluster that maximizes the similarity ($simC$ or $simS$) value. This results in M clusters, each consisting of n motifs. A new seed sequence is chosen for each cluster as the sequence that best represents the cluster. This continues until no more changes occur for the seed sequence set; see Algorithm 1 for details. This strategy can be repeated for different initial seeds and addition orders.

Cluster growing (cg) Since each motif of each S_i must be in a different cluster, each S_i is used in turn as a set of initial seeds. Subsequently, the other motifs are added to their most similar cluster, similarly to the first iteration of the km algorithm, but this procedure is not iterated. Finally, the best solution obtained from the n different starting configurations is reported (see Algorithm 2 for details).

Note that both clustering strategies (km and cg) can be combined with both similarity functions ($simC$ and $simS$). The implications for running time and quality are discussed in the Results section.

Algorithm 1 Clustering with km

Input: sets S_i, $i = 1, \ldots, n$, with $|S_i| = M$; a similarity function sim
Output: Set C of motifs with high information content I

$oldseeds \leftarrow \{\}$
$seeds \leftarrow \{M$ arbitrary elements of $\bigcup_{i=1}^{n} S_i\}$
while $seeds \neq oldseeds$ **do**
    initialize clusters C_j, $j = 1, \ldots, M$, with one seed per cluster
    $oldseeds \leftarrow seeds$
    for $i \leftarrow 1$ to n **do**
        for all motifs s in S_i **do**
            assign s to cluster C_j with maximal $sim(s, C_j)$ over $j = 1, \ldots, M$
    $seeds \leftarrow \{\}$
    for all clusters C_j **do**
        find motif $s \in C_j$ with maximal $\sum_{s' \in C_j} sim(s, s')$
        add s to $seeds$
$C \leftarrow C_j$, with maximal $I(F_{C_j})$ over $j = 1, \ldots, M$
return $(C, I(F_C))$

Algorithm 2 Clustering with cg

Input: sets S_i, $i = 1, \ldots, n$, with $|S_i| = M$; a similarity function sim
Output: Set C of motifs with high information content I

$I_{best} \leftarrow 0$, $C_{best} \leftarrow \{\}$
for $i \leftarrow 1$ to n **do**
    $seeds \leftarrow S_i$
    initialize clusters C_j, $j = 1, \ldots, M$, with one seed per cluster
    for each $k \neq i$ **do**
        for all motifs s in S_k **do**
            assign s to C_j with maximal $sim(s, C_j)$ over $j = 1, \ldots, M$
    $C \leftarrow C_j$, with maximal $I(F_{C_j})$ over $j = 1, \ldots, M$
    if $I(F_C) \geq I_{best}$ **then**
        $I_{best} \leftarrow I$, $C_{best} \leftarrow C$
return (C_{best}, I_{best})

5 Acknowledgements

Richard Münch (TU Braunschweig) is gratefully acknowledged for helpful advice regarding the manual TFBM annotation of PRODORIC.

6 Appendix

6.1 Additional file 1

URL: `http://moraine.cebitec.uni-bielefeld.de/ib08/additional_file1_meanICdiff_vs_time.png`
Description: Average information content improvement plotted against the running time of MoRAine. The y-axis is log-scaled.

6.2 Additional file 2

URL: `http://moraine.cebitec.uni-bielefeld.de/ib08/additional_file2_sens_vs_spec_plots.png`
Description: Specificity vs. sensitivity for $0 \leq l = r \leq 7$. For the reference curve we used original PFMs derived from original database TFBMs.

6.3 Additional file 3

URL: `http://moraine.cebitec.uni-bielefeld.de/ib08/additional_file3_roc_plots.png`
Description: True positive (TP) vs. false positive (FP) rate for $0 \leq l = r \leq 7$. For the reference curve we used original PFMs derived from original database TFBMs

References

[1] PRODORIC sequence logo MalT. http://www.prodoric.de/matrix.php?matrix_acc=MX000139.

[2] PRODORIC sequence logo NarL. http://www.prodoric.de/matrix.php?matrix_acc=MX000003.

[3] 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.

[4] Babu MM and Teichmann SA. Evolution of transcription factors and the gene regulatory network in Escherichia coli. *Nucleic Acids Res*, 31(4):1234–1244, Feb 2003.

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

[6] Baumbach J. CoryneRegNet 4.0 - A reference database for corynebacterial gene regulatory networks. *BMC Bioinformatics*, 8(1):429, Nov 2007.

[7] 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.

[8] 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.

[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] 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.

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

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

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

[14] Hartigan JA. *Clustering Algorithms*. Wiley, 1975.

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

[16] Keilwagen J, Baumbach J, Kohl T, and Grosse I. Computational reassessment of prokaryotic transcription factor binding sites - An application to the nitrate regulator NarL of Escherichia coli. (in preparation).

[17] 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 Res*, 31(13):3576–3579, Jul 2003.

[18] Matys V, Kel-Margoulis OV, Fricke E, Liebich I, Land S, Barre-Dirrie A, Reuter I, Chekmenev D, Krull M, Hornischer K, Voss N, Stegmaier P, Lewicki-Potapov B, Saxel H, Kel AE, and Wingender E. TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res*, 34(Database issue):D108–D110, Jan 2006.

[19] Muench R, Hiller K, Barg H, Heldt D, Linz S, Wingender E, and Jahn D. PRODORIC: prokaryotic database of gene regulation. *Nucleic Acids Res*, 31(1):266–269, 2003.

[20] Muench 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.

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

[22] Rahmann S, Mueller T, and Vingron M. On the power of profiles for transcription factor binding site detection. *Statistical Applications in Genetics and Molecular Biology*, 2(1), 2003.

[23] 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.

[24] Salgado H, Gama-Castro S, Peralta-Gil M, Diaz-Peredo E, Sanchez-Solano F, Santos-Zavaleta A, Martinez-Flores I, Jimenez-Jacinto V, Bonavides-Martinez C, Segura-Salazar J, Martinez-Antonio A, and Collado-Vides J. RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Res*, 34(Database issue):D394–D397, Jan 2006.

[25] Stormo GD, Schneider TD, Gold L, and Ehrenfeucht A. Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli. *Nucleic Acids Res*, 10(9):2997–3011, May 1982.

[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] Teichmann SA and Babu MM. Gene regulatory network growth by duplication. *Nat Genet*, 36(5):492–496, May 2004.

[28] Wingender E. TRANSFAC, TRANSPATH and CYTOMER as starting points for an ontology of regulatory networks. *In Silico Biol*, 4(1):55–61, 2004.