info-gibbs: a motif discovery algorithm that directly optimizes information content during sampling

Matthieu Defrance\textsuperscript{1,*} and Jacques van Helden\textsuperscript{1,*}

\textsuperscript{1}Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe). Université Libre de Bruxelles CP 263. Campus Plaine, Boulevard du Triomphe. B-1050 Bruxelles

ABSTRACT

Motivation: Discovering cis-regulatory elements in genome sequence remains a challenging issue. Several methods rely on the optimization of some target scoring function. The information content or relative entropy of the motif has proven to be a good estimator of transcription factor DNA binding affinity. However, these information-based metrics are usually used as a posteriori statistics rather than during the motif search process itself.

Results: We introduce here info-gibbs, a Gibbs sampling algorithm that efficiently optimizes the information content or the log likelihood ratio of the motif while keeping computation time low. The method compares well with existing methods like MEME, BioProspector, Gibbs or GAME on both synthetic and biological data sets. Our study shows that motif discovery techniques can be enhanced by directly focusing the search on the motif information content or the motif log likelihood ratio.

Availability: http://rsat.ulb.ac.be/rsat/info-gibbs

Contact: defrance@scmbb.ulb.ac.be

1 INTRODUCTION

Gene expression is regulated at the transcriptional level by transcription factors that bind to DNA at specific locations. Several algorithmic approaches have been developed for \textit{de novo} identification of regulatory signals from a set of sequences. Motif discovery methods can be used to construct motifs that represent the specificity of the binding between a transcription factor (TF) and its binding sites (TFBSs).

Depending on the motif representation used, motif discovery methods can be divided into two broad categories: enumerative methods that select over-represented words (exact or degenerated), and heuristics that target the discovery of more complex motifs like position-specific scoring matrices (PSSMs). Among the first category of methods, motifs can be represented by words (van Helden \textit{et al.}, 1998), spaced words (van Helden \textit{et al.}, 2000) or words with multiple gaps and errors (i.e. with degenerated positions) (Pavesi \textit{et al.}, 2001; Sinha and Tompa, 2003, 2002). Considering the second category of methods, aiming at discovering PSSMs, many algorithms have been proposed. This includes the greedy algorithm consensus (Hertz \textit{et al.}, 1990), expectation maximization algorithms like MEME (Bailey and Elkan, 1994), and several algorithms based on a Gibbs sampling strategy: Gibbs (Lawrence \textit{et al.}, 1993; Neuwald \textit{et al.}, 1995; Liu \textit{et al.}, 1995), AlignACE (Roth \textit{et al.}, 1998; Hughes \textit{et al.}, 2000), MotifSampler (Thijs \textit{et al.}, 2002) or BioProspector (Liu \textit{et al.}, 2001). The two latter support higher order Markovian background models. More recently Shida (2006a,b) proposed a Gibbs sampling method that allows a variable stochastic factor (temperature) that enhances Gibbs sampling convergence speed.

Most of the methods that target PSSM motif discovery sort the predicted motifs by computing a posteriori some score such as information content (consensus, MotifSampler), log likelihood ratio (MotifSampler, Gibbs), E-value of the log-likelihood (MEME) or E-value of the information content (consensus).

The information content (Hertz and Stormo, 1999), also called relative entropy, presents the advantage of measuring both the specificity of a motif (low variability within each column) and its contrast relative to the background model. Information content has been claimed to be a good measure of DNA binding affinity (Stormo, 1998).

The program consensus (Hertz \textit{et al.}, 1990) optimizes the information content, but is sensitive to the order of incorporation of the sequences. Genetic algorithms like GAME that try to optimize directly this score have recently emerged (Wei and Jensen, 2006; Chan \textit{et al.}, 2008) but the time and memory complexity of genetic algorithms are higher than more specific algorithms like Gibbs sampling. Furthermore, they require to specify a set of parameters (probabilities of mutation and crossing over, population size, selection operator, . . . ) which are difficult to relate to properties of the input sequences and output motifs (size, number of sites, conservation, . . .).

The scoring function used to sample motifs during the discovery process strongly affects the resulting motifs. Jensen and co-workers emphasized the impact of the input parameters and the scoring functions on the quality of discovered motifs (Jensen \textit{et al.}, 2004). They implemented the software BioOptimizer (Jensen and Liu, 2004), which takes as input a motif returned by some pattern discovery algorithm (BioProspector, Consensus, AlignACE, MEME), and improves it by local optimization of a scoring function based on the log-posterior distribution.

In this paper, we present a motif finding algorithm called info-gibbs, that combines the qualities of Gibbs sampling (time and memory efficiency, interpretability of parameters) using as a scoring scheme either the information content (IC) or the log-likelihood ratio.
ratio (LLR) of the motif. The strategy is to directly compute the IC or LLR of the motif at each step of the sampling. Compared to existing methods, info-gibbs shows good performances in terms of computation time and prediction quality on both simulated and real data sets.

2 METHODS

Problem statement  Given a set of sequences \( \Phi = \{\phi_1, ..., \phi_s\} \) and a motif length \( l \), the problem can be defined as follows: find a set of sequence fragments (sites) \( W = \{w_1, w_2, ..., w_n\} \) that has maximal information content. When considering the search space as a set of potential sites \( S = \{s_1, ..., s_n\} \) (e.g. all allowed positions in sequences), the problem can also be viewed as finding a subset \( W = \{w_1, ..., w_n\} \) of \( S \) that has maximal information content. The search space can optionally be restricted by marking in sequences some positions as forbidden and discarding associated sites from \( S \) (e.g. repetitive elements, iterative masking).

Background model  An important issue when trying to discover short motifs in biological sequences is the choice of the background model, which can greatly influence the quality of the results. Higher order Markov sequence models can improve the detection of regulatory elements (Thijs et al., 2001). The algorithm described here supports Markov models of any order as background models. By default, background frequencies are estimated from the input sequences, but they can also be based on an external sequence set considered as reference (e.g. whole set of promotors of the considered organism).

Matrix representation  The count matrix \( N \) associated with a given motif is constructed from the aligned set of sequence fragments \( W \), by computing the number of occurrences \( N_{i,j} \) of each letter \( j \) of the alphabet \( A = \{A, C, G, T\} \) at each position \( i \) (from 1 to \( l \)). The count matrix \( N \) can be expressed as follows:

\[
N_{i,j}(W) = \sum_{k=1}^{n} \delta(w^{(i)}_k, A_j)
\]

where \( w^{(i)}_k \) is letter at position \( i \) in word \( w_k \) and

\[
\delta(w^{(i)}_k, A_j) = \begin{cases} 1 & \text{if } w^{(i)}_k = A_j \\ 0 & \text{otherwise} \end{cases}
\]

The PSSM \( M \) is derived from the count matrix \( N \) by computing, for each position of the alignment, the relative frequency of each letter. To circumvent the variance due to the small number of sites, the count matrix can be corrected by applying a simple and widely used additive smoothing technique: adding a pseudo-count. The PSSM \( M \) can be expressed as follows:

\[
M_{i,j} = \frac{N_{i,j} + B_j \times \epsilon}{n + \epsilon}
\]

where \( B_j \) is the background frequency of residue \( j \) and \( \epsilon \) is the pseudo-count which takes a Real positive value, usually chosen reasonably small with respect to the number of sites.

2.1 Evaluating motif conservation

Information content  A well established measure of motif conservation is the information content (IC) also known as relative entropy (Schneider et al., 1986; Hertz and Stormo, 1999). Given a PSSM \( M \) and a Bernoulli background model \( B \), the IC of \( M \) can be written as follows:

\[
IC_B(M, B) = \sum_{i=1}^{l} \sum_{j \in A} M_{i,j} \log \frac{M_{i,j}}{B_j}
\]

where \( M_{i,j} \) is the probability of the letter \( j \) at position \( i \) in \( M \) and \( B_j \) the probability of the same letter in the background model.

The IC can be extended to a more general formulation when the positions are not independent. This can be written as follows:

\[
IC(M, B) = \sum_{u \in A^l} P(u|M) \log \frac{P(u|M)}{P(u|B)}
\]

where \( A \) is the alphabet, \( l \) the length of the motif, \( P(u|M) \) the probability to generate the fragment \( u \) given the matrix \( M \) and \( P(u|B) \) the probability to generate the same fragment given the background model \( B \). When a Bernoulli background model is used, Equation 5 can be simplified to Equation 4. For Markov models of higher order, this formula can be rewritten (see supplementary material) and computed in acceptable time for the Markov orders typically used in practice (\( m \leq 5 \)).

Log likelihood ratio  The motif conservation can also be estimated by computing the log likelihood ratio (LLR), defined as the log of the ratio between the probability to generate the set of sites \( W \) given the matrix \( M \) and given the background model \( B \) (that can be a higher order Markov model), respectively. For a set of \( n \) sites \( W = \{w_1, ..., w_n\} \), the LLR can be written as follows:

\[
LLR(M, W, B) = \sum_{k=1}^{n} \log \frac{P(w_k|M)}{P(w_k|B)}
\]

where \( P(w_k|M) \) is the probability to generate the site \( w_k \) given the matrix \( M \) and \( P(w_k|B) \) the probability to generate the same site given the background model \( B \).

Relation between IC and LLR  When considering a Bernoulli background model and a matrix \( M \) constructed from \( W \) using a low (or a null) pseudo-count (i.e. no matrix smoothing), the LLR can be computed by multiplying the IC by the number of fragments that contribute to the motif. IC and LLR are not additive, however, when using higher order Markov background models and/or when the pseudo-count is significant the relation between IC and LLR is no longer linear and can lead to quite significant differences. This difference highlights the nature of these two evaluators. While IC depends on the matrix alone, the LLR directly depends on the sites used to build this matrix.

2.2 Sampling strategy

Given a set of potential sites \( S \) and a background model \( B \), the objective is to find the motif that maximizes some motif scoring function \( W_{\text{max}} = \arg \max_{W \in \Omega(S, B)} \Omega(W|S, B) \), where \( \Omega(W|S, B) \) can be either \( n \cdot IC \) (Equation 5) or \( LLR \) (Equation 6).

In typical situations, this objective is unreachable because the number of possible motifs increases exponentially with the number of potential sites. The ideal solution can thus be approached by sampling the distribution that we define as follows:

\[
P(W|S, B) \propto \frac{\Omega(W|S, B)}{Z_{W}}
\]

where \( Z \) is a normalization factor, and \( T \) a temperature parameter which is detailed in Section 3.3.

3 ALGORITHM

The Gibbs sampling strategy consists in trying to sample \( P(W|S, B) \) iteratively using the conditional probability \( P(w_k|W_{\text{prev}}) \), where \( W_{\text{prev}} \) represents the motif constructed without the fragment.
Table 1. Gibbs site sampling algorithm

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (init)</td>
<td>Choose at random a set of sites $W = {w_1, ..., w_n} \in S$</td>
</tr>
<tr>
<td>2.</td>
<td>Choose a site $w_k$ in $W$ and remove it from $W$</td>
</tr>
<tr>
<td>3. FOR EACH $w$ in $S$ DO</td>
<td>$P(w</td>
</tr>
<tr>
<td>4.</td>
<td>Choose a new site $w_i$ proportionally to $P(w_i</td>
</tr>
<tr>
<td>5.</td>
<td>REPEAT 2., 3., 4. UNTIL convergence</td>
</tr>
</tbody>
</table>

For a given iteration $t$, the new site $w_k^{(t+1)}$ is drawn according to:

$$w_k^{(t+1)} \propto P(w_k|W^{(t)})$$

The Gibbs sampling algorithm first initializes the motif $W^{(0)}$ by selecting a random set of $n$ sites of length $l$. The algorithm (Table 1) then iterates over the following steps: (i) the sampling step where the probability $P(w|W)$ of each possible site given the current motif is computed; (ii) the update step where a new motif is constructed by replacing a given site from the motif by a new site sampled proportionally to $P(w|W)$.

### 3.1 Sampling

In previous implementations of the Gibbs sampler, the probability to draw a site was proportional to the likelihood ratio of the site (site-wise likelihood ratio, $LR_w$).

$$P(w|W) \propto LR_w = \frac{P(w|M)}{P(w|B)}$$

Since IC and the LLR are usually considered as relevant for estimating the quality of the predicted motifs, our strategy is to directly use these statistics during the sampling phase itself rather than as an a posteriori criterion. Given a motif constructed from $W_{\overline{w}}$, choosing a new site $w$ requires to compute $IC(w \cup W_{\overline{w}}, B)$ or $LRR(w \cup W_{\overline{w}}, B)$ for each site in $S$. Compared to the classical site sampler, evaluating IC or LRR at each step of the algorithm can be time consuming, especially if the computation relies on the raw formula (Equation 4, Equation 5 and Equation 6), which needs to be computed for each possible site at each iteration.

However, the overall time performances can be greatly enhanced by rewriting IC and LRR to avoid re-computations. The principle is to construct for each iteration, a table that will store pre-computed partial contributions to IC or LRR. This technique can be applied to compute the IC and LRR when the background is modeled by a Bernoulli sequence. This technique can also be extended to the computation of LRR when the background is modeled by Markov chain but can not be used to compute the IC in the Markovian case (note that info-gibbs also supports direct update of IC under Markov models, but at a higher computation cost). We first show in the following sections how to efficiently compute the IC when the background is a Bernoulli sequence. We then show how to extend this technique to compute efficiently the LLR when the background is Markovian.

### 3.2 Shifting

The sampling algorithm can converge toward a motif that overlaps some actual motif, but shifted left-wise or right-wise by a few positions. To prevent staying blocked in this kind of near local optima, sites should be shifted altogether left or right by few positions. This is achieved by selecting after each iteration the motif resulting from the best shift (one position left, one position right or no shift).

### 3.3 Temperature

The temperature parameter $T$ defined in Equation (7) influences the explored space and the convergence speed of the algorithm. A low temperature ($T < 1.0$) allows a fast convergence but reduces the number of explored solutions. A high temperature ($T > 1.0$) on the other hand allows a wider exploration, but unfavorably affects the convergence speed. The convergence behaviors have been evaluated by measuring the evolution over iterations of the best motif found (i.e. maximal IC) for different values of the temperature parameter on synthetic data sets (see Section 4.1 for details on these data sets). The best results were obtained with the temperature set between 0.8 and 1.0, whereas at higher or lower temperatures, the information content remains at lower levels. Supplementary Figure S1 shows the averaged temporal evolution of the best IC reached for a set of fixed temperature values.

### 3.4 Minimal distance between sites

In order to avoid an attractive effect of self-overlapping motifs (e.g. GAGAGAGAGAGAGAGA), info-gibbs supports a minimal threshold $d_{min}$ on the distance $d(s_i, s_{i+1})$ between two successive
MotifSampler

MEME

Gibbs

BioProspector

AlignAce

0.50

0.75

1.00

HIGH CONSERVATION D=0

AlignAce

BioProspector

GAME

Gibbs

info-gibbs

MEME

MotifSampler

0.352

0.615

0.265

0

0.25

0.50

0.75

1.00

AlignAce BioProspector GAME Gibbs info-gibbs MEME MotifSampler

PC

Sn

PPV

Fig. 1. Software performances on synthetic data. For each software the sensitivity $Sn$ (black bars), the positive predictive value $PPV$ (gray bars) and the performance coefficient $PC$ (line) are shown.

3.5 Multiple runs

Due to its stochastic nature, different runs of the algorithm on the same sequences are expected to return distinct results. As for other software, info-gibbs can automatically manage multiple runs using different starting points (i.e. starting from different random motifs). In this case the sampling algorithm is started from $k$ different points and the motif achieving the highest information content is kept.

3.6 Finding multiple motifs

The site sampling algorithm searches for only one motif at a time. As in other implementations, info-gibbs can discover multiple motifs by doing several runs and iteratively mask (i.e. remove from searched sites) sites already included in previously returned motifs.

3.7 Implementation and a availability

info-gibbs is implemented in ANSI C++ and the command line tool is available upon request. A web interface to info-gibbs is also provided via the RSAT web server (http://rsat.ulb.ac.be/rsat/).

4 RESULTS

4.1 Evaluation on synthetic data

The first evaluation was carried out on a well-defined problem: finding synthetic sites implanted in synthetic sequences. One of the main advantages of this approach is that it provides a well-controlled environment, where all the motif occurrences (sites) are known. Consequently, usual evaluation statistics such as sensitivity and positive predictive value can be accurately computed to estimate algorithm performances. This contrasts with real data sets, where several effective binding sites are likely to have escaped experimental detection, and thus be missing in the annotations. Another advantage of that evaluation is its repeatability and scalability. Since the sequences and the motif itself are randomly generated, the evaluation can be repeated sufficiently to provide accurate measurements.

The data sets are generated in 3 steps: (i) random sequence generation (random-seq); (ii) random motif generation (random-motif); (iii) implantation of the generated motifs in the sequences (implant-sites). The tools used to generate those synthetic data sets have been integrated in the Regulatory Sequences Analysis Tools (van Helden, 2003; Thomas-Chollier et al., 2008).

Generating random sequences In order to generate random sequences that maintain some characteristics of real sequences, we chose to use a Markovian background model of order 4 trained on all upstream sequences of the yeast Saccharomyces cerevisiae. In the case of the yeast genome, Markovian background models of order 3 and 4 seem to give the best results for motif discovery software (Thijs et al., 2001). The background model was chosen to maximize the performances of the software that take advantage of higher order background models. This Markov model was used with random-seq to generate 100 sets of 10 sequences of 500bp.

Generating random motifs The program random-motif generates random motifs of a given length, with a controllable level of conservation ($c$). For each position of the motif a residue is chosen randomly, and its probability is set to $c$. The remaining probabilities $(1-c)$ are shared in an equiprobable way between the other residues.

For our first test, we generated motifs of length 8 where the probabilities were set to 0.91 for the dominant nucleotide and 0.03 for the others, according to Wei and Jensen (2006).

Implanting motif instances Random sites were generated and implanted in sequences using the program implant-sites. For each possible position ($u = L - l + 1$) of each sequence, a motif can be implanted according to a probability $p$. This model assumes no dependencies between site positions. The probability to implant $k$ motif occurrences in a sequence is given by the binomial formula: $P(occ = k) = \binom{u}{k} p^k (1 - p)^{u-k}$. In our conditions, the average expectation was set to 1 occurrence per sequence $p = \frac{1}{u}$.

Motif discovery We tested the capability of info-gibbs and 6 other motif discovery software to detect the implanted motifs (AlignACE, BioProspector, GAME, Gibbs, MEME, MotifSampler). For each tool, the motif length was set to 8 and the expected number of occurrences to 10 (corresponding to one expected occurrence per sequence). When a higher order background model was allowed by the software (BioProspector, info-gibbs, MEME, MotifSampler), we provided the background model used to generate the sequences as input. In other cases only nucleotide frequencies were provided. Since most of the compared software use multiple starting points (initial alignments) by default or have an option to allow multiple starting points we set this parameter to at least 10 runs (when this option was available) and considered only the best motifs predicted among those runs. As the motifs were implanted on the same strand, the parameter to search only on one strand was set when available (all the software except AlignACE).
The other parameters were left to their defaults. For info-gibbs, the sampling strategy relies on the optimization of the LLR which is for time efficiency reason the default strategy.

All the software used in this study predict a motif by aligning a set of sites extracted from the input sequences. In order to evaluate the predictions, we compared the sites discovered with those implanted in the sequences. Implanted sites are qualified as True Positives (TP) when they are also predicted sites and False Negative (FN) otherwise. Predicted sites that do not correspond to implanted sites are qualified as False Positives (FP).

Three classical evaluation statistics were computed: (i) the positive predictive value \( PPV = TP/(TP + FP) \) which quantifies the proportion of correct sites among the predicted ones; (ii) the sensitivity \( Sn = TP/(TP + FN) \) which quantifies the proportion of implanted sites that were covered by the predicted ones; (iii) the performance coefficient \( PC = TP/(TP + FN + FP) \) which captures both the positive predictive value and the sensitivity (Pevzner and Sze, 2000; Tompa et al., 2005).

The comparison of the motif discovery software performances (Figure 1) shows that, under our conditions, info-gibbs and MEME supersede all the other tools. The third best result is obtained by BioProspector, which achieves a \( Sn \) similar to info-gibbs and MEME, but has a lower \( PPV \) and \( PC \).

In order to assess the capability of the motif discovery software to detect motifs with a higher variability, we generated as previously synthetic data sets (100 sets of 10 sequences of 500bp) with implanted motifs whose degree of conservation \((c)\) varies from 0.95 to 0.7. Figure 2 shows that the performances of all programs decrease rapidly when motifs are less conserved, and fail to detect motifs below \( c = 0.7 \) (Numerical values are reported in supplementary Table S1). The ranking of the software does not depend on the degree of conservation \((c)\) of the implanted motif. info-gibbs and MEME perform better than the other tools over the whole range of conservation tested.

The performances on synthetic data are however strongly influenced by somewhat arbitrary choices (background model, site density, \ldots). In order to evaluate the software tools in realistic conditions, we also tested their performances on biological data sets (Section 4.2 and Section 4.3).

**Comparison between scoring methods** We evaluated on synthetic data the impact of three scoring methods used for motif updating (as defined in Section 3.1): (i) classical site-wise likelihood ratio \( LR_{w} \); (ii) motif-wise LLR; (iii) motif-wise IC.

Synthetic motifs with various degree of conservation (0.91, 0.85 and 0.75) were generated and implanted (1 expected site per sequence) in random sequences (100 sets of 10 sequences of 200bp). The sequences were generated according to a Markov model of order 2 trained on yeast upstream non-coding sequences.

In order to evaluate the influence of the background model in the motif discovery process both Bernoulli background models directly estimated by info-gibbs from the input sequences and pre-computed higher order Markovian models were used. The influence of the pseudo-count used to compute the motif score (as defined in Equation 3) was also evaluated by comparing the predictions made using 5 levels of pseudo-count (0.1, 0.5, 1.0, 2.0 and 4.0).

In each case, predicted motifs were assessed by comparing the positions of the implanted sites with those of the sites included in the discovered motif and by computing as previously the performance coefficient \((PC)\). The results are show in Figure 3.

In this evaluation the differences between the scoring methods do not seem to be influenced by the background model (first and second column of Figure 3) even if all the strategies tend to give slightly better results when higher order background models are used. In contrast, the pseudo-count parameter influences deeply the results. Around or lower than the commonly used value (pseudo-count \( = 1.0 \)) the performances of the motif-wise IC and LLR strategies are similar and always better than the site-wise likelihood ratio. At unusual levels of pseudo-count (pseudo-count \( = 4.0 \)) only the motif-wise LLR strategy remains efficient.

### 4.2 Evaluation on RegulonDB data

The second evaluation focused on transcriptional regulation of *Escherichia coli* K12. Regulons (i.e. sets of genes regulated by the same transcription factor) and associated transcription factor binding sites where retrieved from the RegulonDB 6.0 database (Gama-Castro et al., 2008), which provides a comprehensive set of curated data for the *Escherichia coli* K12 transcriptional regulatory network.

Starting from the binding sites table, we selected all the transcription factors for which RegulonDB contains an annotated PSSM. This provides a reference set of 32 regulons (supplementary Table S3) with good annotations about target genes, binding sites and regulatory motifs (PSSMs). For each regulon, we retrieved sequences upstream of the start codon up to the next gene, with a maximal length of 500bp.

Motif discovery software were run on each regulon with motif length fixed to 16 and expected number of occurrences set to 2 sites per sequences. Like the previous analysis we set the number of runs to at least 10 (when this option was available) and considered the best motifs predicted. For all the software the parameters were set to search on both strands. All other parameters were left to

![Fig. 2. Software performances on synthetic data. For each software and a level of motif conservation ranging from 0.95 to 0.70, the performance coefficient \(PC\) is reported.](http://bioinformatics.oxfordjournals.org/Downloaded from http://bioinformatics.oxfordjournals.org/)}
their defaults and no background model were provided as input to the software excepting the G+C frequency for programs that require it (AlignACE). Under these conditions most of the software compute a Bernoulli background model and the right column shows the results when using a Markovian background model of order 1 (order 2 gives similar results).

The asymptotic covariance was computed using the software sstat (Pape et al., 2008) with an upper threshold set to 0.001 for the alpha value (corresponding to a random expectation of 1 prediction per 1000bp). We normalized the asymptotic covariance by dividing it by the maximal possible value, obtained by comparing the annotated motif with itself. Under these conditions, a predicted motif very close to the annotated one will give a score close to 1.0 while a prediction very distant from the annotation will result in a score close to 0.

Since most of the evaluated software are based on stochastic algorithms (all programs except MEME), we averaged the asymptotic covariance over 100 independent trials over each data set. As a negative control, we also compared the annotated motifs to random predictions (referred as random), obtained by running info-gibbs with 0 iterations (See supplementary Table S4 for the full results).

To summarize the results we computed the number of predicted motifs that were close to the annotated ones by counting the number of regurlons for which the asymptotic covariance were greater than or equal to thresholds ranging from 0.6 to 0.9 (Figure 4). In this study info-gibbs shows better performances than other tools and was the only software to predict motifs that were very close to the annotated one (asymptotic covariance ≥ 0.9).

We discuss hereafter 3 selected regulons which illustrate the diversity among software predictions, and permit to highlight some strengths and weaknesses of info-gibbs. The inverse cumulative distribution of the asymptotic covariance between annotated and predicted regulatory motifs for the PurR, DnaA and H-NS regulons are shown in Figure 5. The histograms of the asymptotic covariance distribution are also shown in supplementary Figure S2 while the mean, the first, second and third quartile (i.e. 0.25 quantile, median and 0.75 quantile) are reported in supplementary Table S2.

Due to the stochastic nature of the algorithms, we found it essential to base our evaluation on the analysis of the whole distribution of asymptotic covariance values, rather than reducing the results to some summary statistics as the mean or the median.
Indeed, the inverse cumulative distribution and the histograms indicate not only the central tendency, but also the dispersion of the performances across various runs of the programs.

The first example, corresponding to 19 genes regulated by the PurR transcription factor, is typical of a well-conserved motif that is easily identified by most software. The full distribution however shows interesting differences between the respective performances. Methods based on information content (GAME, info-gibbs) return in most cases motifs that are very close to the annotated ones (First quartile > 0.9 indicating that 75% of the predictions reach an asymptotic covariance of at least 0.9). Three other programs (MEME, BioProspector, and MotifSampler) perform similarly well. Alignace gives lower results, yet significantly better than random predictions. The histograms of Gibbs and AlignAce show a wide dispersion, suggesting a poor capability to converge towards reproducible results.

The second example, related to the DnaA transcription factor, shows very contrasted results. Most of the software except info-gibbs predict motifs that are very far from the annotated one, and their performances are close to random predictions. Obviously, the dispersion is also very high for all the stochastic software (i.e. not MEME).

Fig. 5. Inverse cumulative distribution of the asymptotic covariance between annotated and predicted regulatory motifs for Escherichia coli K12 PurR, DnaA and H-NS regulons for each software. The random behaviors are represented by the dashed lines.

The last example highlights the difficulty to find the poorly conserved H-NS motif. As shown in the logo representation (Schneider and Stephens, 1990; Crooks et al., 2004) of the annotated motif (Figure 5), only one position in the middle of motif is well conserved. Given the poor information content of this annotated motif, it is not surprising that info-gibbs fails to identify it. However, all the other software also fail to return relevant results. Actually, the distributions are close to the random motif selection for all software.

4.3 Analysis of yeast ChIP-chip data sets

The third evaluation focused on Harbison et al. (2004) ChIP-chip data. We discovered motifs in promoters of 29 sets of target genes detected by ChIP-chip experiments for 17 yeast transcription factors (some factors were tested in various culture media). Among the clusters found in the Harbison experiments, we selected all the transcription factors for which a matrix was annotated in the SCPD database (Zhu and Zhang, 1999) and for which Harbison et al. (2004) reported at least 5 target genes. This test corresponds to realistic conditions, since those data sets are known to be more noisy, each set of genes likely contains a mixture of effective target genes and of false positives. In addition, some factors were tested in various culture media, which may affect their activity, and thereby the preferential binding to their target promoters.

For each set of target genes, we retrieved sequences upstream of the start codon up to the next gene, with a maximal length of 800bp.

Motif discovery was performed on each regulon with motif length fixed to 16 and expected number of occurrences set to 2. Since the differences between Bernoulli background models and higher order models are significant with yeast genomes, we have chosen for this data set the background model that gives the best results for all the software that can use higher order background models. We provided to those software a Markov model of order 3 trained on all yeast promoters and a Bernoulli model trained on the same set to the others. For all the software the parameters were set to search on both strands. All other parameters were left to their defaults. Discovered motifs were evaluated by computing their averaged asymptotic covariance (over 10 trials) with the motifs annotated in SCPD (Supplementary Table S5).

All programs show contrasted performances (Figure 6): they correctly detect between 0 and 8 motifs among the 29 data sets. On these data sets, info-gibbs and MotifSampler supersede the other tools and all programs supporting higher-order Markov models show higher performances (BioProspector, info-gibbs, MEME and MotifSampler).

5 DISCUSSION AND CONCLUSION

We introduced here a new stochastic motif finding algorithm that directly optimizes information content or log-likelihood ratio rather than computing it a posteriori. This approach showed interesting results on both simulated and biological data: info-gibbs performs equally to or better than all the other methods tested here on the synthetic data sets, on bacterial regulons and on yeast ChIP-chip data.

Assessing the quality of motif discovery is a difficult task, for several reasons. In contrast with protein structure, we do not possess
a data set that could be considered as the target to which predictions can be compared. One problem faced by Tompa et al. (2005) in their community-based assessment of 13 motif discovery programs was the definition of suitable testing sets. Their data set contained a mixture of biological and synthetic sequences, with implanted or native transcription factor binding sites. We took inspiration from this experience, but adopted a different strategy to face the lack of annotations, by separating the evaluation in two independent tests: (i) synthetic motifs implanted in synthetic sequences, with an evaluation based on site comparisons; (ii) biological sequences with their native sites, with a motif-based rather than site-based evaluation. Another originality of our evaluation is to directly address the intrinsic lack of reproducibility of stochastic approaches, by analyzing the distribution of matching statistics rather than choosing the average performance.

The most important limitation of the proposed approach, which is shared with some of the other methods, is that the user has to specify input parameters which have a strong impact on the result, in particular the motif length and the expected number of occurrences. One strategy (supported by MEME) is to test various lengths and select the motif that optimizes some a posteriori statistics (e.g. information per column). The expected number of occurrences is also problem-dependent and also need to be optimized (Shida, 2006a). Despite these limitations, the overall performances of info-gibbs are promising. Furthermore, the tool is integrated in the extensive software suite RSAT, which provides a flexible and powerful environment for deciphering the regulatory elements from genome sequences.

FUNDING
This was supported by the the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, project P6/25 (BioMaNet), who funded the postdoc grant of MD. The BiGRe laboratory is member of the BioSapiens Network of Excellence funded under the sixth Framework program of the European Communities (LSHG-CT-2003-503265).

REFERENCES


