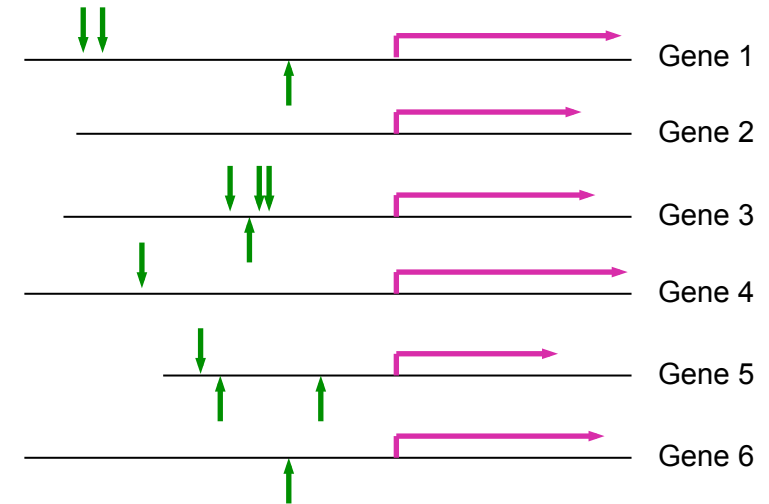


Matrix-based pattern discovery algorithms

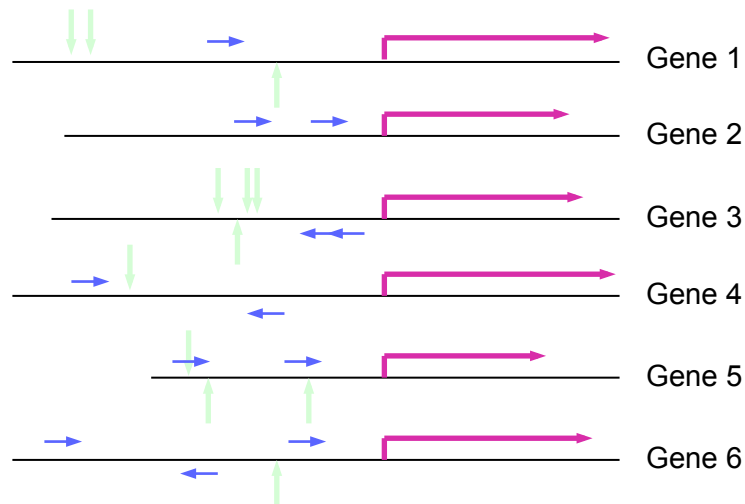
Pattern discovery situation

- Question
 - We dispose of a set of s promoter sequences containing x instances (sites) of a cis-regulatory motif.
 - Starting from the sequences, we want to discover the motif.
- Hypothesis
 - The motif is over-represented in the sequence set.



Building a matrix from an arbitrary set of sites

- A simple procedure
 - Select x sites of length w in the input sequences.
 - Align them at their first position.
 - Build a matrix.
- If the sites are selected at random, the motif is likely to be non informative.



Site sequences

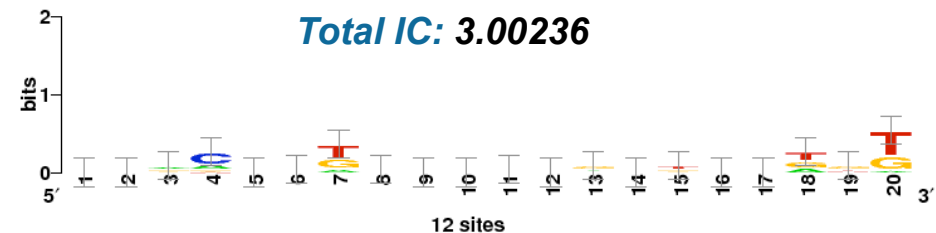
```

→ G C A C C A T C C G T T T A G A C T T A
→ C G G C G A T G C T G A G G T T G G A T
→ G A A A C A G A A T A C G C G A G T G T
→ C T T C T A G G G T C C T T T T G G G T
→ C G A C T C A G A G G A A T T C A T C G
→ A T A C A A G T G C A G A C A C T A C G
→ A C C T C G A T G A G A C T T C T A G T
→ G G T G A C T A C C C T G G G G A T T T
→ T C A C A T T T A C A G G G T A A G G T
→ G G G A A C G T T A G A G T C T C T T T
→ A T A A G C T G C A A T G T T A T G G G
→ C T G C T G T T T G G G A C C C G A G G
    
```

Matrix built from those sites

A	3	1	6	3	4	5	2	2	3	3	4	4	3	1	1	4	3	3	1	1
C	4	3	1	7	3	4	0	1	4	3	2	2	1	3	2	4	2	0	2	0
G	4	4	3	1	2	2	4	4	3	3	5	3	6	3	3	1	4	4	6	4
T	1	4	2	1	3	1	6	5	2	3	1	3	2	5	6	3	3	5	3	7

Total IC: 3.00236

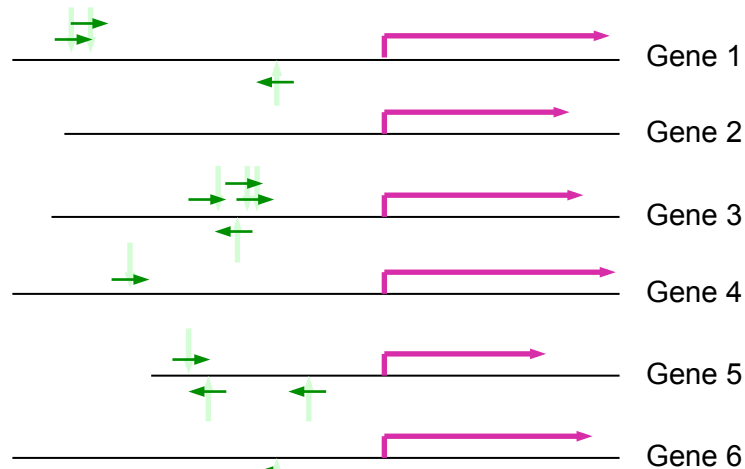


Building a matrix with related sites

- A simple procedure
 - Select x sites of length w in the input sequences.
 - Align them at their first position.
 - Build a matrix.
- If the matrix is build with the « correct » sites, we expect to observe a high information content.

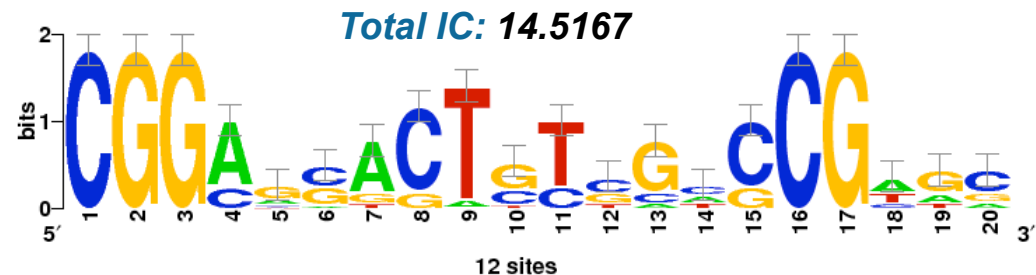
Site sequences

→	C	G	G	C	G	C	A	C	T	C	T	C	G	C	C	C	G	A	A	C
→	C	G	G	A	G	G	G	C	T	G	T	C	G	C	C	C	G	C	T	C
→	C	G	G	A	G	G	G	C	T	G	T	C	G	C	C	C	G	C	T	C
→	C	G	G	A	G	C	A	G	T	G	C	G	G	C	G	C	G	A	G	G
→	C	G	G	A	G	C	A	G	T	G	C	G	G	C	G	C	G	A	G	G
→	C	G	G	A	A	G	A	C	T	C	T	C	C	T	C	C	G	T	G	C
→	C	G	G	A	A	G	A	C	T	C	T	C	C	T	C	C	G	T	G	C
→	C	G	G	A	G	C	A	C	T	G	T	T	G	A	G	C	G	A	A	G
→	C	G	G	C	G	G	T	C	T	T	T	C	G	T	C	C	G	T	G	C
→	C	G	G	C	A	C	A	C	A	G	T	G	G	A	C	C	G	A	A	C
→	C	G	G	A	C	A	A	C	T	G	T	T	G	A	C	C	G	T	G	A
→	C	G	G	A	T	C	A	C	T	C	C	G	A	A	C	C	G	A	G	A



Matrix built from those sites

A	0	0	0	9	3	1	9	0	1	0	0	0	1	4	0	0	0	6	3	2
C	12	0	0	3	1	6	0	10	0	4	3	6	2	5	9	12	0	2	0	7
G	0	12	12	0	7	5	2	2	0	7	0	4	9	0	3	0	12	0	7	3
T	0	0	0	0	1	0	1	0	11	1	9	2	0	3	0	0	0	4	2	0



Finding the optimal matrix – a straightforward algorithm

- A straightforward approach
 - Test all possible motifs that can be built from the sequence by aligning x sequence fragments of length w .
 - Compute a score (e.g. information content, log-likelihood, P-value) associated to each motif.
 - Report the highest-scoring motif.
- Is this approach tractable ?
 - We will estimate its complexity in the next slides ...

Pattern discovery: typical dimensionality for a very small dataset

- Typical case 1: GAL genes
 - s 6 sequences (promoters of the annotated GAL genes)
 - L average promoter size (yeast) 500 bp
 - sps expected sites per sequences: 2 (multiple sites are frequent in yeast)
 - x expected number of sites: $sps \cdot s = 12$
 - w matrix width = 20
- Let us assume that
 - A signal can be found on any strand of any sequence
 - Number of possible site positions: $n = 2s(L - w + 1) = 5772$
 - Each sequence contains 0 or several occurrences -> the number of possible alignments equals the number of ways to choose 12 among the 5772 possible sites.

$$N_{alignments} = C_n^x = C_{2s(L-w+1)}^x = C_{5772}^{12} = 2.9 \cdot 10^{36}$$

Pattern discovery: typical dimensionality for a reasonable dataset

- **Typical case 2:** yeast promoters bound by a TF in a ChIP-chip experiment (e.g. Harbison et al. 2004)
 - s 50 sequences
 - L average promoter size (yeast) 500 bp
 - sps expected sites per sequences: 2 (multiple sites are frequent in yeast)
 - occ_e expected sites: $sps*s=100$
 - w matrix width = 20
- Let us assume that
 - A signal can be found on any strand of any sequence
 - Number of possible site positions: $n=2s(L-w+1)=48100$
 - Each sequence contains 0 or several occurrences -> the number of possible alignments equals the number of ways to choose 100 among the 48100 possible sites.

$$N_{alignments} = C_{2s(L-w+1)}^{occ_e} = C_{48100}^{100} \approx \infty$$

sites	positions	alignments
1	962	4.62E+05
2	1924	5.69E+11
3	2886	7.98E+17
4	3848	1.18E+24
5	4810	1.81E+30
6	5772	2.82E+36
7	6734	4.46E+42
8	7696	7.13E+48
9	8658	1.15E+55
10	9620	1.86E+61
15	14430	2.19E+92
20	19240	2.75E+123
25	24050	3.55E+154
30	28860	4.69E+185
35	33670	6.27E+216
40	38480	8.48E+247
45	43290	1.16E+279
50	48100	Inf

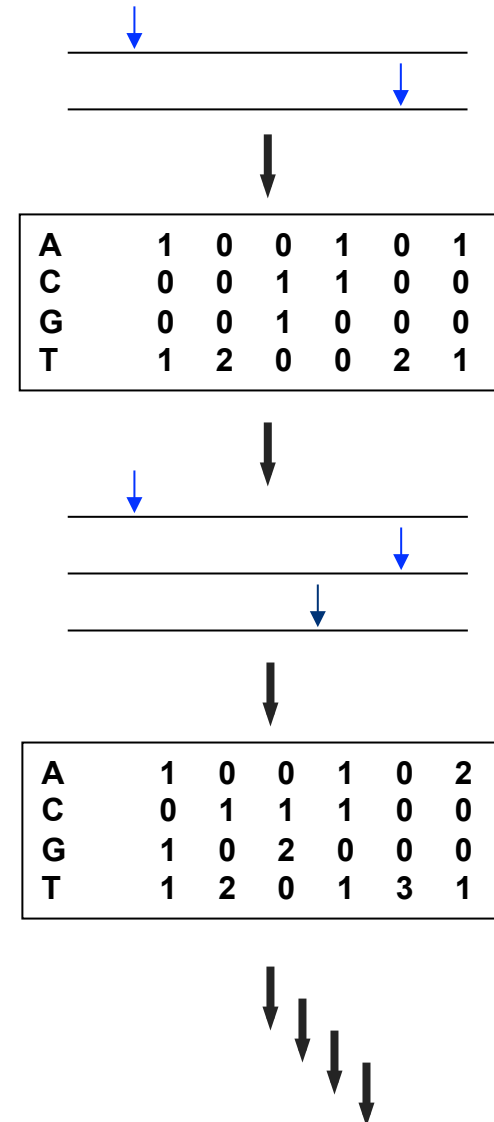
Matrix-based pattern discovery

- General problem
 - The number of possible matrices is too large to be tractable
- Approaches
 - Define heuristics to extract a matrix with highest possible information content (lowest probability to be due to random effect) → optimization techniques
- Approaches working with regulatory sequences
 - Greedy algorithm (consensus)
 - Expectation-maximization (MEME)
 - Gibbs sampling (gibbs, AlignACE, BioProspector, MotifSampler, info-gibbs, ...)

The greedy algorithm “consensus”

Pattern discovery: greedy algorithm (consensus, by Jerry Hertz)

1. Create all possible matrices with two sites taken from the two first sequences ($n*n$ possibilities).
 - Typically $1000*1000=1.000.000$ possible matrices, each made of 2 sites.
2. Retain the most informative matrices only
 - E.g. the 1000 matrices with the highest information content
3. Create all possible combinations between each of these matrix and each possible site in the next sequence.
 - Typically 1000 previous matrices x 1000 new sites.
4. Iterate from previous steps until all sequences are incorporated
5. Return the most significant matrices



Greedy algorithm: summary

■ Strengths

- Time increases linearly with the size of the sequence set.
- Direct optimization of the information content, which is generally a relevant criterion for estimating the relevance of a motif.

■ Weaknesses

- Sensitive to sequence ordering in the input data set.
- Returns multiple matrices, but they are generally slight variants of the same pattern.
- Cannot deal with higher-order Markov models.

■ References

- Hertz et al. (1990). Comput Appl Biosci 6(2), 81-92.
- Hertz, G. Z. & Stormo, G. D. (1999). Bioinformatics 15(7-8), 563-77.
- Stormo, G. D. & Hartzell, G. W. d. (1989). Proc Natl Acad Sci U S A 86(4), 1183-7.

Expectation- Maximization (EM)

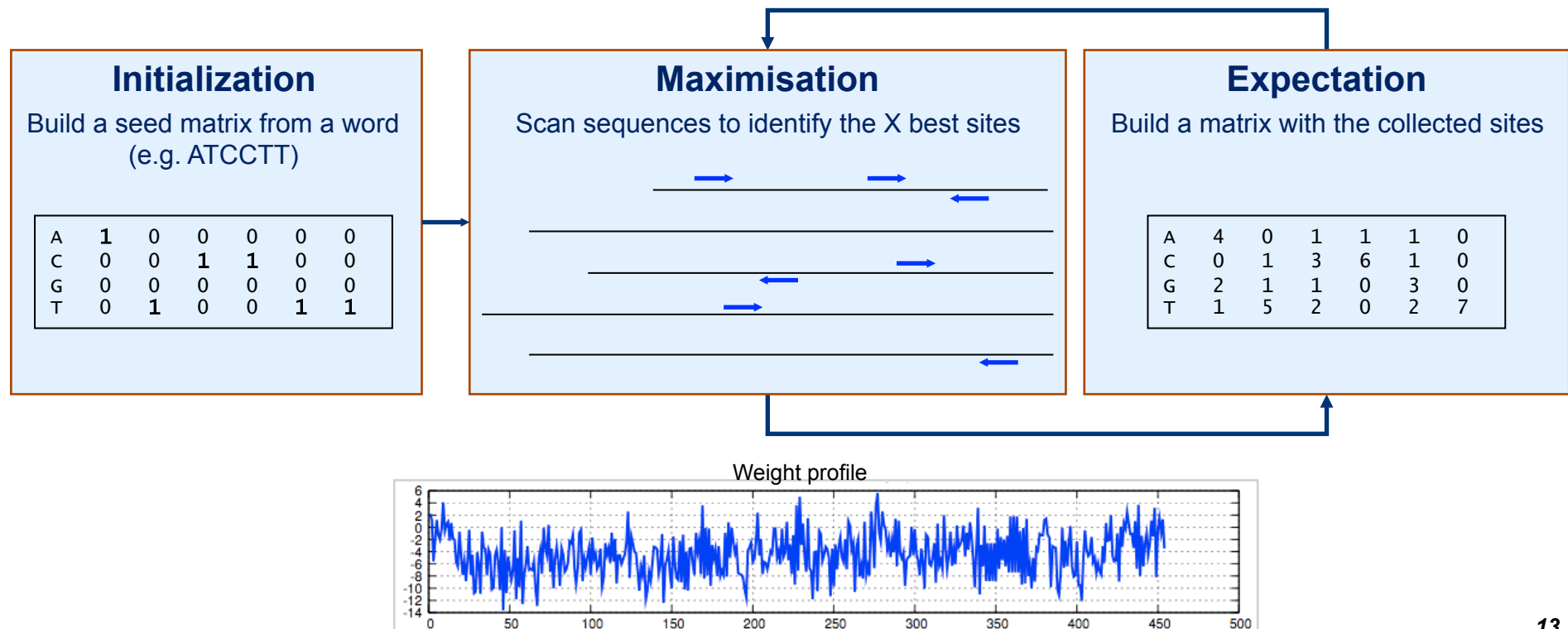
MEME - Multiple EM for Motif Elicitation

■ EM

- Instantiate a “seed” motif
 - Maximization: select the X highest scoring sites
 - Expectation: build a new matrix from the collected sites

■ Multiple EM

- Iterate over each k-mer found in the input set
 - building a matrix from the k-mer
 - Run an EM (expectation/maximisation) algorithm to optimize the matrix.
- Return the highest scoring matrices.

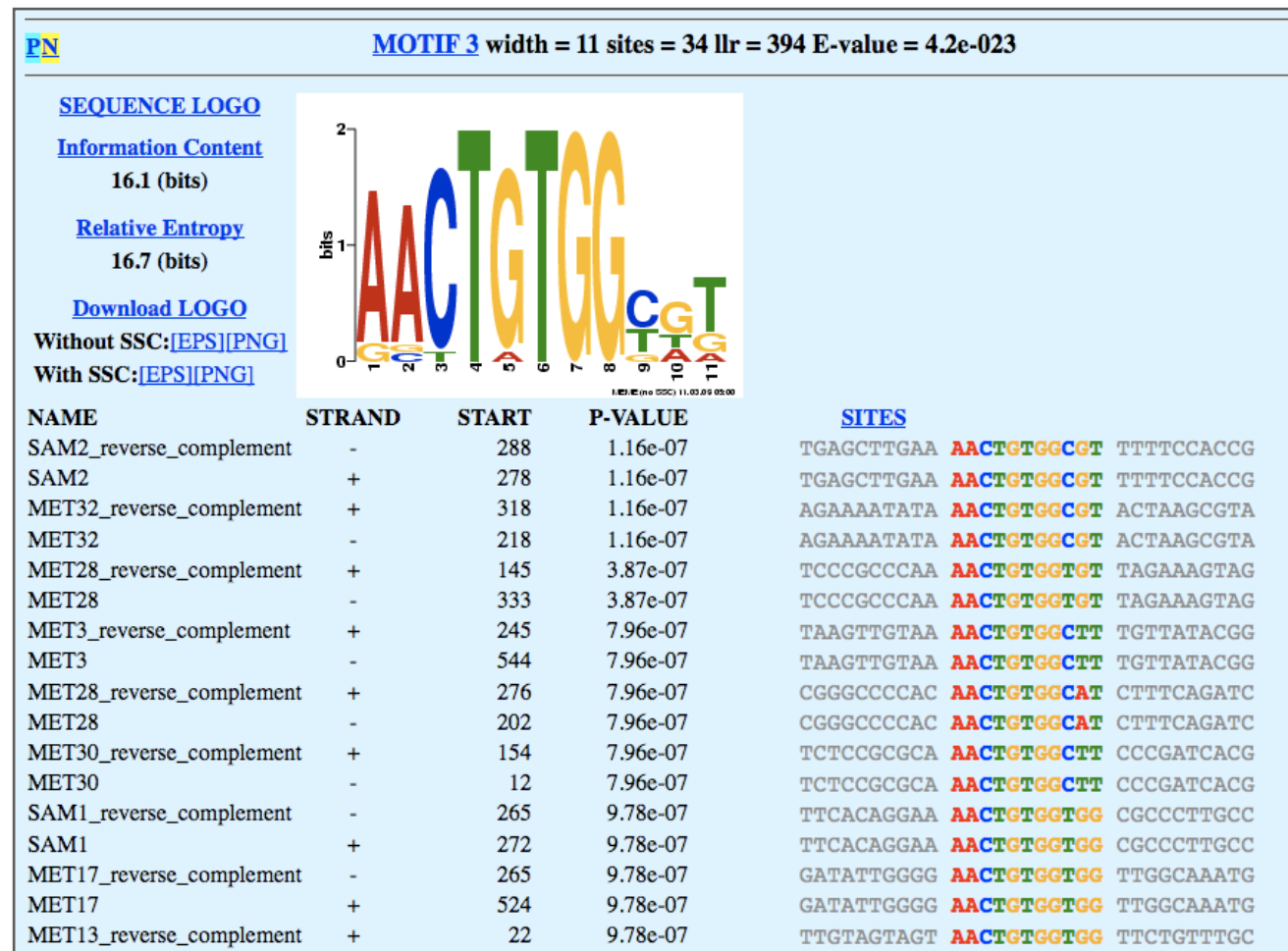


MEME - Multiple EM for Motif Elicitation

- Web interface + downloadable program
 - <http://meme.nbcr.net/>
- Reference
 - Timothy L. Bailey and Charles Elkan (1994). Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, pp. 28-36, AAAI Press, Menlo Park, California.
- Strengths
 - Flexible options
 - Matrices are scored with the E-value (expected number of false positives). Very low E-values are generally indicative of good results.
 - Supports multiple-widths (test various matrix widths and returns the most informative).
 - Supports higher-order background models (Markov chains)
 - This parameter strongly affects the result.
 - In my hands, higher order background models seem to give better results at least with yeast data sets.
- Weakness
 - Computing time increases quadratically with the size of the sequence set.

Example of MEME result

- We ran MEME with 30 yeast genes involved in methionine metabolism and sulfur assimilation
 - (We actually collected all genes having MEY\d+ or SAMd\+ in their names)
- MEME returned 3 motifs
 - The first ones are uninformative poly-A and polyT motifs.
 - The third one is the motif bound by Met31p or Met32p.

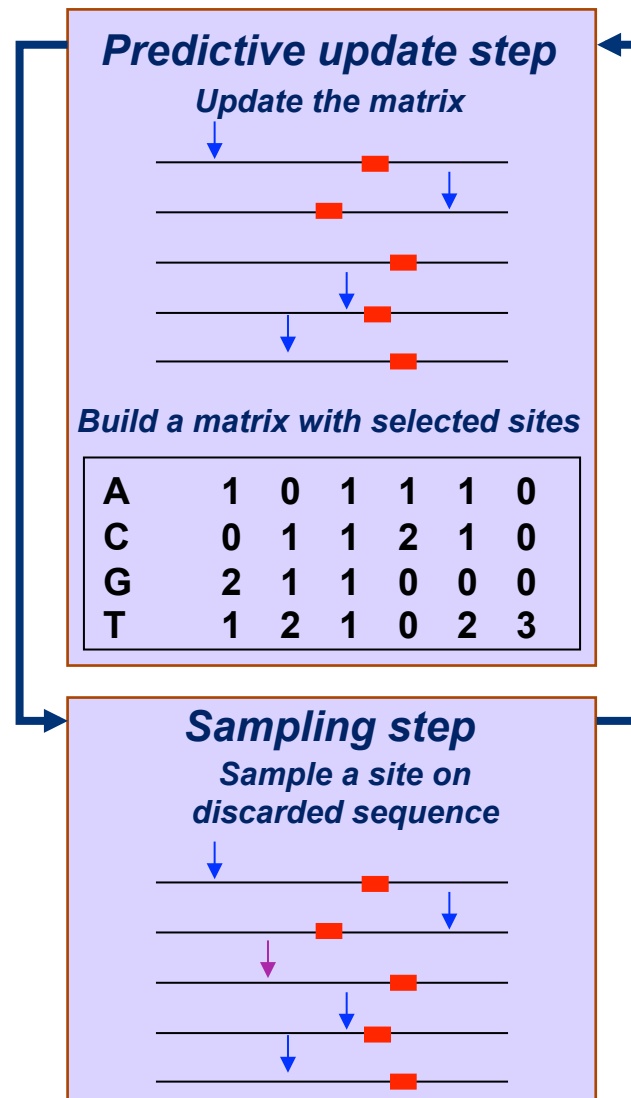


***Gibbs sampling
(stochastic Expectation - Maximization)***

Pattern discovery: The Gibbs sampler (gibbs motif sampler, by Andrew Neuwald)

Pretend you know the motif, this might become true

- Initialization
 - select a random set of sites in the sequence set
 - Create a matrix with these sites
- Sampling
(Stochastic Expectation)
 - Isolate one sequence from the set, and score each position (site) of the sequence.
 - Select one “random” site, with a probability proportional to the score (Ax, see next slide).
- Predictive update
(Maximization)
 - Replace the old site with a new site, and update the matrix
- Iterate steps 2 and 3 for a fixed number of cycles



After N iterations

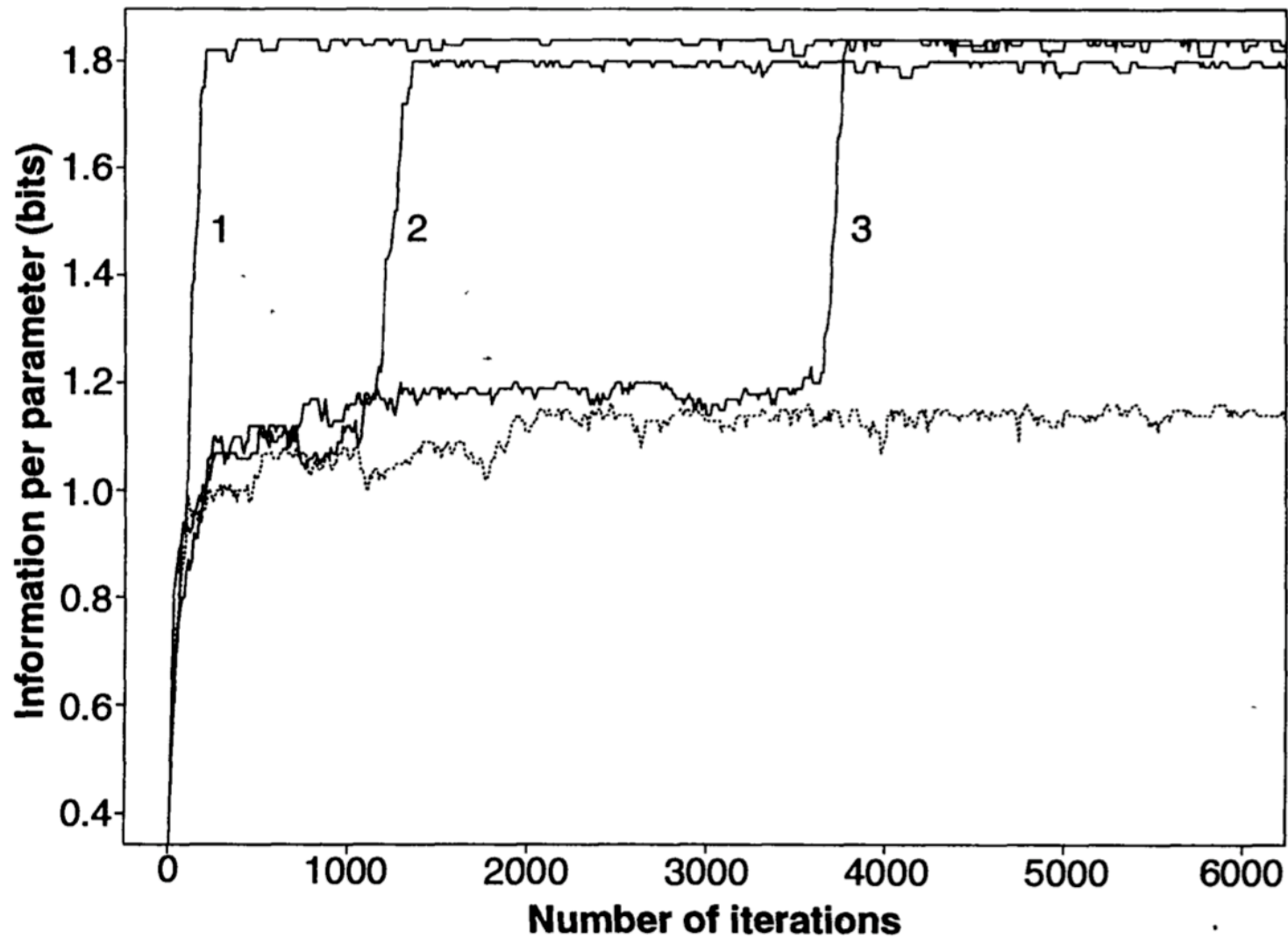
Found

Not found

Stochastic vs deterministic behaviour

- Why to select a random site ?
 - A deterministic behaviour would consist in selecting, at each iteration, the highest scoring site (the one which matches best the matrix)
 - This would give poor results because the program is attracted too fast towards local optima.
- Stochastic behaviour
 - At each iteration, the next site is selected in a stochastic rather than deterministic way: the probability of each site to be selected is proportional to its scoring with the matrix
 - This allows to avoid weak local optima, and converge towards better solutions.

Gibbs sampling: optimization of information content



source: Lawrence et al.(1993). Science 262(5131), 208-14.

Gibbs sampling - scoring scheme

$$A_x = Q_x / P_x$$

A_x weight of segment x
(used for random selection)
 Q_x probability to generate segment x
according to pattern probabilities q_{ij}
 P_x probability to generate segment x
according to the background
probabilities p_i

$$q_{i,j} = \frac{c_{i,j} + b_j}{N - 1 + B}$$

i index for the site
 j index for the residue
 $c_{i,j}$ counts for residue j at site i
 N number of sequences
 b_j pseudo-count for residue j
 B sum of pseudo-counts

$$F = \sum_{i=1}^W \sum_{j=1}^R c_{i,j} \ln \left(\frac{q_{i,j}}{p_j} \right)$$

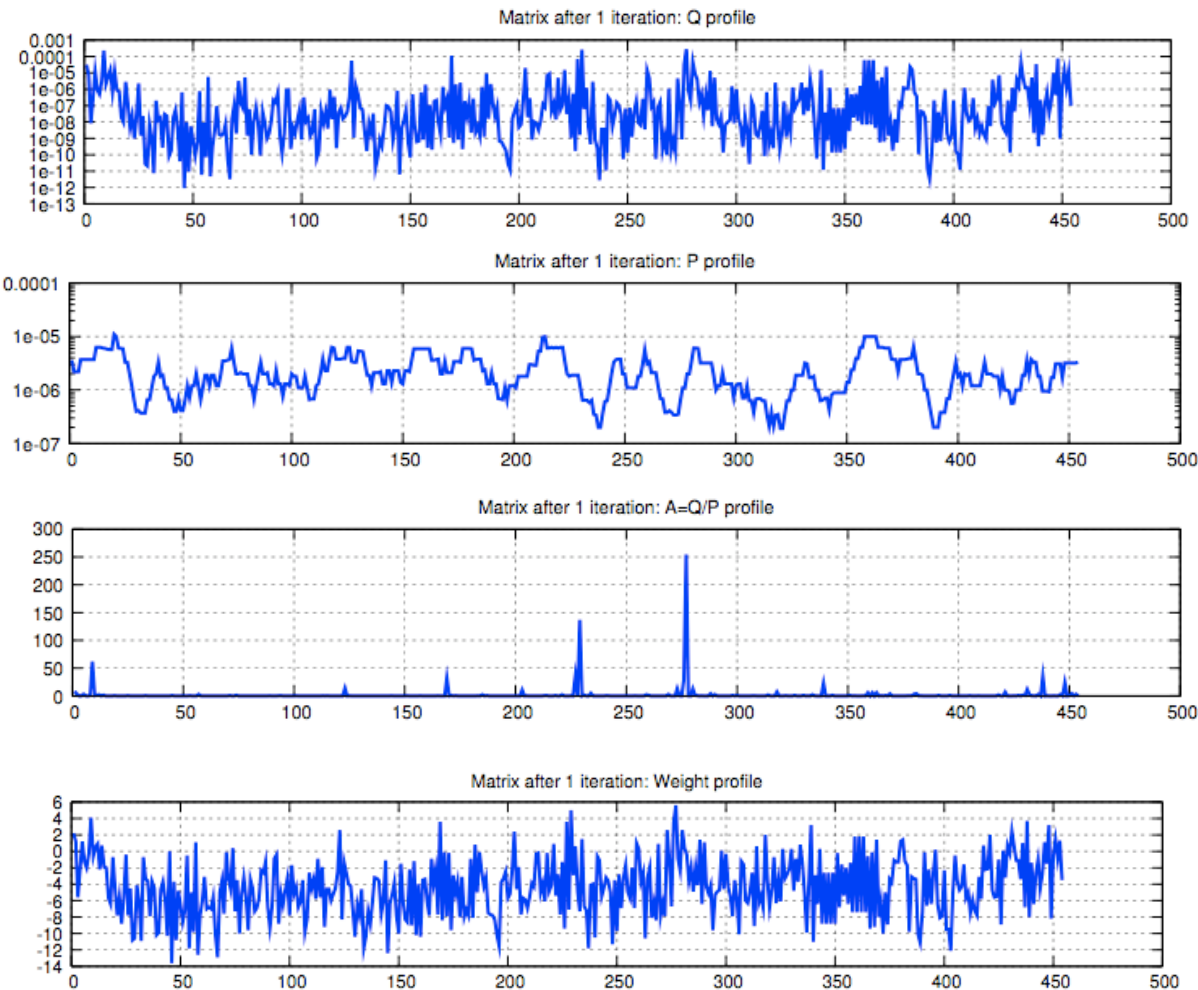
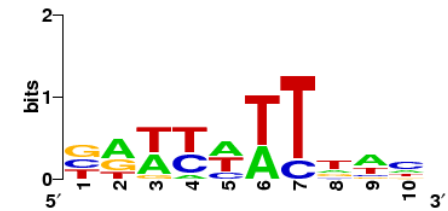
W width of the matrix
 R number of distinct residues
 p_j prior probability for residue j

A = Q/P profiles after the first iteration (random seed)

$$A_x = Q_x / P_x$$

- A_x weight of segment x
 (used for random selection)
 Q_x probability to generate segment x according to pattern
 probabilities q_{ij}
 P_x probability to generate segment x according to the
 background probabilities p_i

1 iteration, IC per col=0.30

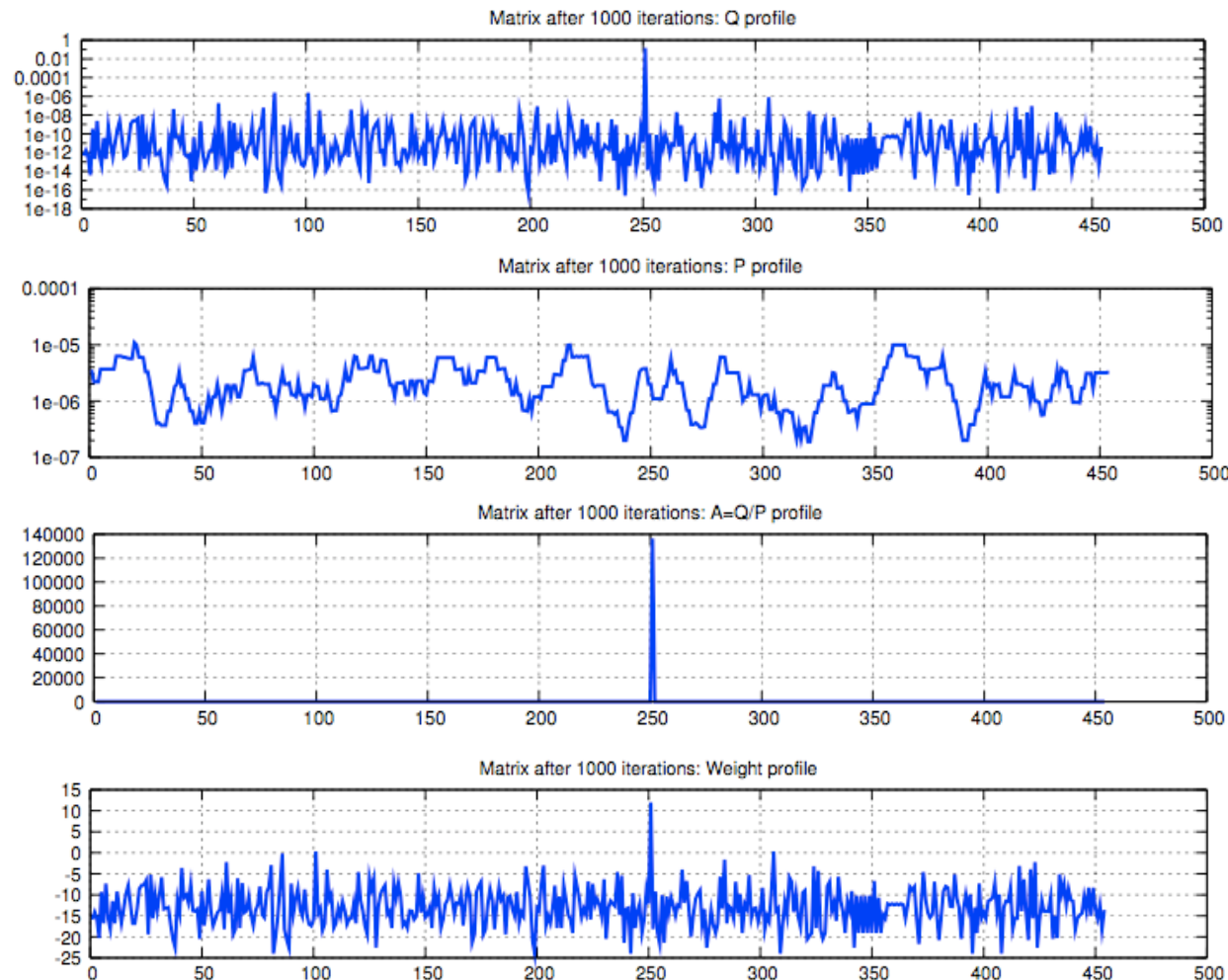
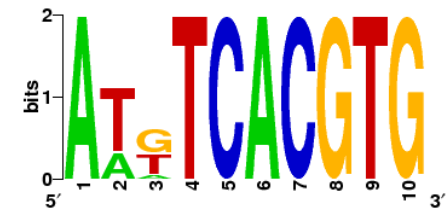


$A = Q/P$ profiles after 1000 iterations (Met4p motif found)

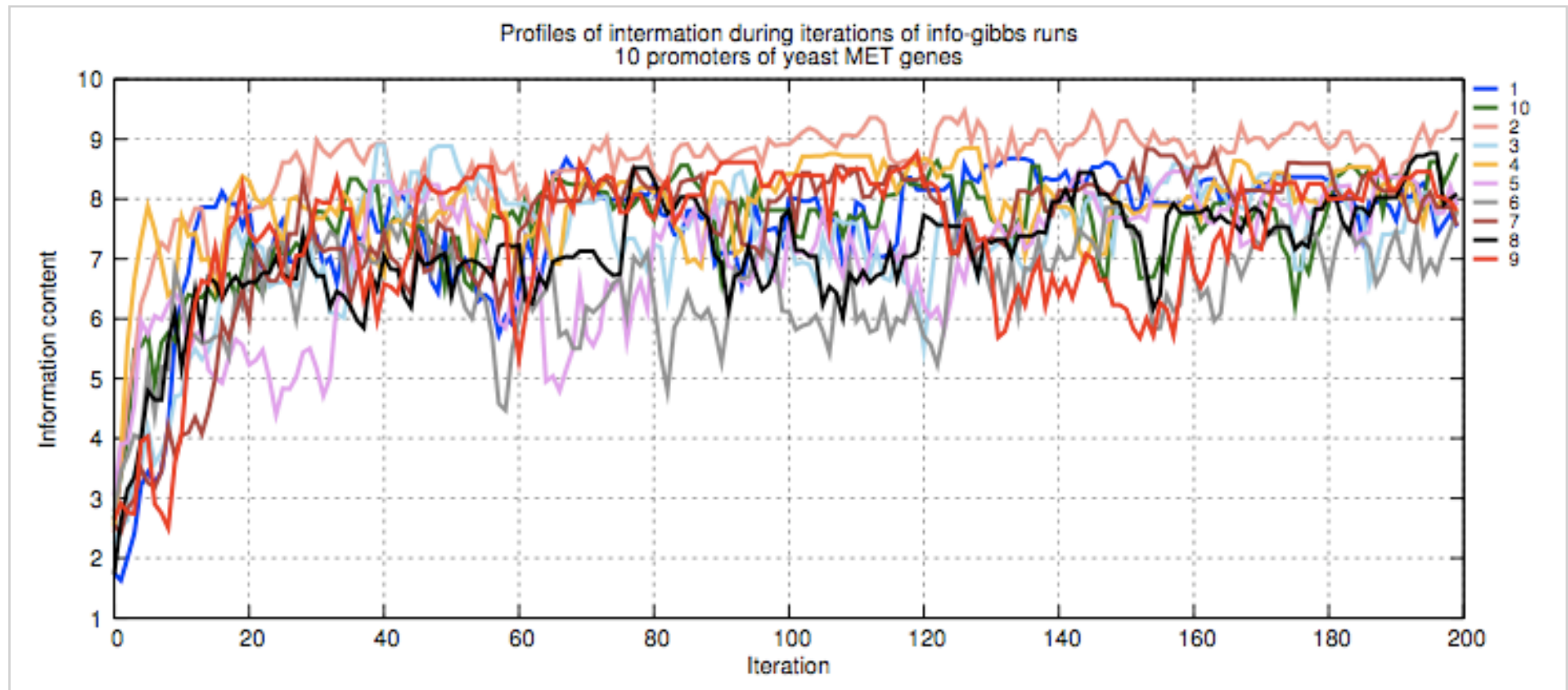
$$A_x = Q_x / P_x$$

- A_x weight of segment x
 (used for random selection)
 Q_x probability to generate segment x according to pattern
 probabilities q_{ij}
 P_x probability to generate segment x according to the
 background probabilities p_i

1000 iterations, IC per col=0.95

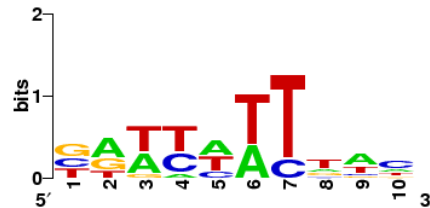


Profiles of information content during iterations of info-gibbs runs



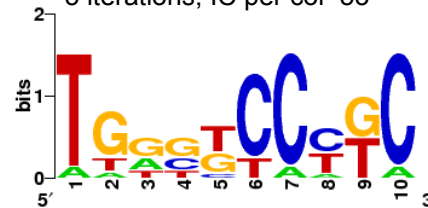
Examples of results

1 iteration, IC per col=0.303332



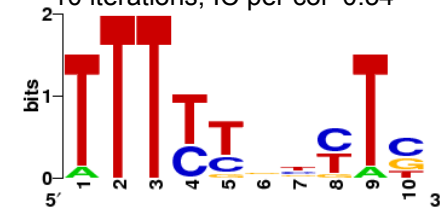
(another run)

5 iterations, IC per col=53



(another run)

10 iterations, IC per col=0.54



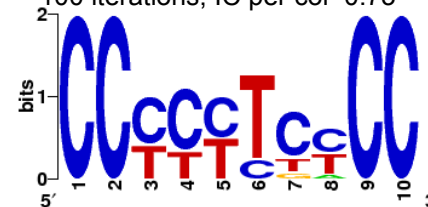
(another run)

100 iterations, IC per col=0.87



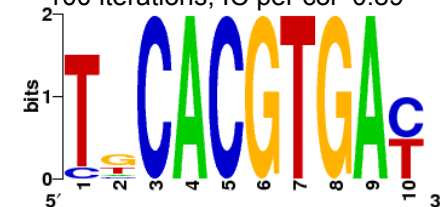
(another run)

100 iterations, IC per col=0.73



(another run)

100 iterations, IC per col=0.89



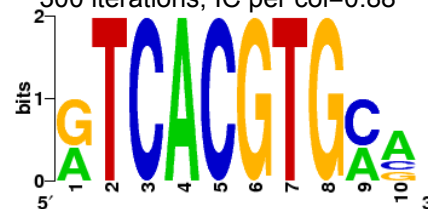
(another run)

200 iterations, IC per col=0.94



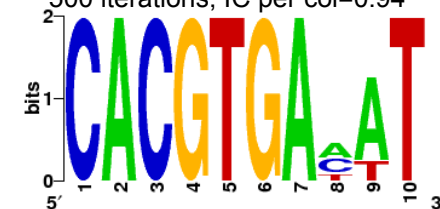
(another run)

300 iterations, IC per col=0.88



(another run)

500 iterations, IC per col=0.94



(another run)

1000 iterations, IC per col=0.89



(another run)

1000 iterations, IC per col=0.90



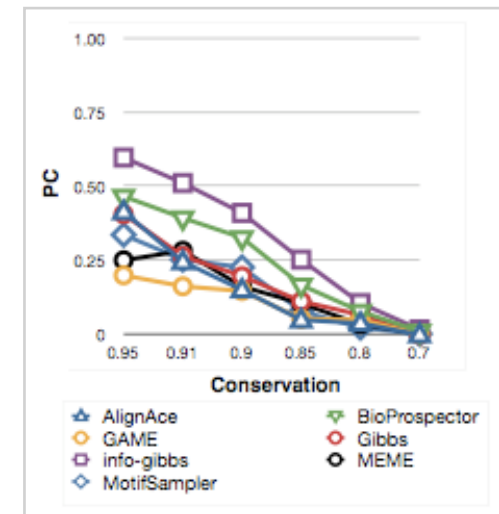
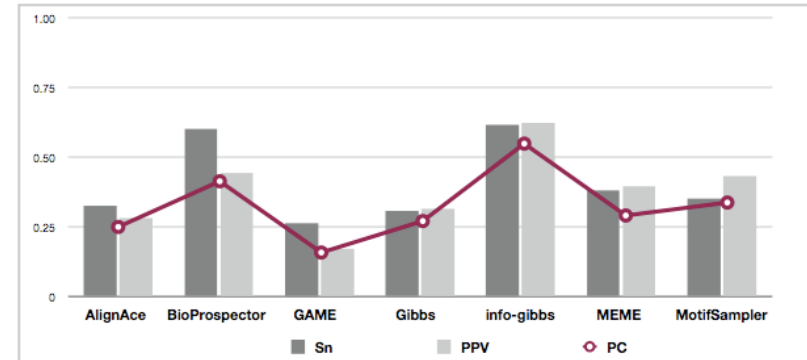
(another run)

1000 iterations, IC per col=0.90

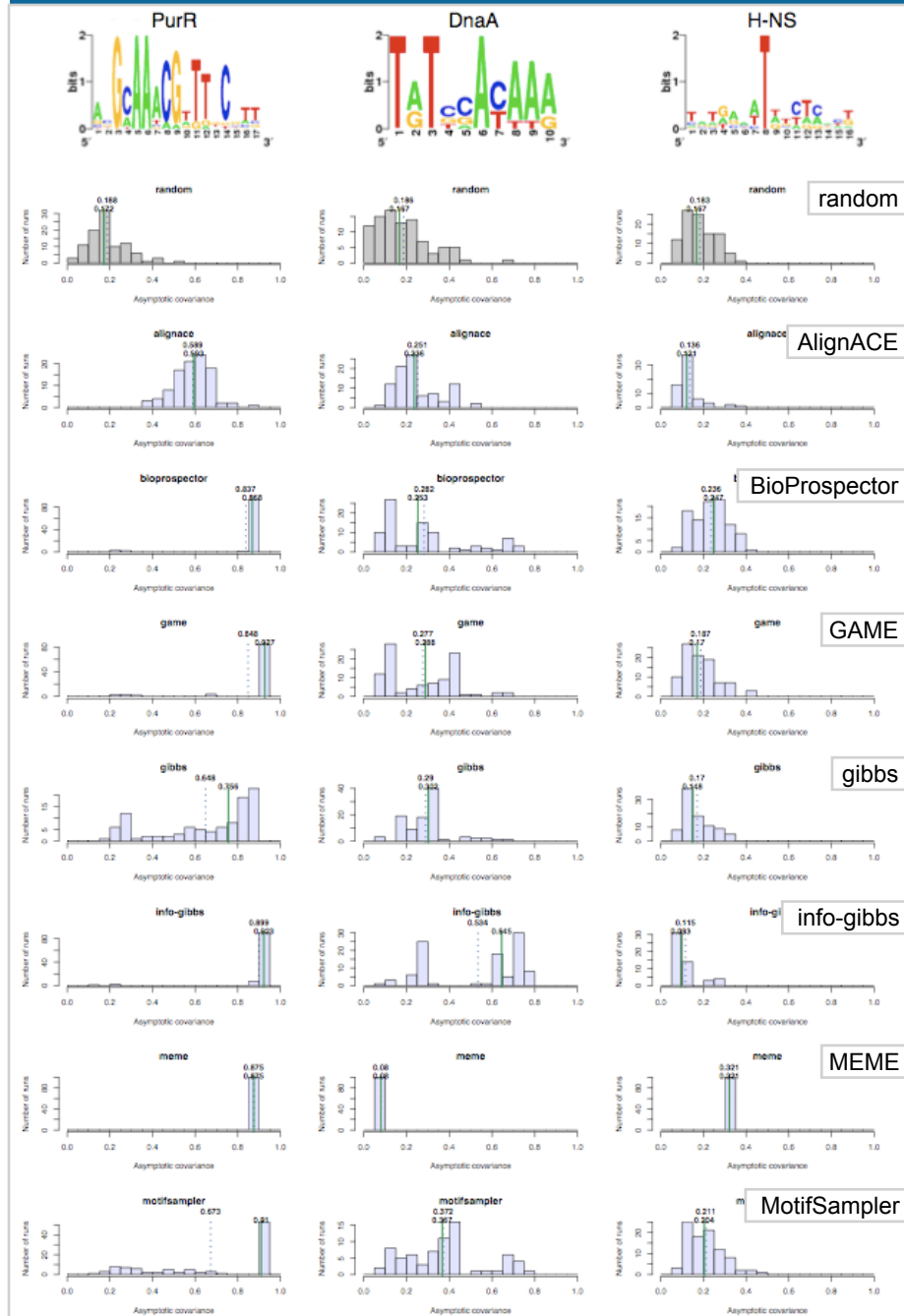


Evaluation on synthetic data

- Synthetic data
 - Generate random sequences (*random-seq*)
 - Generate random PSSMs (*random-motif*)
 - Generate random sites from those random motifs (*random-sites*)
 - Implant random sites at random positions of random sequences (*implant-sites*)
- Motif discovery
 - Those sequences are submitted to various algorithms.
 - The sites used to build the discovered motifs are compared to the implanted sites.
 - The process is run on 100 different artificial data sets.
- Statistics
 - Sensitivity: (correct sites) / (implanted sites)
 - $Sn = TP / (TP + FN)$
 - Positive Predictive Value: (correct sites) / (sites used to build motifs)
 - $PPV = TP / (TP + FP)$
 - Performance Coefficient : (correct sites) / (union of discovered and implanted sites)
 - $PC = TP / (TP + FP + FN)$
- Advantages of this evaluation protocol
 - The evaluation is accurate, since we control the implanted sites.
 - We can test the impact of various parameters (sequence length, number of sites, degree of conservation of the motifs, ...)
- Weaknesses of the evaluation
 - Performances on synthetic data may differ from performances on real biological sequences.
 - This evaluation is biased since we developed one of the algorithms. Even if we attempt to be as fair as we can, we know better how to handle our algorithm than those developed by other people.



Evaluation with known regulons



- We took all the regulons annotated in RegulonDB <http://regulondb.ccg.unam.mx/>
- For each factor, we collect the promoters of all target genes.
- We run each program 100 times, in order to evaluate the intrinsic variability of the results due to the stochasticity of the gibbs sampling.
- We compare discovered and annotated motifs by computing the asymptotic covariance according to Pape et al. (Bioinformatics 2008, 24:350-7).
- We estimate the random expectation by analyzing the distribution of asymptotic covariance for random motifs built by picking up random positions in the input sequences.

Fig. 1. Software performances on the RegulonDB data sets. The asymptotic covariance between annotated and predicted regulatory motifs for 32 *Escherichia coli* K12 regulons was computed over 100 trials. The mean is reported for a random selection and for each motif discovery software.

Factor	Genes	Sites	IC	random	AlignACE	BioProspector	GAME	Gibbs	info-gibbs	MEME	MotifSampler
AgaR	3	11	9.685	0.270	0.624	0.654	0.462	0.709	0.632	0.535	0.648
AraC	5	13	7.239	0.195	0.295	0.405	0.391	0.371	0.199	0.619	0.372
ArgR	16	18	9.805	0.247	0.824	0.836	0.893	0.796	0.902	0.832	0.872
CpxR	33	42	5.835	0.205	0.170	0.188	0.235	0.235	0.104	0.461	0.197
DeoR	3	7	7.776	0.200	0.278	0.209	0.328	0.243	0.222	0.651	0.423
DgsA	5	7	13.522	0.220	0.815	0.508	0.640	0.799	0.713	0.749	0.622
DnaA	6	8	6.690	0.186	0.251	0.282	0.277	0.290	0.534	0.080	0.372
FadR	9	12	8.609	0.216	0.732	0.807	0.751	0.563	0.825	0.821	0.515
Fis	70	133	5.292	0.176	0.201	0.206	0.321	0.372	0.204	0.312	0.294
FlhDC	20	18	6.816	0.207	0.411	0.580	0.546	0.546	0.387	0.582	0.516
FruR	22	12	10.768	0.167	0.889	0.862	0.830	0.819	0.909	0.878	0.867
GlpR	4	17	7.848	0.213	0.577	0.740	0.659	0.692	0.687	0.394	0.637
GntR	6	10	10.289	0.196	0.575	0.781	0.769	0.576	0.829	0.791	0.766
H-NS	52	35	4.936	0.183	0.136	0.236	0.187	0.170	0.115	0.321	0.211
IHF	77	84	5.267	0.297	0.000	0.359	0.351	0.253	0.259	0.402	0.335
IclR	2	10	4.782	0.227	0.132	0.417	0.081	0.104	0.236	0.427	0.232
LexA	25	23	10.995	0.216	0.659	0.846	0.891	0.855	0.901	0.855	0.892
MalT	5	14	6.048	0.321	0.456	0.638	0.642	0.423	0.710	0.690	0.472
MetJ	9	23	7.099	0.180	0.472	0.705	0.727	0.659	0.664	0.687	0.686
Nac	10	12	6.762	0.224	0.244	0.248	0.238	0.301	0.268	0.365	0.275
NagC	7	10	10.832	0.263	0.323	0.234	0.278	0.332	0.412	0.306	0.302
NanR	3	6	6.497	0.252	0.520	0.569	0.541	0.560	0.666	0.534	0.460
NtrC	15	17	8.886	0.215	0.827	0.839	0.822	0.844	0.895	0.830	0.748
OmpR	12	19	8.103	0.283	0.297	0.149	0.135	0.283	0.118	0.253	0.271
PhoB	14	120	11.411	0.205	0.124	0.699	0.639	0.429	0.115	0.751	0.567
PhoP	20	19	8.162	0.240	0.195	0.275	0.334	0.496	0.611	0.186	0.350
PurR	19	18	10.496	0.188	0.589	0.837	0.848	0.648	0.899	0.875	0.673
RcsAB	10	9	7.870	0.231	0.315	0.293	0.266	0.242	0.156	0.617	0.393
SoxS	18	18	6.753	0.201	0.271	0.288	0.304	0.307	0.344	0.400	0.273
TorR	5	6	6.546	0.170	0.482	0.654	0.357	0.654	0.674	0.188	0.443
TrpR	5	10	14.537	0.213	0.758	0.613	0.856	0.681	0.858	0.878	0.840
TyrR	9	17	9.085	0.229	0.854	0.779	0.282	0.802	0.790	0.746	0.704

Gibbs sampling: summary

- Strengths
 - Fast
 - Probabilistic description of the patterns
 - Can run with proteins or DNA
- Weaknesses
 - Stochasticity: returns a different result at each run
 - Can be attracted by local maxima
 - solution: run repeatedly and check which motifs come often
 - No threshold on pattern significance
 - → frequent false positive
- Note: the original Gibbs sampler was based on Bernoulli background models
 - → in yeast, often returns A/T-rich regions
 - This is however improved in some versions of the Gibbs samplers which use Markov chains for estimating the background probabilities (eg the MotifSampler developed by Gert Thijs)

AlignACE, ScanACE and CompACE *gibbs sampler tools for regulatory sequence analysis*

- Single/both strands
- Return multiple matrices, with iterative masking preventing slight variants of the same pattern
- Matrix clustering
- A posteriori evaluation of pattern significance, by analysing the whole-genome frequency of the discovered matrix.
- References
 - Roth et al. (1998). Nat Biotechnol 16(10), 939-45.
 - Tavazoie et al. (1999). Nat Genet 22(3), 281-5.
 - Hughes et al. (2000). J Mol Biol 296(5), 1205-14.
 - McGuire et al. (2000). Genome Res 10(6), 744-57.

Some gibbs sampling implementations

- Gibbs 1993
 - The first implementation of the gibbs sampler for finding motifs in biological sequences
 - *Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F. and Wootton, J. C. (1993). Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. Science 262, 208-14.*
- Gibbs 1995
 - 0 or several matches per sequence
 - column sampling (spacings can be admitted between columns of the matrix)
 - *Neuwald, A. F., Liu, J. S. and Lawrence, C. E. (1995). Gibbs motif sampling: detection of bacterial outer membrane protein repeats. Protein Sci 4, 1618-32.*
- AlignACE
 - Specific implementation for DNA (double strand is treated)
 - post-filtering of motifs according to number of matches in the genome, in order to discard frequent motifs
 - *Roth, F. P., Hughes, J. D., Estep, P. W. and Church, G. M. (1998). Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. Nat Biotechnol 16, 939-45.*
- BioProspector
 - Higher-order Markov-chains to estimate background probabilities.
 - *Liu, X., Brutlag, D. L. and Liu, J. S. (2001). BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. Pac Symp Biocomput, 127-38.*
- MotifSampler
 - Higher-order Markov-chains to estimate background probabilities.
 - *Thijs, G., Lescot, M., Marchal, K., Rombauts, S., De Moor, B., Rouze, P. and Moreau, Y. (2001). A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. Bioinformatics 17, 1113-22.*
- info-gibbs
 - Direct optimization of the information content rather than Qx/Px ratio.
 - Defrance & van Helden (2009). *Submitted manuscript.*

Summary: matrix-based pattern discovery

Matrix-based pattern discovery: summary

■ Strengths

- More specific description of degeneracy than with string-based approaches (frequency of each residue at each position).
- The resulting motif is more accurate than a string for pattern matching (more sensitive scoring scheme).

■ Weaknesses

- Intrinsic impossibility to explore all possible alignments.
- Results strongly depend on parameter setting. Two essential parameters have to be selected :
 - Matrix width
 - Expected number of sites
- The best parameter may change depending on the organism, sequence number, site density, etc.. Choosing the appropriate setting requires experience.