Inferring Models of cis-Regulatory Modules

BMI/CS 776

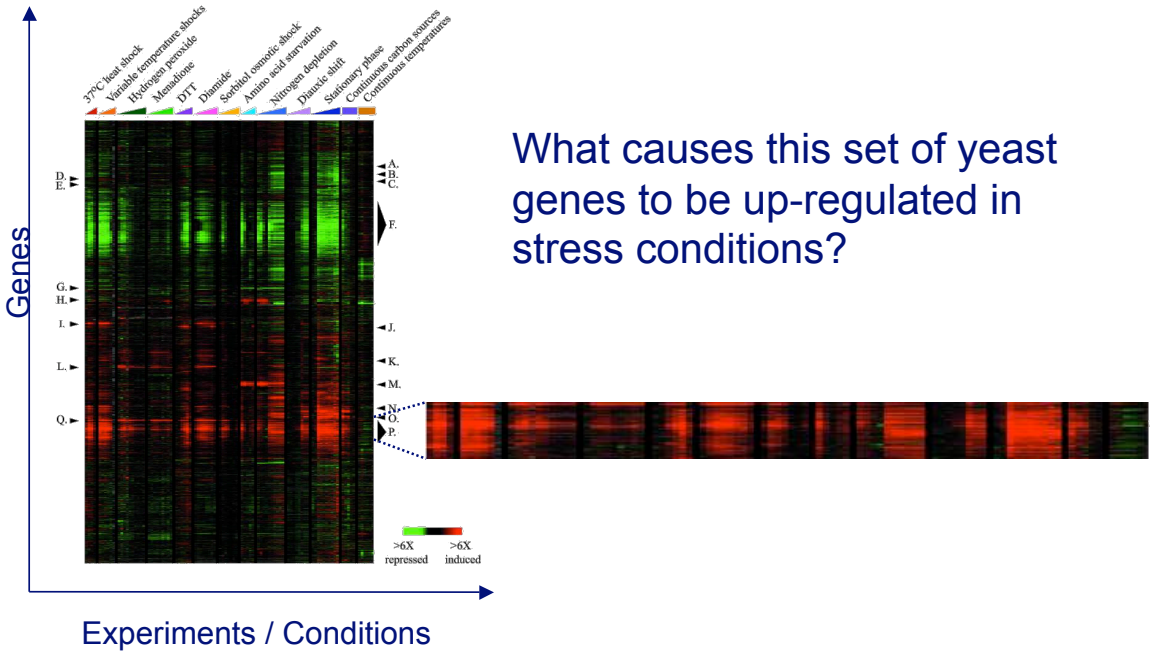
www.biostat.wisc.edu/bmi776/

Spring 2009

Mark Craven

craven@biostat.wisc.edu

A Common Type of Question

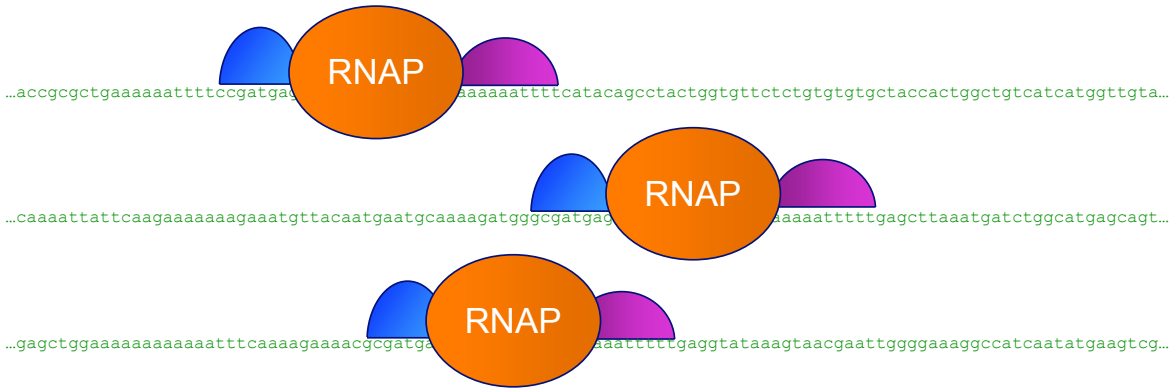


What causes this set of yeast genes to be up-regulated in stress conditions?

Figure from Gasch *et al.*, *Mol. Biol. Cell*, 2000

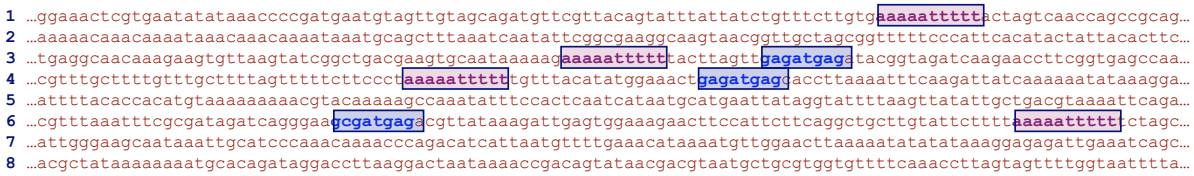
cis-Regulatory Modules (CRMs)

- co-expressed genes are often controlled by specific configurations of binding sites



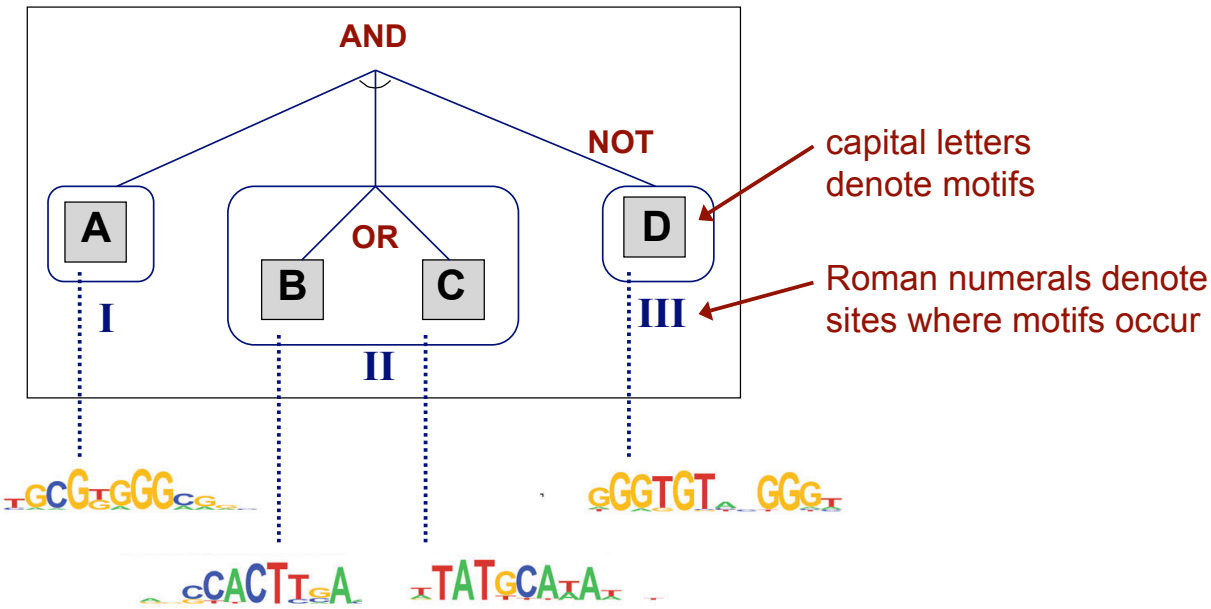
CRM Learning Task

- **Given:** sequences believed to share a CRM, and "negative" sequences believed not to contain the CRM
- **Learn:** models of the binding-site motifs and the spatial and logical relationships that characterize the CRM



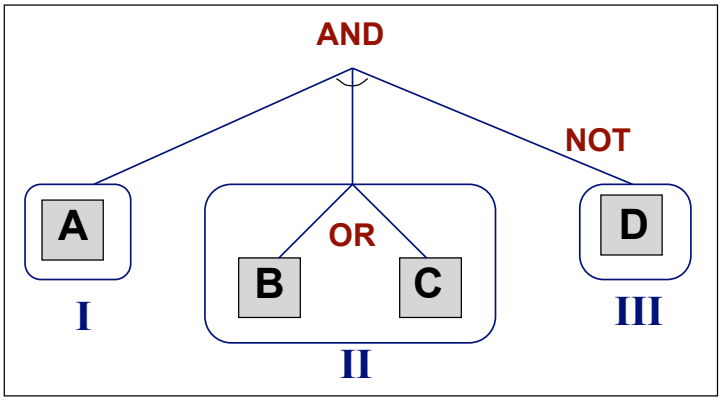
CRM Representation

- to characterize CRMS we want to represent logical relationships among motifs



CRM Representation

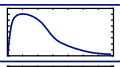
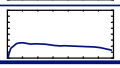
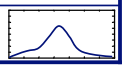
logical aspects: AND, OR, and NOT



- we also want to represent spatial relationships that characterize binding sites

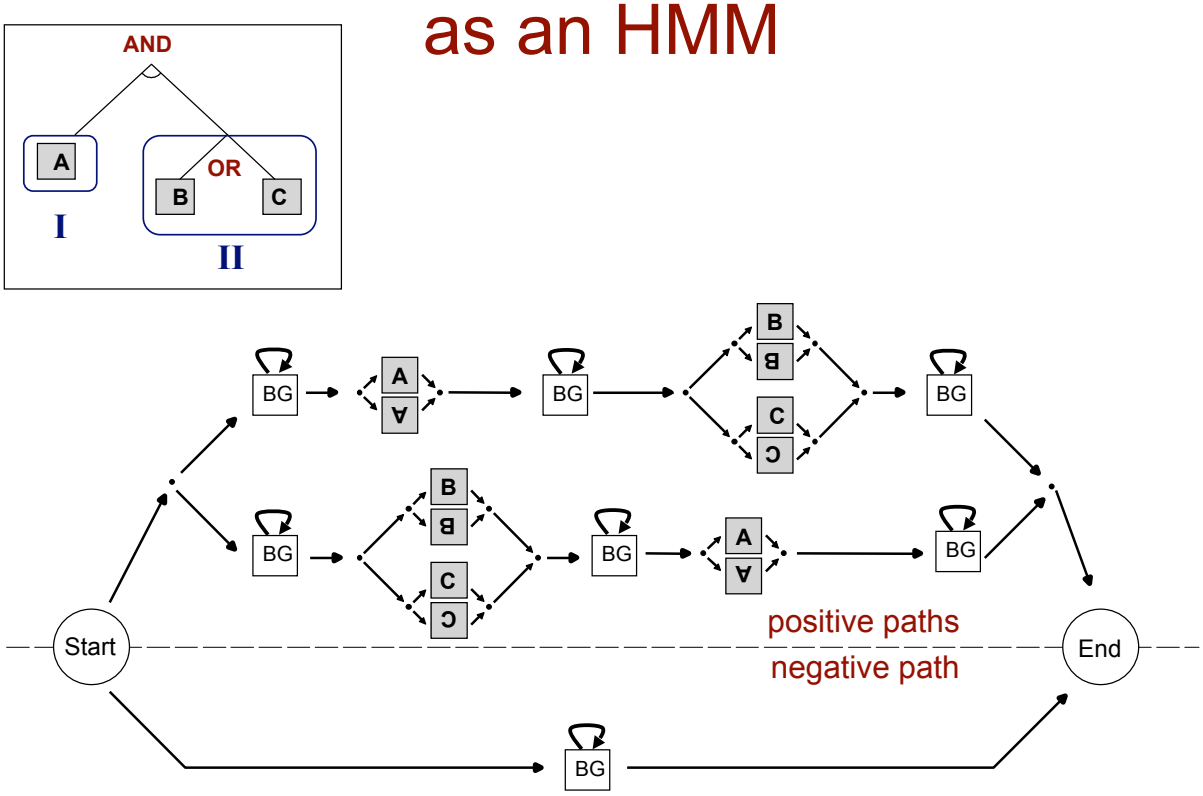
spatial aspects: order, distance, and strand

Order	I	II
II	0.5	
III	0.2	0.7

Distance	I	II
II		
III		

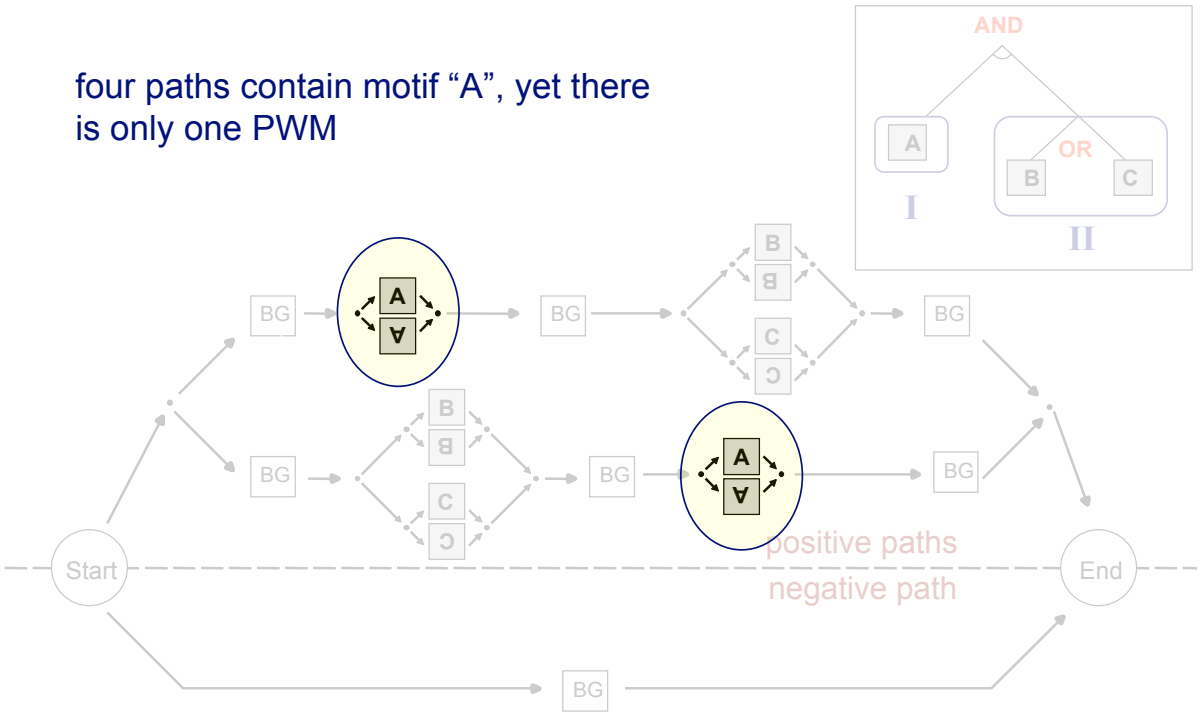
Strand	P(template)
I	0.8
II	0.5
III	0.3

The CRM representation viewed as an HMM



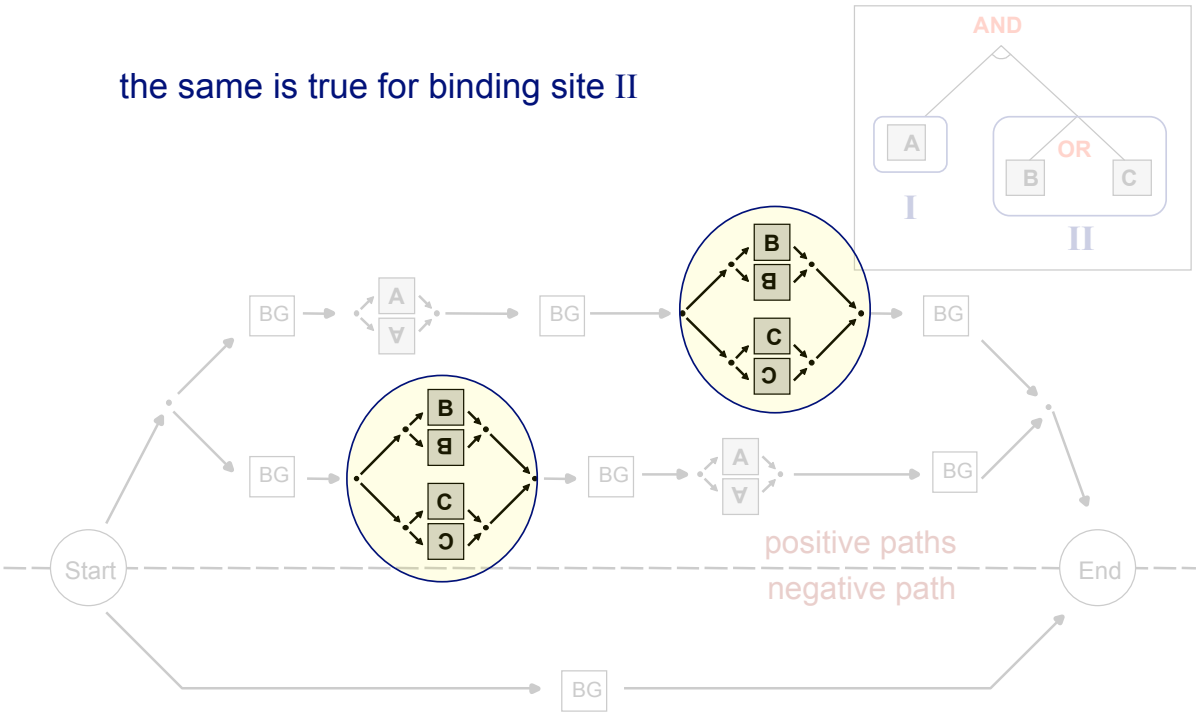
Parameter Tying

four paths contain motif “A”, yet there is only one PWM



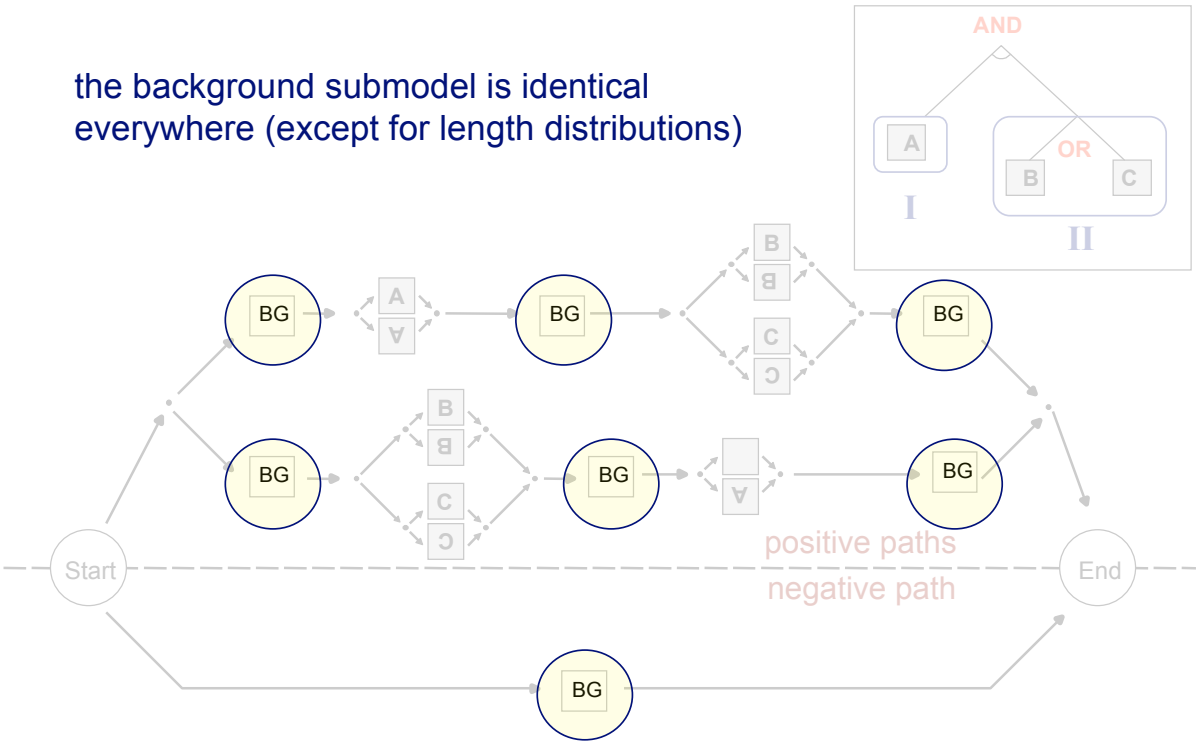
Parameter Tying

the same is true for binding site II

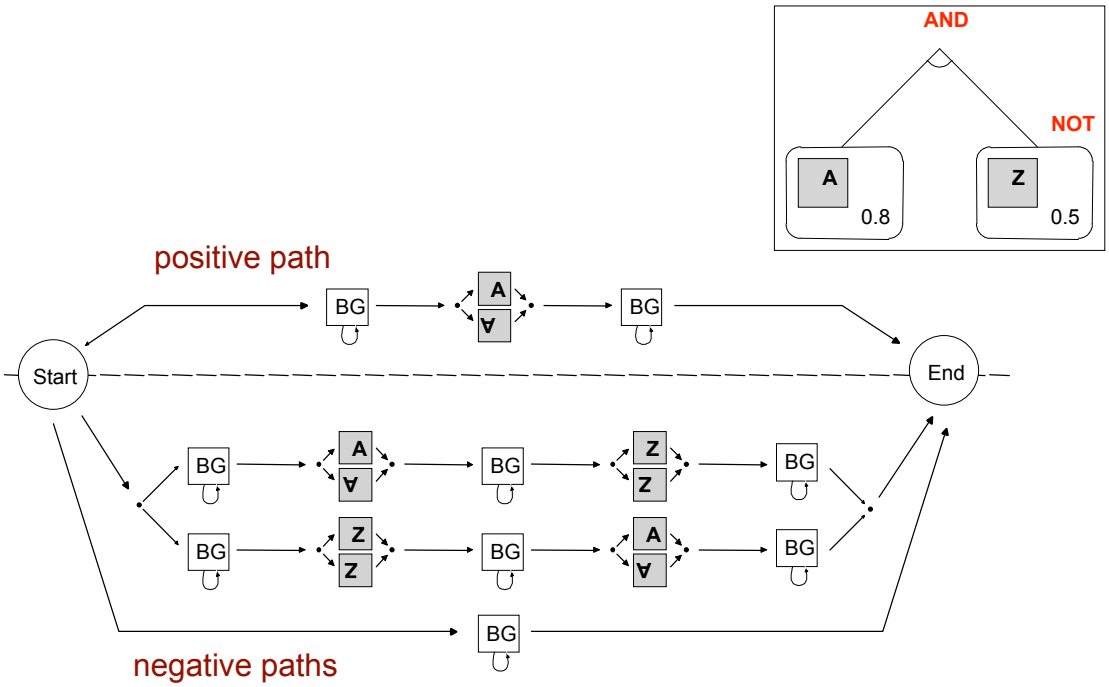


Parameter Tying

the background submodel is identical
everywhere (except for length distributions)

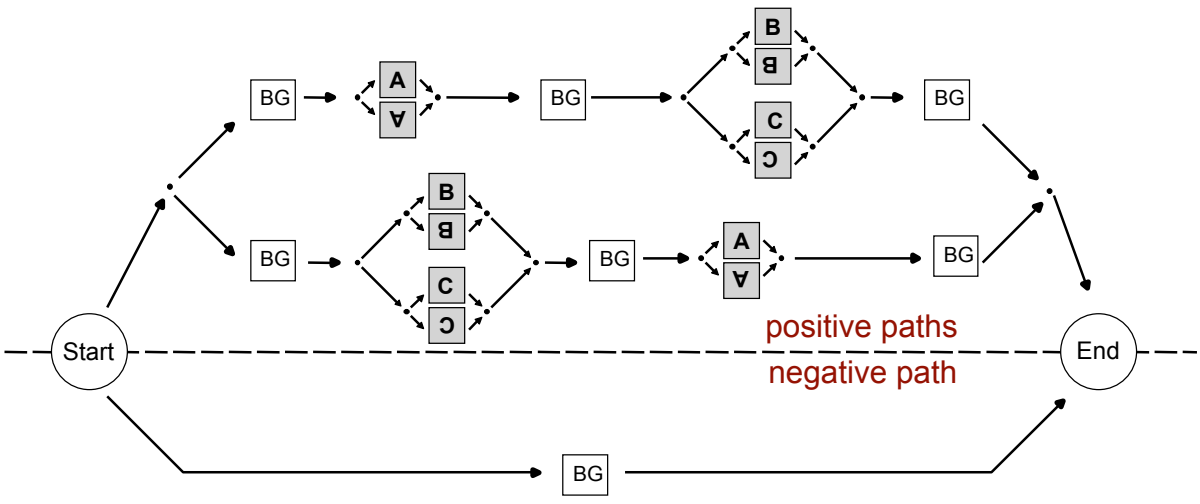


HMM with a *Negated* Binding Site



The Classification Task

- given a test sequence, we can calculate the most probable path (Viterbi algorithm) or the summed probability for all positive paths (Forward algorithm)



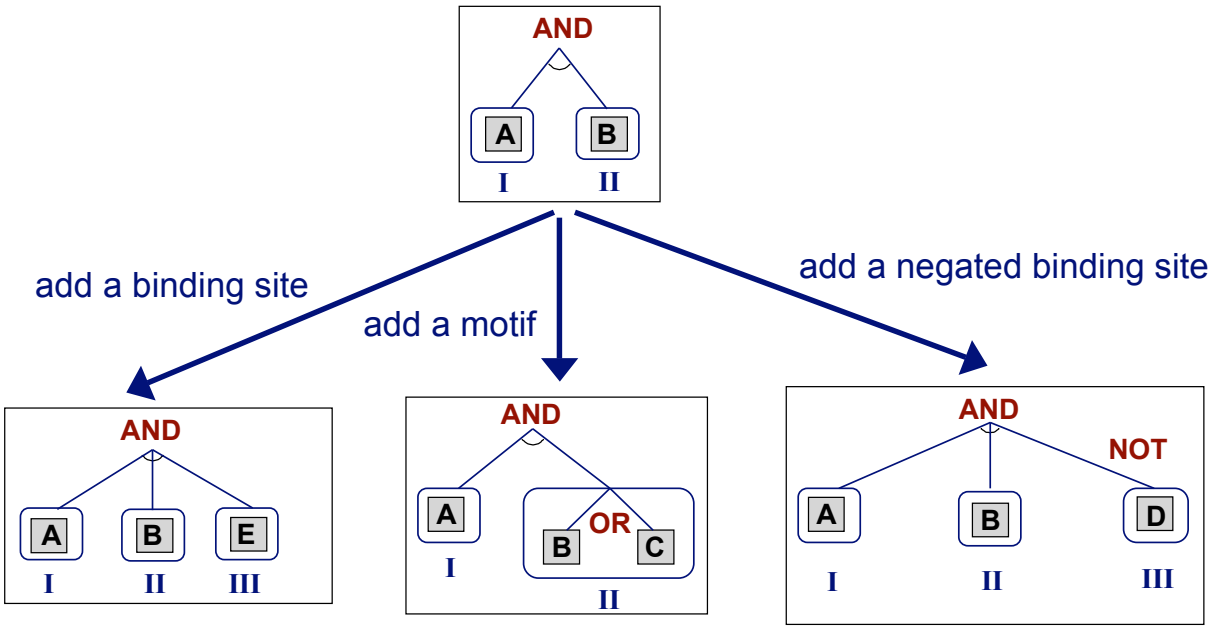
The Learning Tasks

1. *parameter learning* : estimating the probability parameters of the HMM
   - use Baum-Welch since we don't know the correct parse for training sequences

2. *structure learning*: determining the topology of the HMM

Learning The HMM Structure

- the structure search is carried out on the *compact* representation
- three operators



Beam Search for Structure Learning

given: initial structure I, search operators
$beam \leftarrow \{ I \}$
repeat
    $H \leftarrow$ highest scoring structure in beam
    foreach operator O
        $S \leftarrow O$ applied to H
        if $|beam| < k$
            add S to beam
        else if score(S) > score(L) for lowest scoring element L in beam
            remove L from beam
            add S to beam
until stopping criteria met
return: highest scoring structure in beam

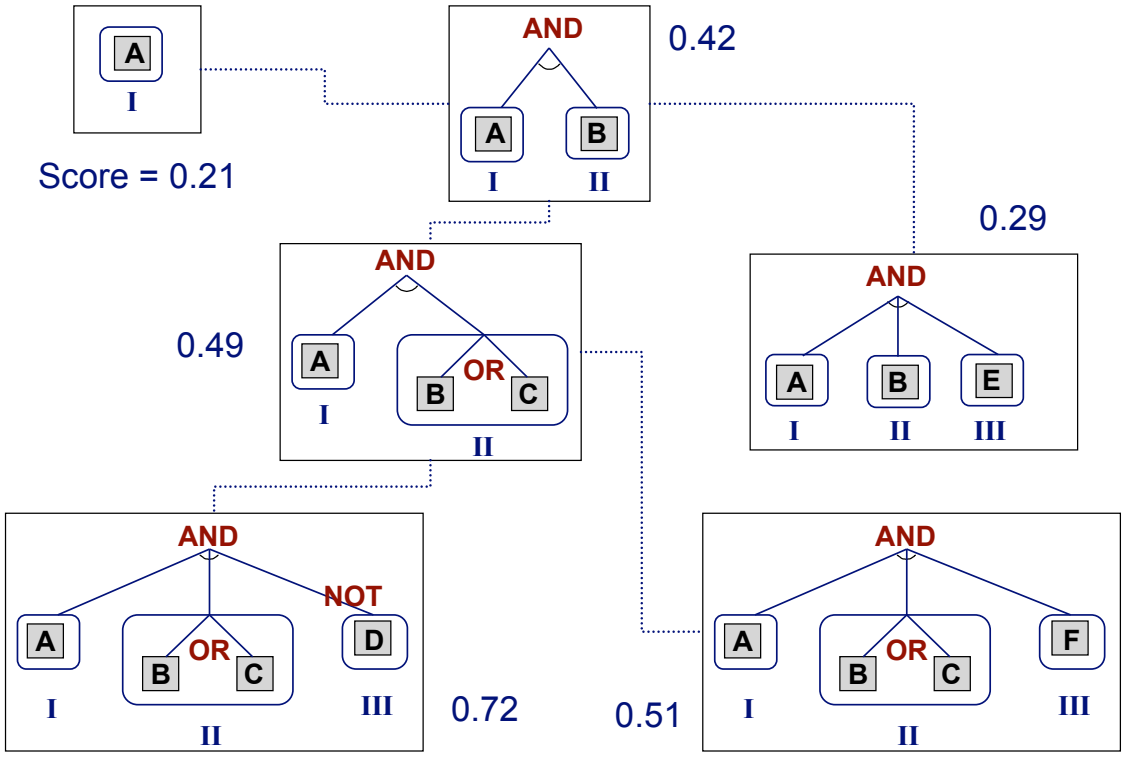
Beam Search in Structure Learning

"scoring" a structure entails

1. retraining parameters of HMM using something like Baum-Welch
   - thus learning a model for the new motif
   - for background states, only length parameters are retrained

2. estimating accuracy of model using a held-aside tuning set

Learning The HMM Structure

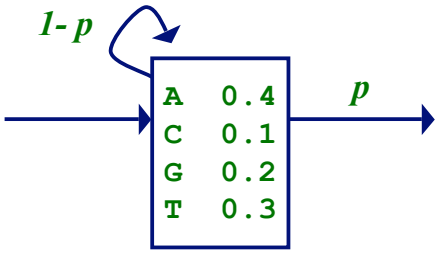


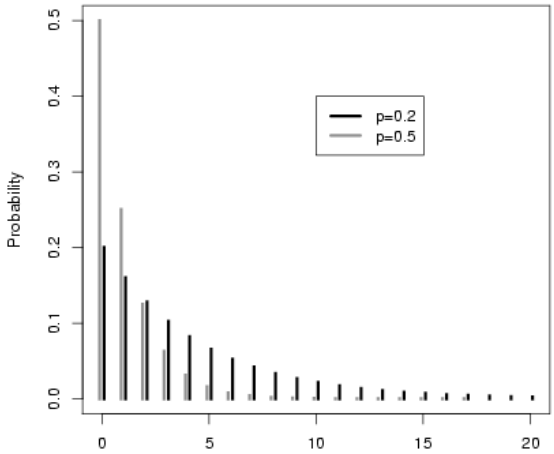
Semi-Markov HMMs (a.k.a. Generalized HMMs)

- to encode distance preferences, models use semi-Markov states for the background
- key idea: decouple length from composition

Duration Modeling in HMMs

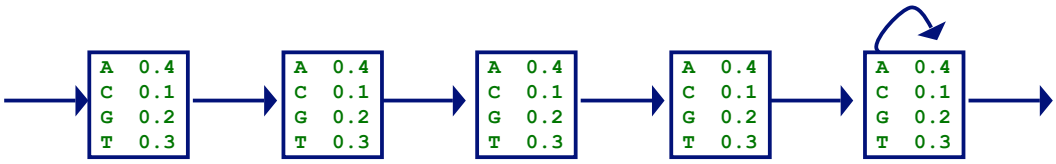
- suppose we have a type of sequence for which the base distribution is the same regardless of length
- the simplest way to model it:



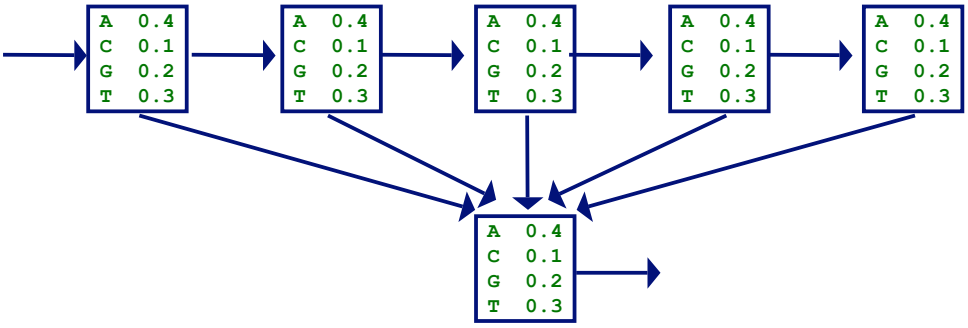


- this encodes a *geometric* distribution (shifted by 1) on the length of sequences

Duration Modeling in HMMs



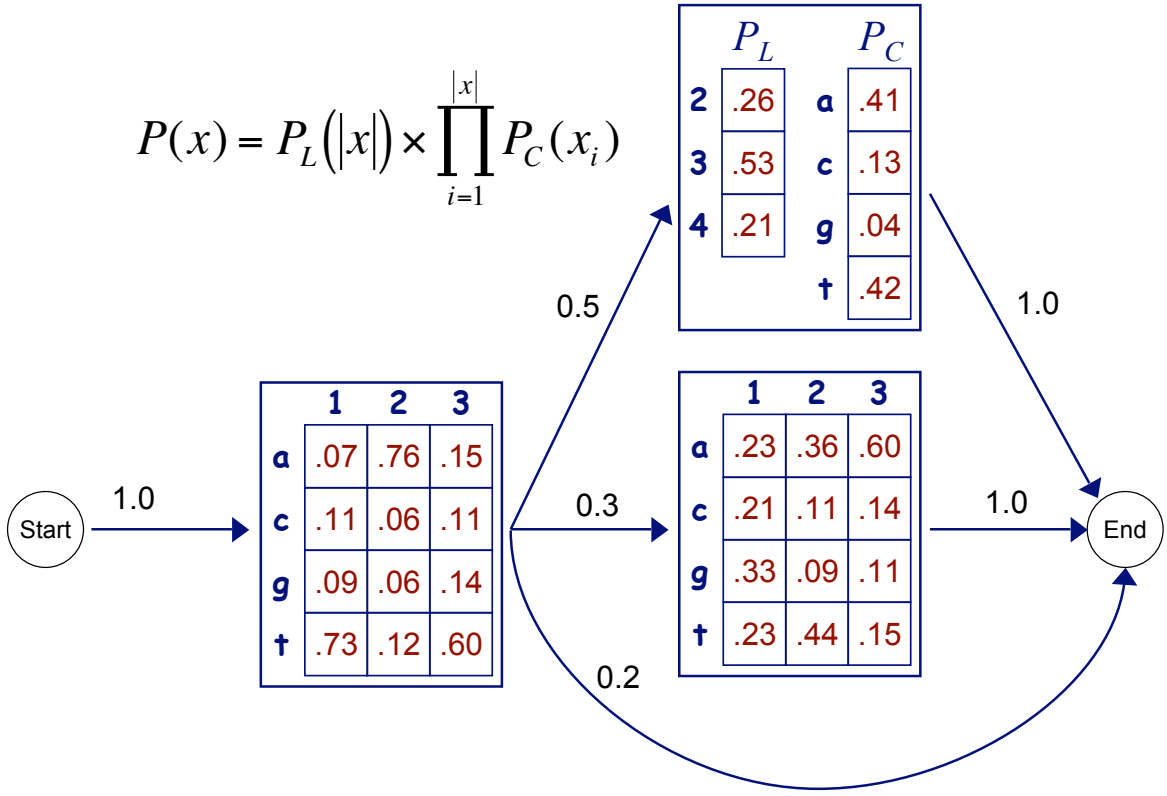
- min length = 5; geometric distribution over longer sequences



- any distribution over length 2 to 6

A Simple Semi-Markov Model

$$P(x) = P_L(|x|) \times \prod_{i=1}^{|x|} P_C(x_i)$$



DP with Semi-Markov Models

- review: Forward algorithm recurrence for HMMs

$$f_l(i) = \sum_k f_k(i-1)\ \underbrace{a_{kl}}_{\text{transition from } k \text{ to } l}\ \underbrace{P(x_i \mid q_l)}_{\text{prob. of emitting } x_i \text{ from } l}$$

- for semi-Markov models: each Forward value assumes we're ending a segment in the given state

$$f_l(i) = \sum_k \sum_{d=1}^{D} \left[f_k(i-d)\ a_{kl}\ \underbrace{P(d \mid q_l)}_{\text{prob. of length } d \text{ segment from } l}\ \underbrace{\prod_{j=i-d+1}^{i} P(x_j \mid q_l)}_{\text{prob. of emitting } x_{i-d+1} \ldots x_i \text{ from } l} \right]$$

Semi-Markov Models

- representing a parse π, as a sequence of states and associated lengths (durations)

$$\vec{q} = \{q_1, q_2, \ldots, q_n\} \qquad \vec{d} = \{d_1, d_2, \ldots, d_n\}$$

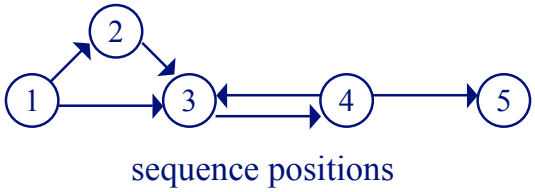
- the joint probability of generating parse π and sequence x

$$P(x,\pi) = a_{start,1} P(d_1 \mid q_1) P(x_1 \mid q_1, d_1) \times \prod_{k=2}^{n} a_{k-1,k} P(d_k \mid q_k) P(x_k \mid q_k, d_k)$$

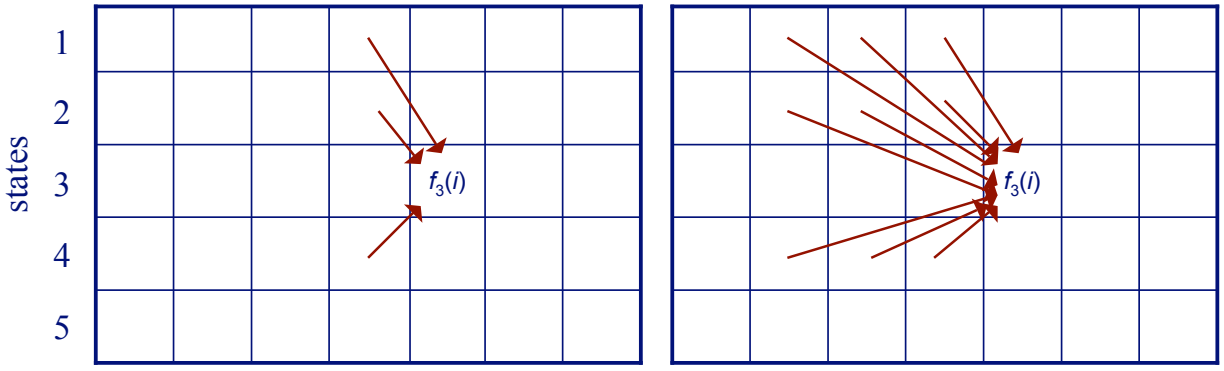
transition probabilities

the k^{th} segment of the sequence

DP with Semi-Markov Models



sequence positions



complexity of Viterbi/Forward/Backward in standard HMMs is $O(S^2L)$ where S = number of states, L = sequence length

complexity in semi-Markov HMMs is $O(S^2LD)$ where D = maximum length of a segment

Speeding up DP in Structure Search

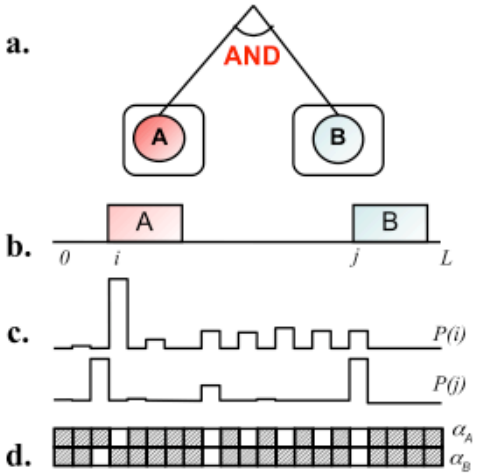
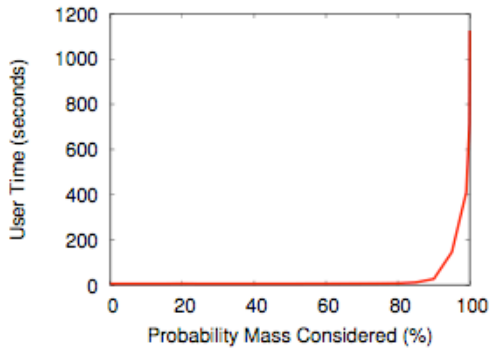


Figure 4.8 Illustration of efficient dynamic programming in SCRM2. **a.** A CRM logical structure. **b.** Possible binding site locations on a DNA sequence **x**. **c.** A probability distribution over the locations of binding sites A and B, respectively. These probabilities tend to be extreme (a motif is present at a location or it is not) and high probabilities are sparsely distributed. **d.** A forward dynamic programming matrix f, where $f_A(i)$ represents the likelihood of sequence **x** from location 1 to i when site A occurs at location i.

Speeding up DP in Structure Search



Probability Mass Considered (%)	User Time (min:sec)
100	18:45
99.99	15:22
99.9	11:52
99	6:50
95	2:26
90	0:27.2
75	0:7.19
50	0:5.78
10	0:6.00
1	0:5.79

can consider parses comprising 90% of probability mass in 2.4% of the time required to consider all parses

Figure 4.9 Time to train one two-binding site CRM model on 500 bp yeast sequences from [Lee *et al.*, 2002] as a function of the motif location probability mass examined during SCRM2's dynamic programming calculations.

Empirical Evaluation of CRM Learning Method

- 25 yeast data sets
  - each with 15 – 214 promoter sequences
  - 100 negative promoter examples in each set
- each has evidence of two binding proteins (Lee *et al.*, *Science* 2002)

- cross-validation methodology

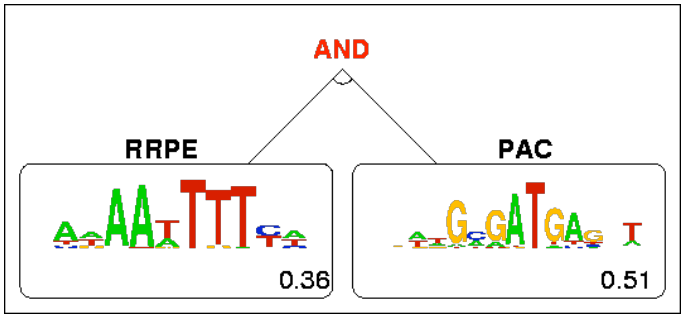
Empirical Evaluation of CRM Learning Method

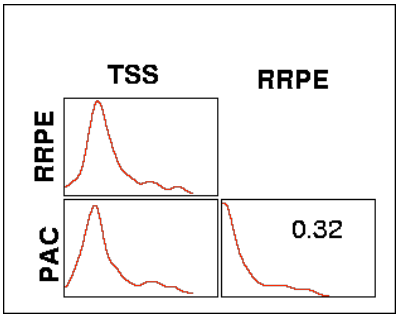
- assess models by calculating how likely they would have test-set accuracy as good by chance

- Segal and Sharan's method (*RECOMB* 2004) induced models with (statistically significant) accuracy better than chance on 12 out of 25 data sets
- our method learns models that are better than chance on 21 out of 25

Analyzing Yeast Stress Data Sets

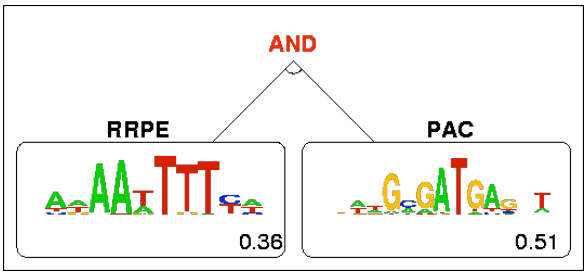
- three data sets, each associated with environmental stress response (ESR)
- our method attains statistically significant accuracy on all three
- one data set includes known promoter elements, which are recovered

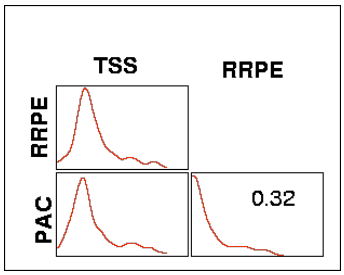


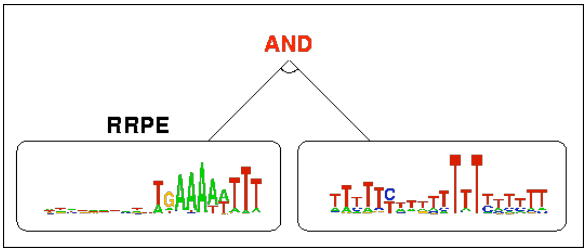


Does the Rich Representation Provide Value?

- consider a variant of our learner that cannot represent logical and spatial aspects ("bag of motifs" representation)



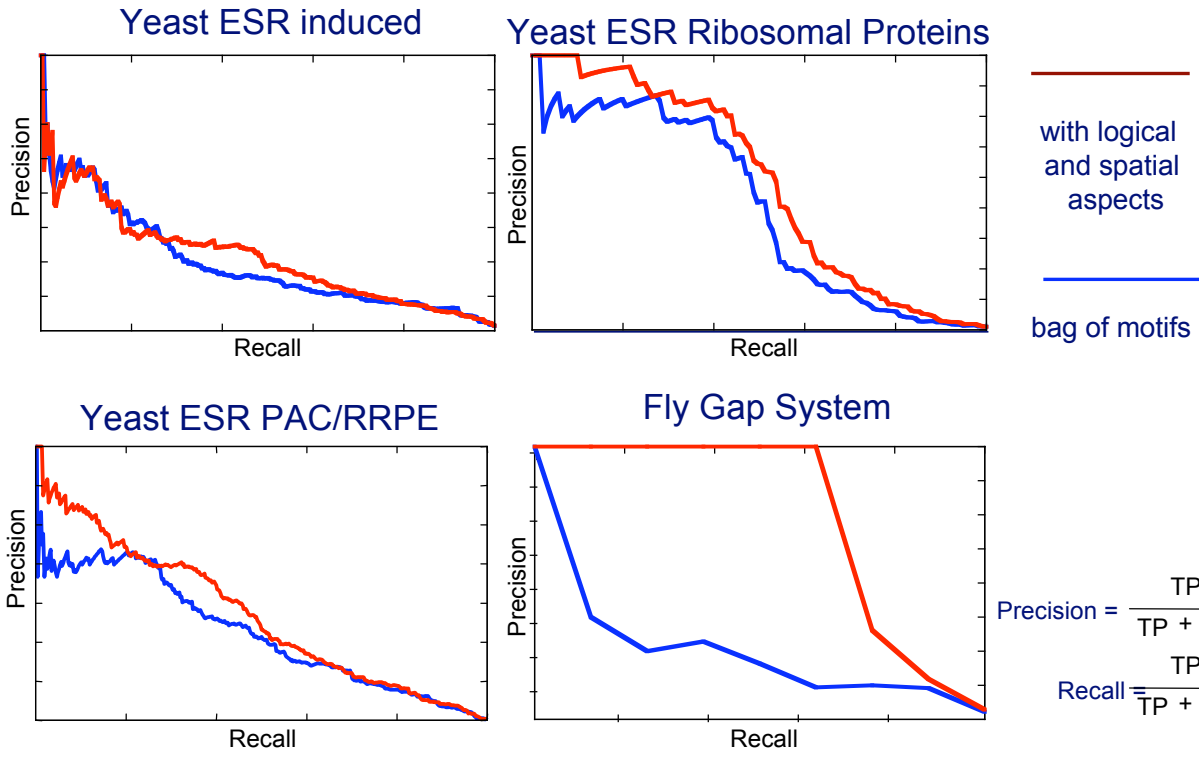




bag of motifs

- PAC element not recovered
- predictive accuracy worse

Does the Rich Representation Provide Value?



FIRE

- **Given** a set of sequences grouped into clusters
- **Find** motifs, and relationships, that have high *mutual information* with the clusters
- (also can do this when sequences have continuous values instead of cluster labels)

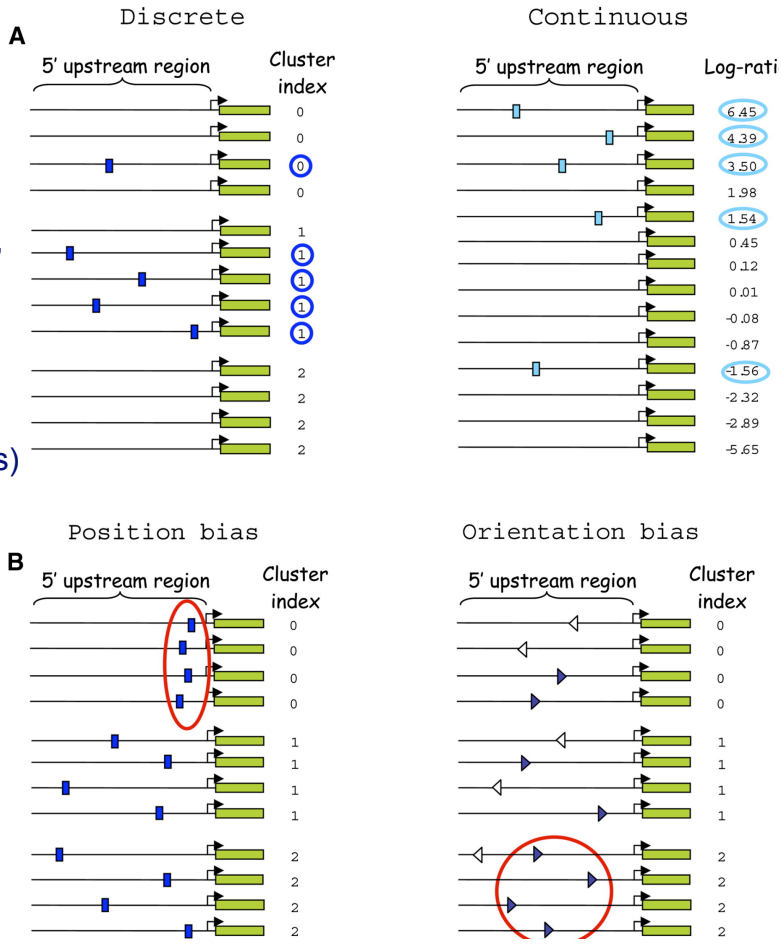


Figure from Elemento et al. *Molecular Cell* 2007

Mutual Information

- optimal code uses $-\log_2 P(c)$ bits for event with probability $P(c)$
- *entropy* is the expected value of the information required to specify the value of a random variable

$$H(C) = -\sum_{c=1}^{|C|} P(c)\log_2 P(c)$$

- *mutual information* quantifies how much knowing the value of one variable tells about the value of another

$$I(M;C) = H(M) - H(M \mid C)$$

Mutual Information

- we can compute the mutual information between a motif and the clusters as follows

$$I(M;C) = \sum_{m=0}^{1}\sum_{c=1}^{|C|} P(m,c)\log_2 \frac{P(m,c)}{P(m)P(c)}$$

m=0, 1 represent absence/presence of motif

c ranges over the cluster labels

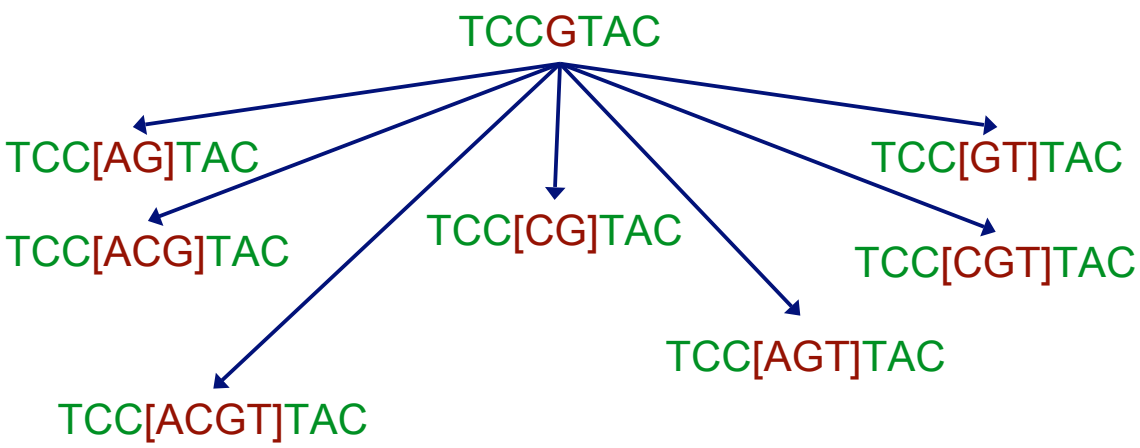
Finding Motifs in FIRE

- motifs are represented by regular expressions; initially each motif is represented by a strict *k*-mer (e.g. TCCGTAC)

1. test all *k*-mers (*k*=7 by default) to see which have significant mutual information with the class label
2. filter *k*-mers using a significance test
3. generalize each *k*-mer into a motif
4. filter motifs using a significance test

Key Step in Generalizing a Motif in FIRE

- randomly pick a position in the motif
- generalize in all ways consistent with current value at position
- score each by computing mutual information
- retain the best generalization



Generalizing a Motif in FIRE

given: *k*-mer, *n*

best ← null
repeat *n* times
    motif ← *k*-mer
    repeat
        motif ← GeneralizePosition(*motif*)   // shown on previous slide
    until convergence (no improvement at any position)
    if score(*motif*) > score(*best*)
        best ← *motif*

return: *best*

Generalizing a Motif in FIRE: Example

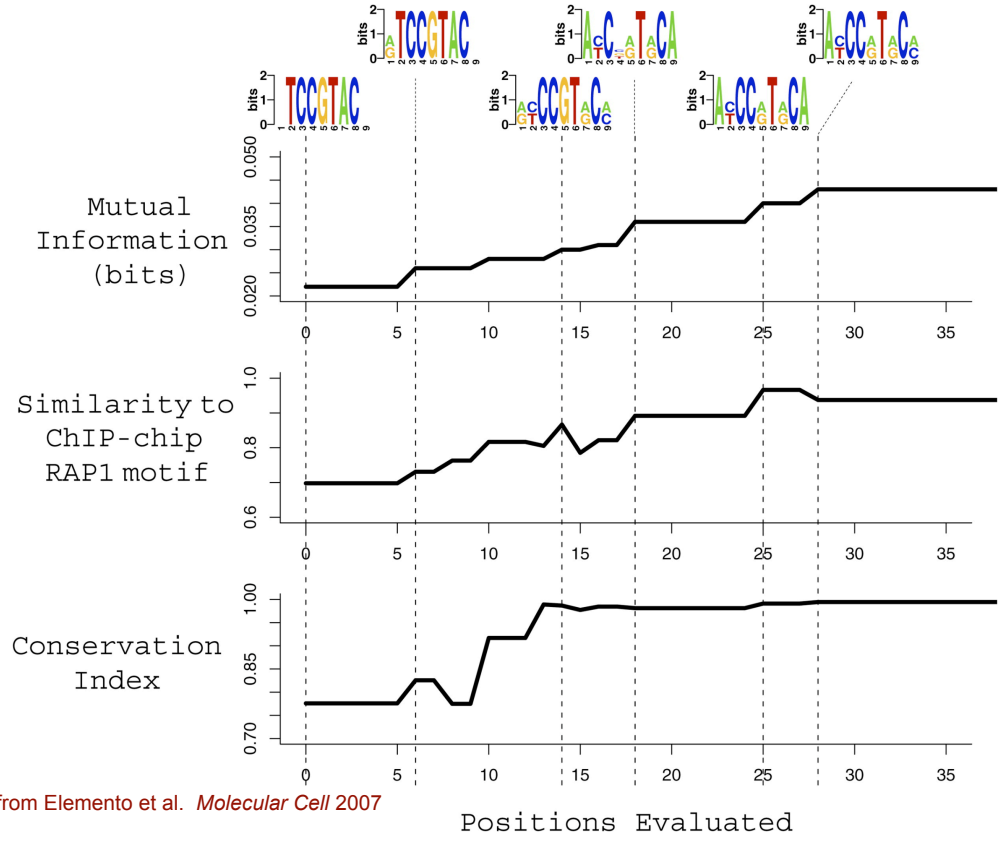


Figure from Elemento et al. *Molecular Cell* 2007

Characterizing Predicted Motifs in FIRE

- mutual information used to asses various properties of found motifs

orientation bias

$I(S;C)$ s=0, 1 indicates strand of given of motif

position bias

$I(P;O)$ P ranges over position bins, o=0, 1 indicates clusters in which the motif is overrepresented or not

interactions

$I(M_1;M_2)$ m_1=0, 1 indicates clusters in which motif 1 is overrepresented or not; similarly for m_2

Discussion

- Noto & Craven
  - HMM structure search to find CRM model
  - search operators apply to compact, logical representation instead of directly to HMM
  - employs generalized HMM approach to model *background* sequence lengths
- FIRE
  - mutual information used to identify motifs and relationships among them
  - motif search is based on generalizing informative *k*-mers
- in contrast to many motif-finding approaches, both CRM methods take advantage of *negative* sequences
- FIRE returns all informative motifs/relationships found, whereas the Noto & Craven approach returns single discriminative model