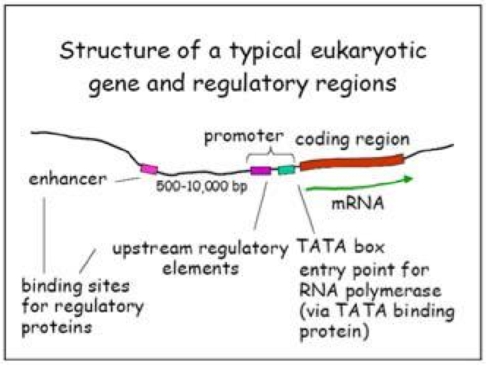
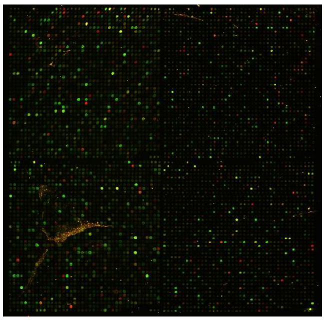
Lecture 3: PWMs and other models for regulatory elements in DNA

-Sequence motifs
-Position weight matrices (PWMs)
-Learning PWMs combinatorially, or probabilistically
    -Learning from an alignment
    -Ab initio learning: Baum-Welch & Gibbs sampling
-Visualization of PWMs as sequence logos
-Search methods for PWM occurrences
-Cis-regulatory modules

Regulatory motifs of DNA

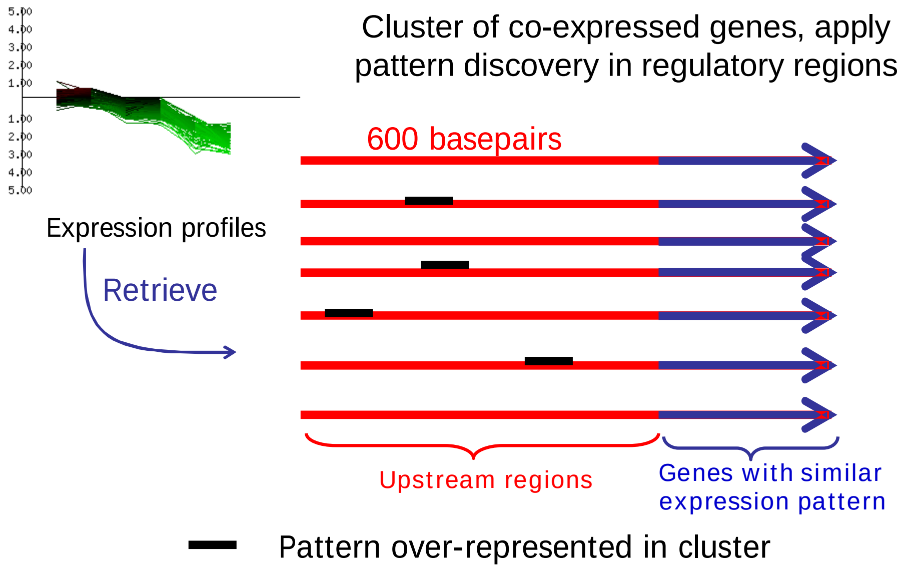

Structure of a typical eukaryotic gene and regulatory regions
promoter
coding region
enhancer
500-10,000 bp
mRNA
upstream regulatory elements
TATA box
entry point for RNA polymerase (via TATA binding protein)
binding sites for regulatory proteins
proteins



Measuring gene activity ('expression') with a microarray

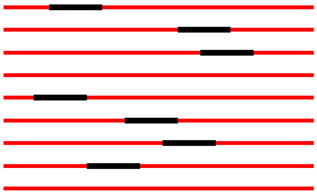
Jointly regulated gene sets

Cluster of co-expressed genes, apply pattern discovery in regulatory regions

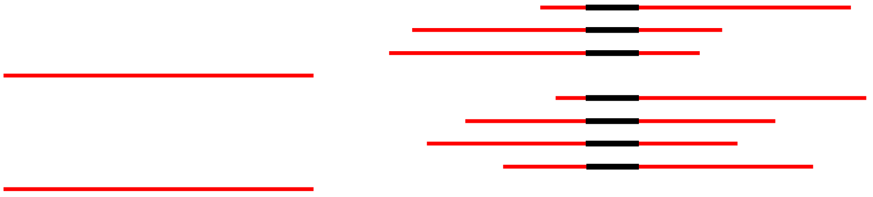


Find a hidden motif

Find a hidden motif



Find a hidden motif (cont.)



Multiple local alignment!

Smith-Waterman??

Motif?

- Definition: Motif is a pattern that occurs (unexpectedly) often in a string or in a set of strings

- The problem of finding repetitions is similar

```
cgccgagtgacagagacgctaatcagg
ctgtgttctcaggatgcgtaccgagtg
ggagacagcagcacgaccagcggtggc
agagacccttgcagacatcaagctctt
tgggaacaagtggagcaccgatgatgt
acagccgatcaatgacatttccctaat
gcaggattacattgcagtgcccaagga
gaagtatgccaagtaatacctccctca
cagtg...
```

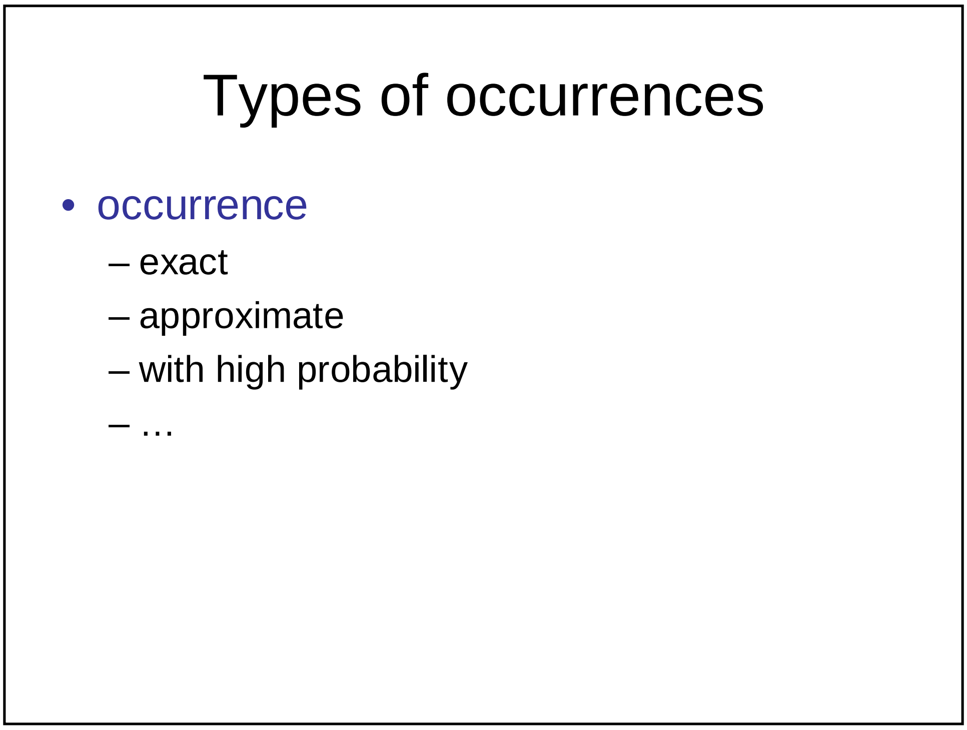
Longest repeat?

Example: A 50 million bases long fragment of human genome. We found (using a suffix-tree) the following repetitive sequence of length 2559, with one occurrence at 28395980, and other at 28401554r

```
ttagggtacatgtgcacaacgtgcaggtttgttacatatgtatacacgtgccatgatggtgtgctgcacccattaactcgtcatttagcgttaggtatatctcc
gaatgctatccctccccctcccccaccccacaacagtccccggtgtgtgatgttccccttcctgtgtccatgtgttctcattgttcaattcccacctatgagt
gagaacatgcggtgtttggttttttgtccttgcgaaagtttgctgagaatgatggtttccagcttcatccatatccctacaaaggacatgaactcatcatttttt
tatggctgcatagtattccatggtgtatatgtgccacattttcttaacccagtctacccttgttggacatctgggttggttccaagtctttgctattgtgaatag
tgccgcaataaacatacgtgtgcatgtgtctttatagcagcatgatttataatcctttgggtatatacccagtaatgggatggctgggtcaaatggtatttcta
gttctagatccctgaggaatcaccacactgacttccacaatggttgaactagtttacagtcccagcaacagttcctatttctccacatcctctccagcacctgt
tgtttcctgactttttaatgatcgccattctaactggtgtgagatggtatctcattgtggttttgatttgcatttctctgatggccagtgatgatgagcattttt
catgtgtttttggctgcataaatgtcttcttttgagaagtgtctgttcatatccttcgcccactttgatggggttgtttgttttttcttgtaaatttgttgga
gttcattgtagattctgggtattagccctttgtcagatgagtaggttgcaaaaatttctcccattctgtaggttgcctgttcactctgatggtggtttcttctgc
tgtgcagaagctctttagtttaattagatcccatttgtcaatttggcttttgttgccatagcttttggtgttttagacatgaagtccttgcccatgcctatgtcc
tgaatggtattgcctaggttttcttctagggtttttatggttttaggtctaacatgtaagtctttaatccatcttgaattaattataaggtgtatattataaggt
gtaattataaggtgtataattatatattaattataaggtgtatattaattataaggtgtaaggaagggatccagtttcagctttctacatatggctagccagtt
ttccctgcaccatttattaaatagggaatcctttccccattgcttgtttttgtcaggtttgtcaaagatcagatagttgtagatatgcggcattatttctgaggg
ctctgttctgttccattggtctatatctctgttttggtaccagtaccatgctgttttggttactgtagccttgtagtatagtttgaagtcaggtagcgtgatggtt
ccagctttgttcttttggcttaggattgacttggcaatgtgggctctttttggttccatatgaactttaaagtagtttttccaattctgtgaagaaattcattg
gtagcttgatggggatggcattgaatctataaattaccctgggcagtatggccattttcacaatattgaatcttcctacccatgagcgtgtactgttcttccatt
tgtttgtatcctctttatttcattgagcagtggtttgtagttctccttgaagaggtccttcacatcccttgtaagttggattcctaggtattttattctctttga
agcaattgtgaatgggagttcactcatgatttgactctctgtttgtctgttattggtgtataagaatgcttgtgatttttgcacattgattttgtatcctgagac
tttgctgaagttgcttatcagcttaaggagattttgggctgagacgatggggttttctagatatacaatcatgtcatctgcaaacagggacaatttgacttcct
cttttcctaattgaatacccgttatttccctctcctgcctgattgccctggccagaacttccaacactatgttgaataggagtggtgagagagggcatccctgt
cttgtgccagttttcaaagggaatgcttccagtttttgtccattcagtatgatattggctgtgggtttgtcatagatagctcttattattttgagatacatccca
tcaatacctaatttattgagagtttttagcatgaagagttcttgaattttgtcaaaggccttttctgcatcttttgagataatcatgtggtttctgtctttggttc
tgtttatatgctggagtacgtttattgattttcgtatgttgaaccagccttgcatcccagggatgaagcccacttgatcatggtggataagctttttgatgtgct
gctggattcggtttgccagtattttattgaggatttctgcatcgatgttcatcaaggatattggtctaaaattctcttttttgttgtgtctctgtcaggctttgg
tatcaggatgatgctggcctcataaaatgagttagg
```

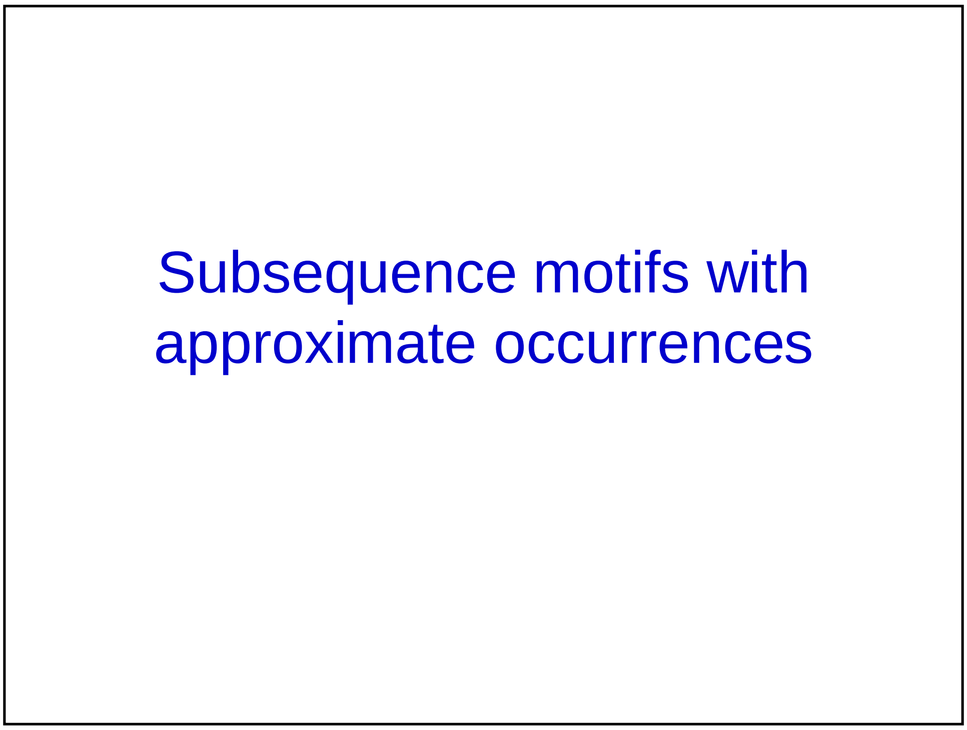
Representations of motifs

- pattern
  - substring
  - substring with gaps
  - string in generalized alphabet (e.g., IUPAC)
  - finite automaton
  - Hidden Markov Model
  - binding affinity matrix
  - cluster of binding affinity matrices
  - … (= the hidden structure to be learned from data)



Types of occurrences

- occurrence
  - – exact
  - – approximate
  - – with high probability
  - – …



Subsequence motifs with approximate occurrences

Motif finding problem

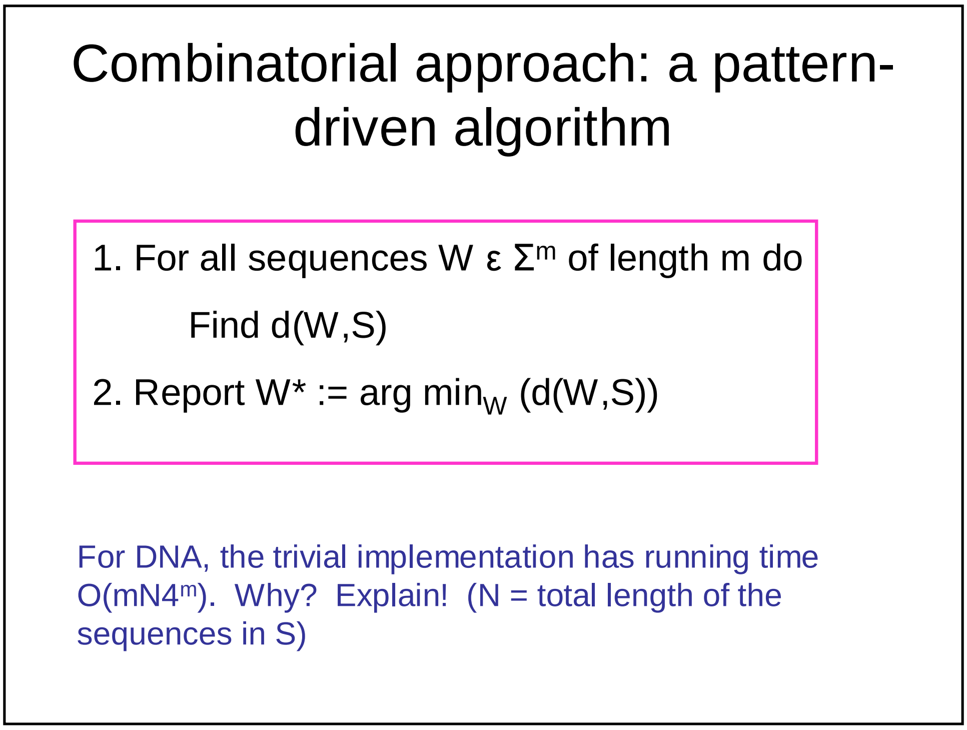
- Given: a set of sequences S = $x^{(1)},x^{(2)},...,x^{(n)}$ from alphabet *Σ,* and a motif length m
- Find the *best* motif of length m that *occurs* in the training data S
- Best? Occurs?

- **Combinatorial approach**: motif = sequence of length m
  - Approximate occurrences defined using Hamming distance (= number of mismatches)

- **Probabilistic approach**: motif = PWM (position weight matrix) of length m
  - Approximate occurrences defined using the probability distribution

Combinatorial approach

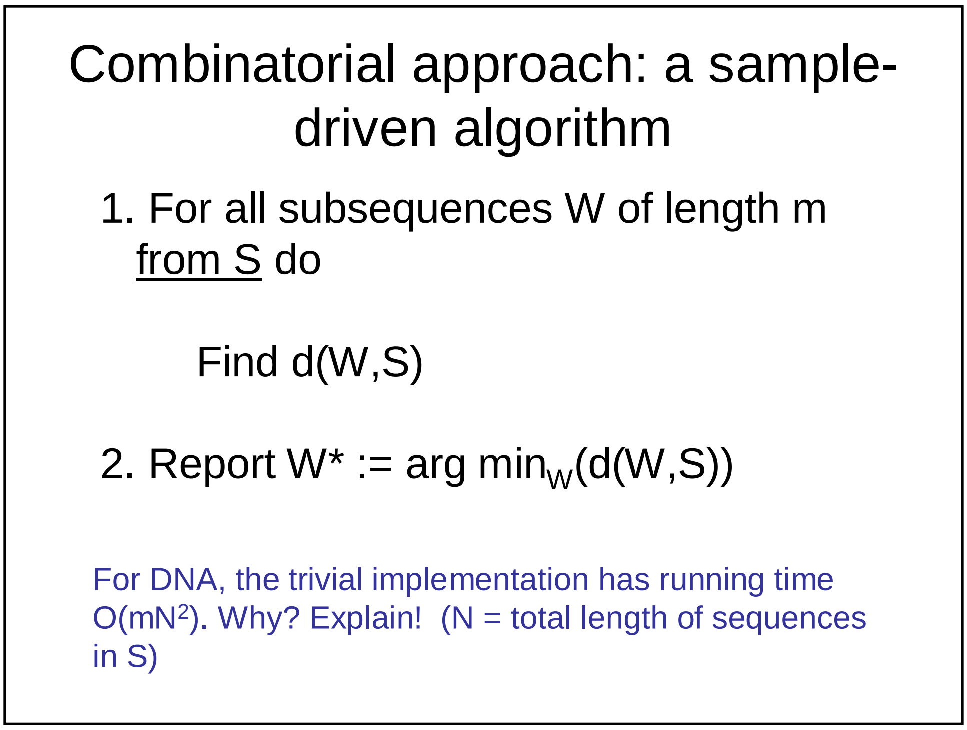
- $d(W,x^{(i)})$ = minimum Hamming distance between W and any subsequence of $x^{(i)}$ of length m
- $d(W,S) = \sum_i d(W,x^{(i)})$



Combinatorial approach: a pattern-driven algorithm

```
1. For all sequences $W \in \Sigma^m$ of length m do
        Find d(W,S)
2. Report $W^* := \arg\min_W (d(W,S))$
```

For DNA, the trivial implementation has running time $O(mN4^m)$. Why? Explain! (N = total length of the sequences in S)



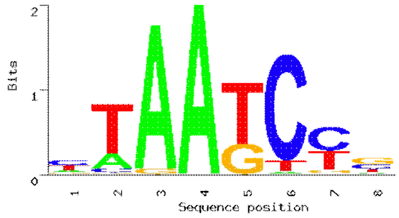
Combinatorial approach: a sample-driven algorithm

```
1. For all subsequences W of length m from S do
        Find d(W,S)
2. Report $W^* := \arg\min_W(d(W,S))$
```

For DNA, the trivial implementation has running time $O(mN^2)$. Why? Explain! (N = total length of sequences in S)

Probabilistic approach

- Learn Position Weight Matrices (PWMs) from data
- Probabilistic model (in fact, a simple HMM)



Positionally weighted pattern

- Weighted pattern *w* = *(w$_{ij}$)* of length *m* in alphabet Σ: |Σ| x m matrix of real-valued scores
- Also called as: position weight matrix (PWM), position-specific scoring matrix (PSSM), profile(-HMM), motif, ... (Stormo et al 1980, Gribskov et al 1987, Henikoff et al 1990,...)
- Public collections of PWMs: TRANSFAC, JASPAR

	1	2	3	4	5	6
A	0.3	0.0	0.1	0.2	1.0	0.3
C	0.1	0.8	0.5	0.2	0.0	0.4
G	0.2	0.0	0.4	0.3	0.0	0.0
T	0.4	0.2	0.0	0.3	0.0	0.3

Construction of PWM from an alignment

- Given:
  - – aligned set of binding sites (= sequences of length m)
  - – background distribution q_a for a є Σ

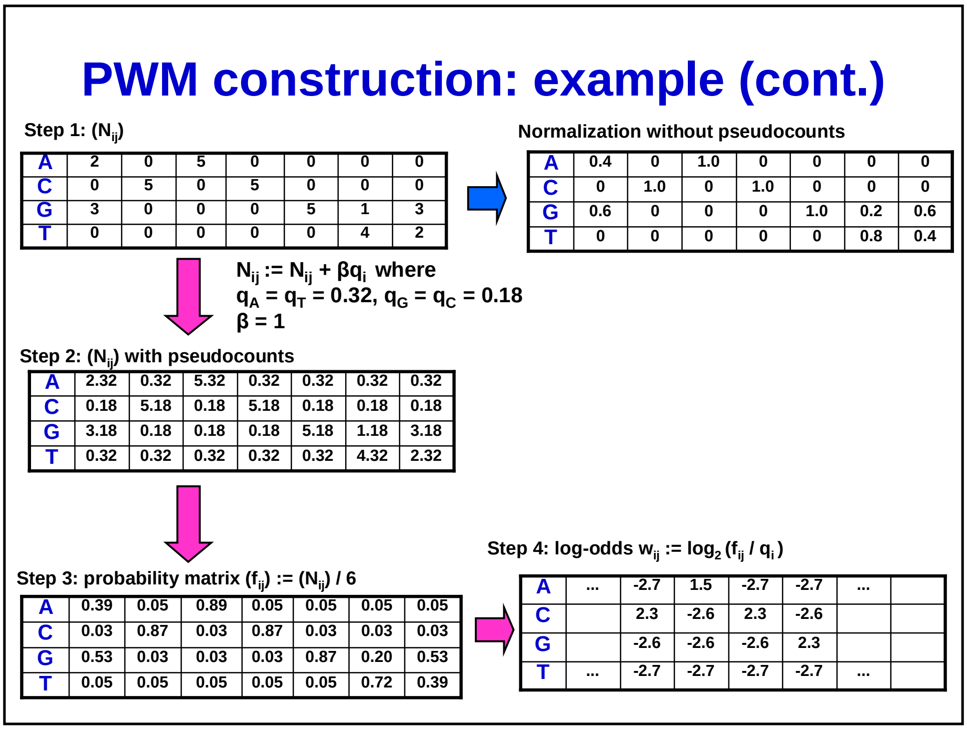
- **PWM construction algorithm:**
  1. **Construct count matrix (N_{ij}): N_{ij} := the number of symbols i є Σ on column j of the aligment**
  2. **Add pseudocounts:**
     - **N_{ij} := N_{ij} + 1 (= Laplace rule), or**
     - **N_{ij} := N_{ij} + βq_i where β is a scaling parameter that determines the total number of pseudocounts in an alignment column**
  3. **Construct probability matrix (f_{ij})by normalizing the (pseudo)counts: f_{ij} := N_{ij} / $\sum_i N_{ij}$**
  4. **Construct log-odds of signal vs background: w_{ij} := log(f_{ij} / q_i) (= weighted pattern *w*)**

PWM construction: example

```
 CTCACACGTGGG
  TCACACGTGGGA
ATTAGCACGTTTT
 TTAGCACGTTTCGC
CGCTGCACGGGGCC
```

(N_{ij}):

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



PWM construction: example (cont.)

Step 1: (N_{ij})

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

Normalization without pseudocounts

A	0.4	0	1.0	0	0	0	0
C	0	1.0	0	1.0	0	0	0
G	0.6	0	0	0	1.0	0.2	0.6
T	0	0	0	0	0	0.8	0.4

$N_{ij} := N_{ij} + \beta q_i$ where
$q_A = q_T = 0.32$, $q_G = q_C = 0.18$
$\beta = 1$

Step 2: (N_{ij}) with pseudocounts

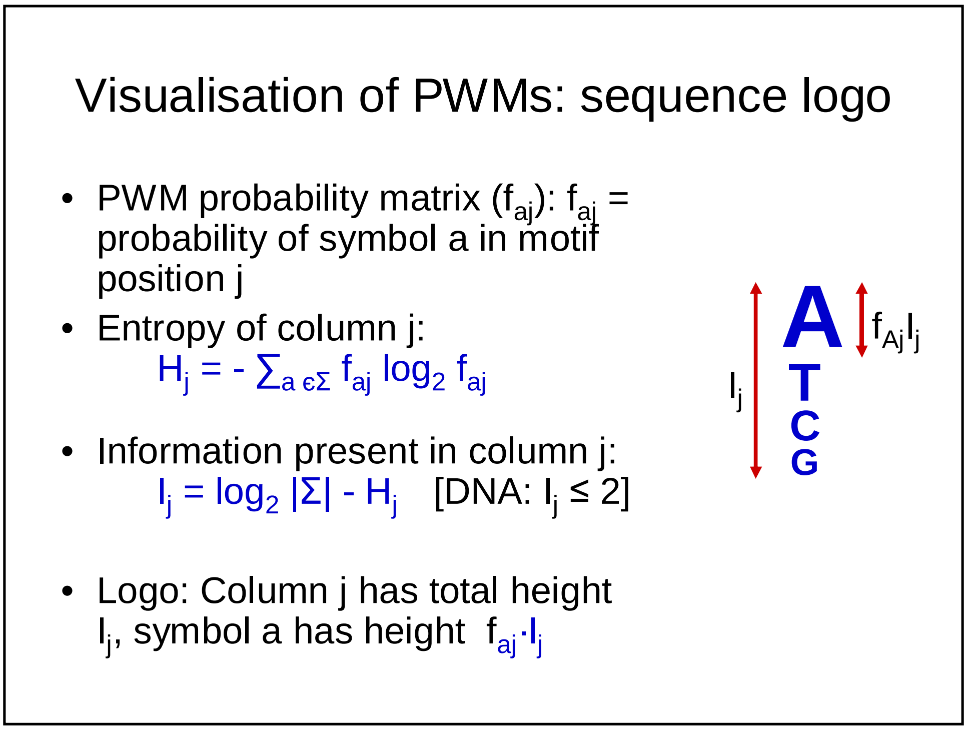
A	2.32	0.32	5.32	0.32	0.32	0.32	0.32
C	0.18	5.18	0.18	5.18	0.18	0.18	0.18
G	3.18	0.18	0.18	0.18	5.18	1.18	3.18
T	0.32	0.32	0.32	0.32	0.32	4.32	2.32

Step 3: probability matrix $(f_{ij}) := (N_{ij}) / 6$

A	0.39	0.05	0.89	0.05	0.05	0.05	0.05
C	0.03	0.87	0.03	0.87	0.03	0.03	0.03
G	0.53	0.03	0.03	0.03	0.87	0.20	0.53
T	0.05	0.05	0.05	0.05	0.05	0.72	0.39

Step 4: log-odds $w_{ij} := \log_2 (f_{ij} / q_i)$

A	...	-2.7	1.5	-2.7	-2.7	...	
C		2.3	-2.6	2.3	-2.6		
G		-2.6	-2.6	-2.6	2.3		
T	...	-2.7	-2.7	-2.7	-2.7	...	



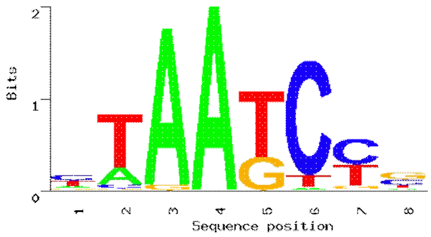
Visualisation of PWMs: sequence logo

- PWM probability matrix (f_{aj}): f_{aj} = probability of symbol a in motif position j
- Entropy of column j:
  $H_j = - \sum_{a \in \Sigma} f_{aj} \log_2 f_{aj}$
- Information present in column j:
  $I_j = \log_2 |\Sigma| - H_j$ [DNA: $I_j \leq 2$]
- Logo: Column j has total height I_j, symbol a has height $f_{aj} \cdot I_j$

I_j
A
T
C
G
$f_{Aj} I_j$

Sequence logo: an example

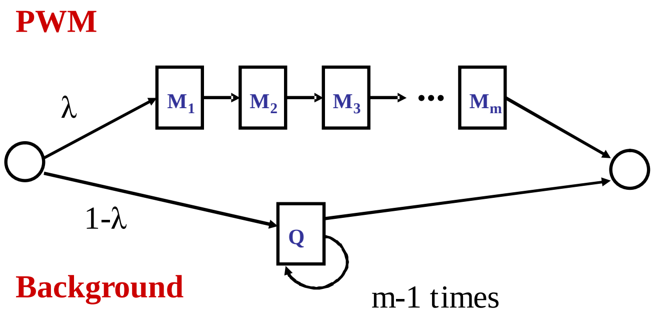
A	9	11	49	51	0	1	1	4
C	19	3	0	0	0	45	25	16
G	5	1	2	0	17	0	4	21
T	18	36	0	0	34	5	21	10



Ab initio construction of PWMs

- No alignment given, just the training sequences $S = x^{(1)}, ..., x^{(N)}$
- Find from S all m-long words
- Assume each word comes either from motif or from background (don’t care about overlaps!)
- Find the most likely
  - motif model (PWM),
  - background model, and
  - classification of the m-long words of S into motif and background instances

HMM for mixture of multinomials



PWM

λ

M_1 M_2 M_3 … M_m

1-λ

Q

Background

m-1 times

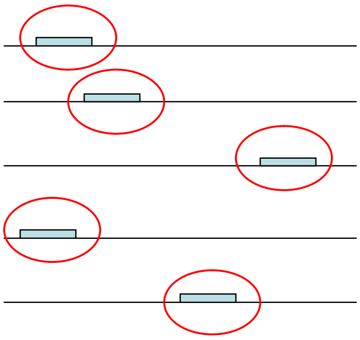
PWM by the EM algorithm

- Train the mixture model using EM algorithm by iterating
  - E step
  - M step
- Training data: all m-words from S
- For more details, see:
  - T.L. Bailey & C. Elkan: The value of prior knowledge in discovering motifs with MEME
  - Other papers on MEME

Ab initio motif finding: Gibbs sampling

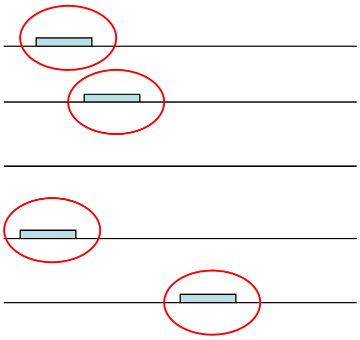
- Another popular probabilistic algorithm for motif (PWM) discovery
- Local search algorithm

Gibbs sampling: basic idea



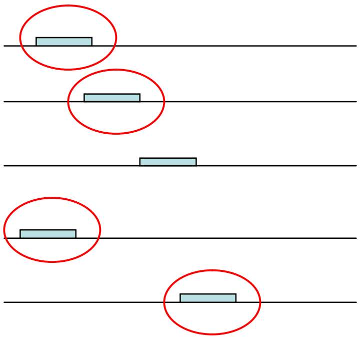
Current motif = PWM formed by circled substrings

Gibbs sampling: basic idea

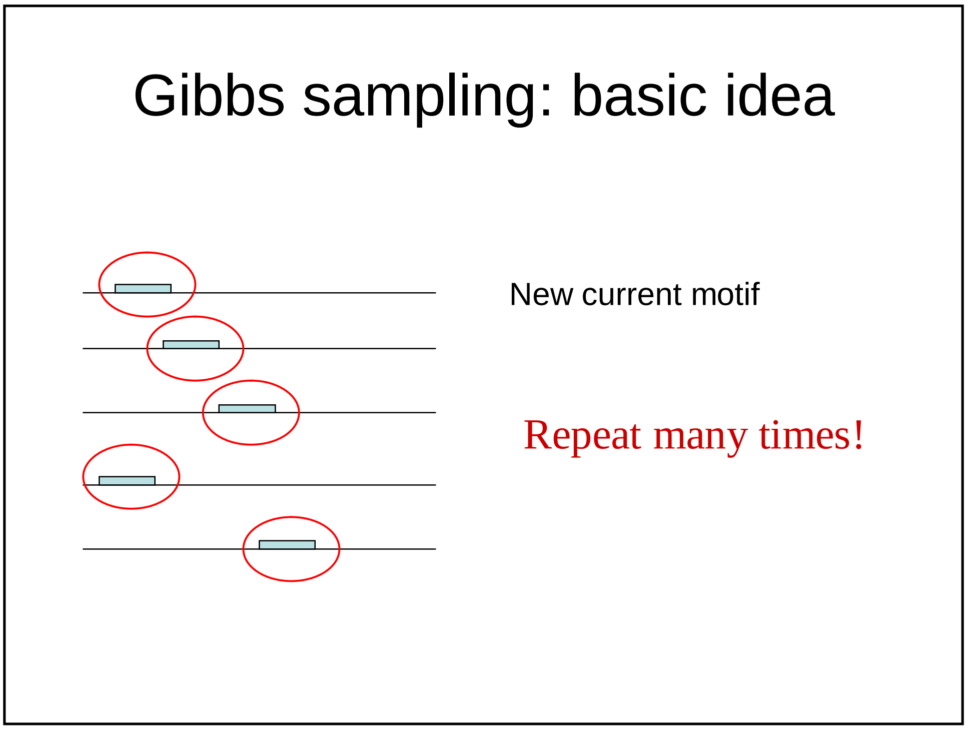


Delete one substring

Gibbs sampling: basic idea



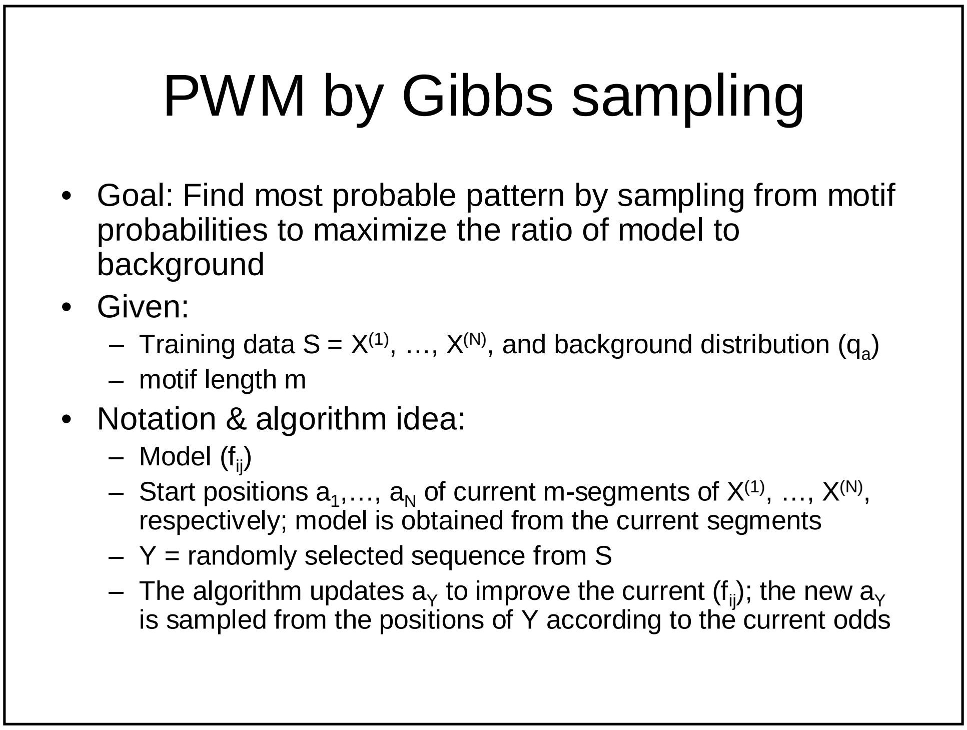
Try a replacement: Compute its score, and accept the replacement depending on the score.



Gibbs sampling: basic idea

New current motif

Repeat many times!



PWM by Gibbs sampling

- Goal: Find most probable pattern by sampling from motif probabilities to maximize the ratio of model to background
- Given:
  - Training data S = $X^{(1)}$, …, $X^{(N)}$, and background distribution (q_a)
  - motif length m
- Notation & algorithm idea:
  - Model (f_{ij})
  - Start positions a_1,…, a_N of current m-segments of $X^{(1)}$, …, $X^{(N)}$, respectively; model is obtained from the current segments
  - Y = randomly selected sequence from S
  - The algorithm updates a_Y to improve the current (f_{ij}); the new a_Y is sampled from the positions of Y according to the current odds

Gibbs sampling algorithm

```
• For i := 1, ...,N do
      $a_i$ := random(1,..., $|x^{(i)}|$-m+1)
• Repeat many times (say, 100N times)
      Y := random(1, ..., N)
      for i in {A,C,G,T} and for j := 1, ..., m do
          $f_{ij} := (c_{ij} + \beta q_i) / (N - 1 + \beta)$  where
              $c_{ij}$ = the number i's in the column j of the alignment of the N-1 m-segments that for $1 \le h \le N$, $h \ne Y$, start at position $a_h$ of sequence $x^{(h)}$, and
              $\beta$ is the weight for the pseudocounts (say, $\beta = N^{1/2}$)
    end forfor
    $a_y$ := weighted-random(1,...,|y| - m +1), where y denotes the sequence $x^{(Y)}$, and the weight of position i in this random selection is
        $f_{y(i),1}\ f_{y(i+1),2} \ldots f_{y(i+m-1),m} / q_{y(i)}\ q_{y(i+1)} \ldots q_{y(i+m-1)}$
        (=the odds of position i being a motif versus background)
  end repeat
• Output ($f_{ij}$)
```

Empirical comparison of PWM learning tools

- M. Tompa, N. Li, T. Bailey et al.: Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23:137-144, 2005.

Search methods to find good occurrences of PWMs

Segment score by a PWM w

	1	2	3	4	5	6
A	0.3	0.0	0.1	0.2	1.0	0.3
C	0.1	0.8	0.5	0.2	0.0	0.4
G	0.2	0.0	0.4	0.3	0.0	0.0
T	0.4	0.2	0.0	0.3	0.0	0.3

s_1 s_2 s_3 s_4 s_5 s_6

G T A C A C

Score = 2.1

$$\boxed{Score = \sum_{i=1}^{m} w[s_i, i]}$$

Significance thresholding

- Assume that we have evaluated the score by w for every m-segment of a sequence S. ***When is a segment score significant?***

- Background distribution of K-segments $u = u_1 \ldots u_m$:
  $$Prob(u) = q(u_1) \ldots q(u_m)$$

- Statistical testing: for a p-value *p*, the corresponding score threshold $k = k(p)$ is a value such that in the background distribution
  $$Prob(u : Score_w(u) \geq k) = p$$

- If $Score_w(u) \geq k$, then the score of *u* differs from the background on significance level *p*

Pattern Matching Problem

- Text *S* in alphabet *Σ*, length *n*
- $|\Sigma|$ *x m* weighted pattern *w*
- Score threshold $k = k(p)$

- **Problem: Find all positions *i* of *S* such that the score given by pattern w for the m-segment starting at *i* is above the threshold *k***

Example: k = 2

S = **CGTACACTCGGTA**

	1	2	3	4	5	6
A	0.3	0.0	**0.1**	0.2	**1.0**	0.3
C	0.1	0.8	0.5	**0.2**	0.0	**0.4**
G	**0.2**	0.0	0.4	0.3	0.0	0.0
T	0.4	**0.2**	0.0	0.3	0.0	0.3

Score = 2.1

Match at pos 2

Two basic algorithms

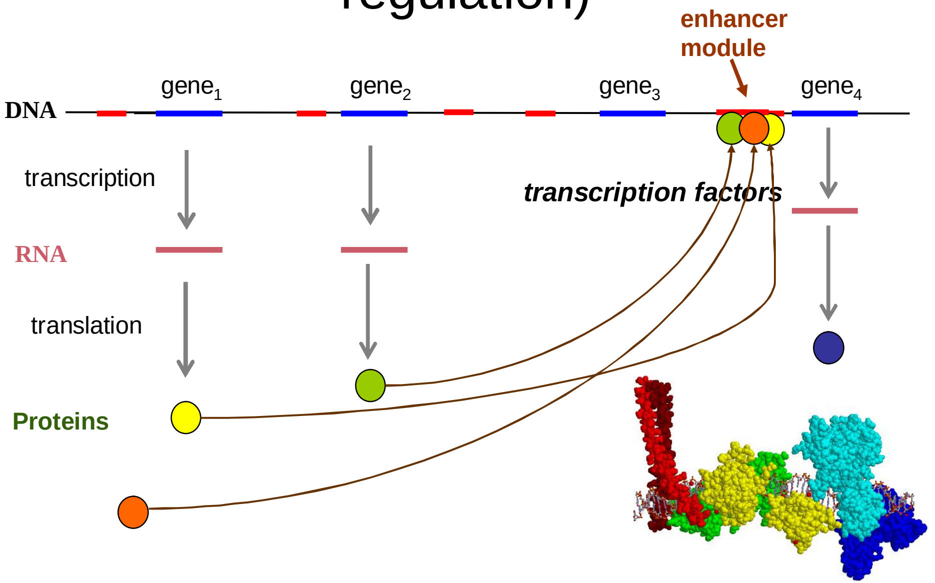
- Naive algorithm (NA): O(mn)
- Lookahead scoring:
  - For each column of w, precompute the maximum possible score that can be accumulated after the column ('lookahead score')
  - Stop checking the current m-segment as soon as it is clear that k cannot be achieved (i.e., if current score + lookahead < k)
- Permuted lookahead scoring [Wu 2000] (PLS):
  - evaluate the columns of M in the order of decreasing expected loss

$$L_j = \max_a w(a,j) - \mathbf{E}_q(w(a,j))$$

Cis-regulatory modules: a Higher-Order Motif Finding Problem

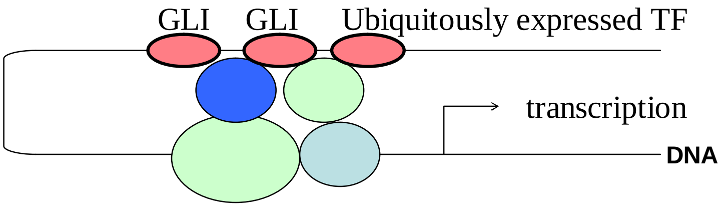
- Usually more than one motif is involved in regulation. Also, there are many regulatory proteins that control the expression of a gene, and the set of regulatory proteins involved is different under different situations.
- Cross-species sequence alignment (*phylogenetic footprinting*)

Gene regulatory modules (cis-regulation)

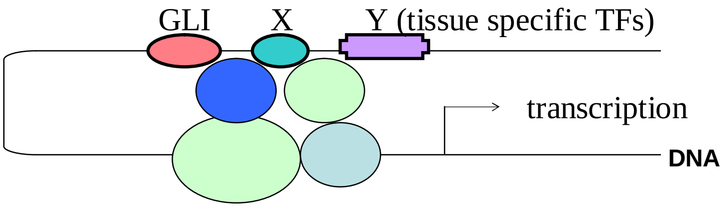


Model of cell type specific regulation of target gene expression

Common targets (e.g. Patched):



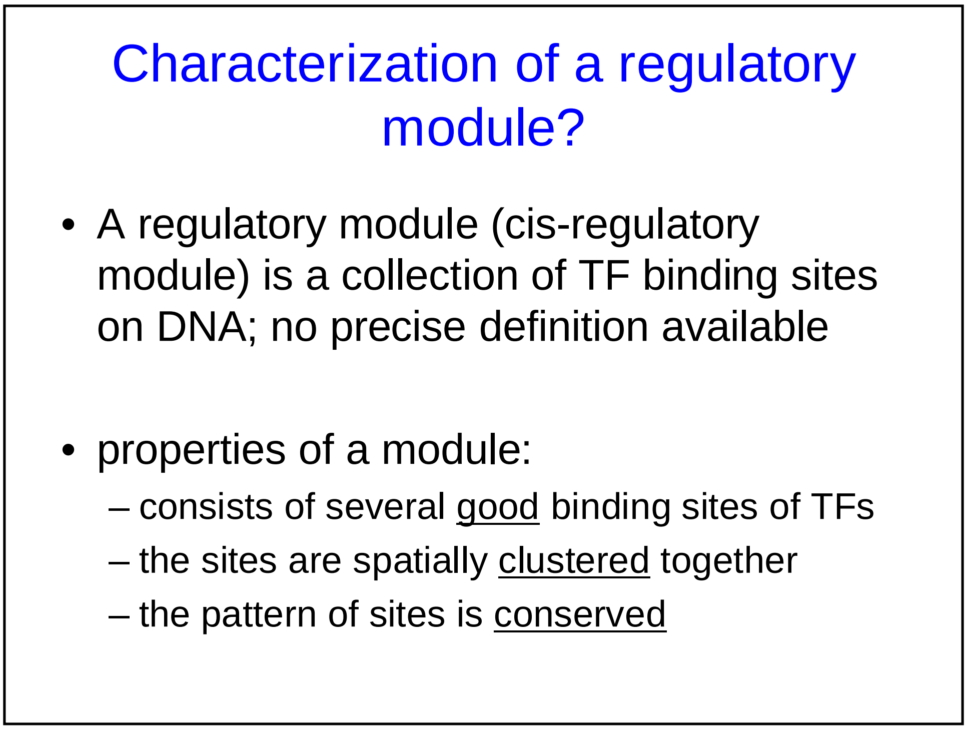
Cell type specific targets (e.g. N-myc):



Some vague remarks

- Gene expression regulation in multicellular organisms is controlled in combinatorial fashion by *transcription factors* (TFs)
- Transcription factors bind to DNA cis-elements on enhancer modules (promoters)
- Multiple factors need to bind to activate the module
- In mammals, the modules are few and far

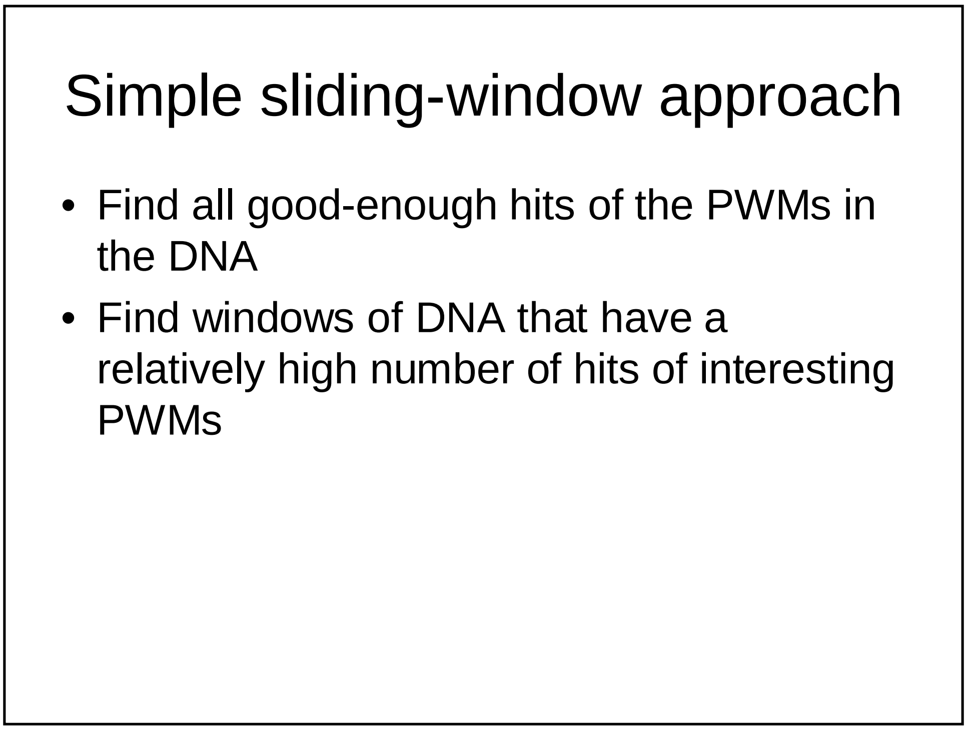
- **The problem**: Locate functional regulatory modules, that is, find **interesting patterns** of TF binding sites in DNA.



Characterization of a regulatory module?

- A regulatory module (cis-regulatory module) is a collection of TF binding sites on DNA; no precise definition available

- properties of a module:
  - consists of several good binding sites of TFs
  - the sites are spatially clustered together
  - the pattern of sites is conserved



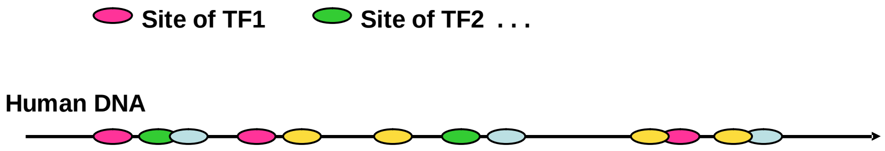
Simple sliding-window approach

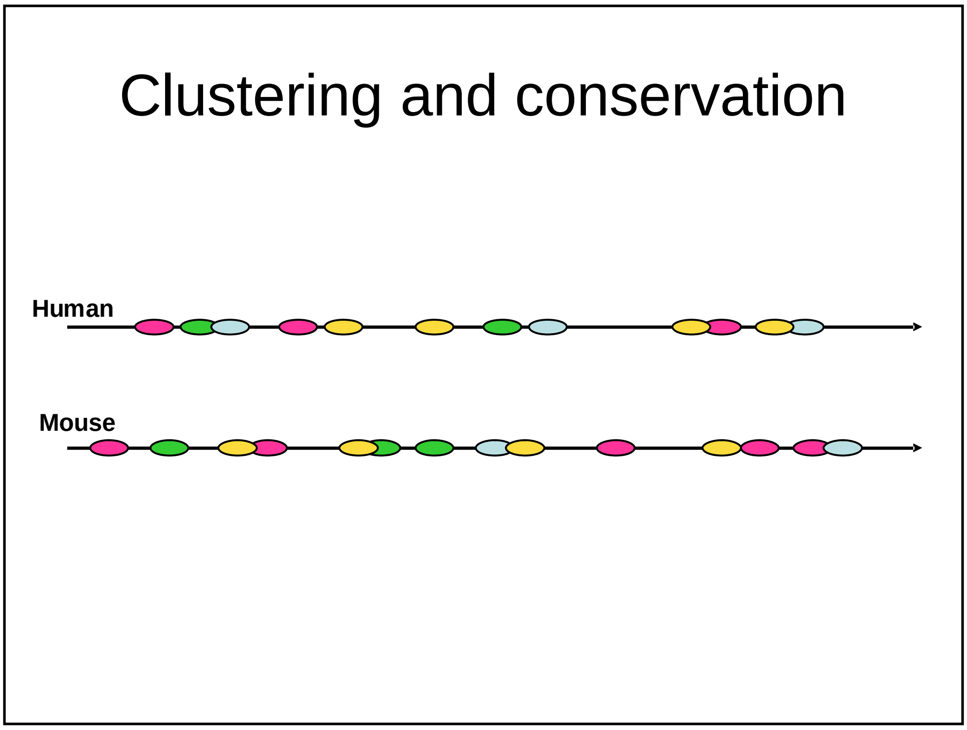
- Find all good-enough hits of the PWMs in the DNA
- Find windows of DNA that have a relatively high number of hits of interesting PWMs

Phylogenetic footprinting: find conserved motifs of binding sites

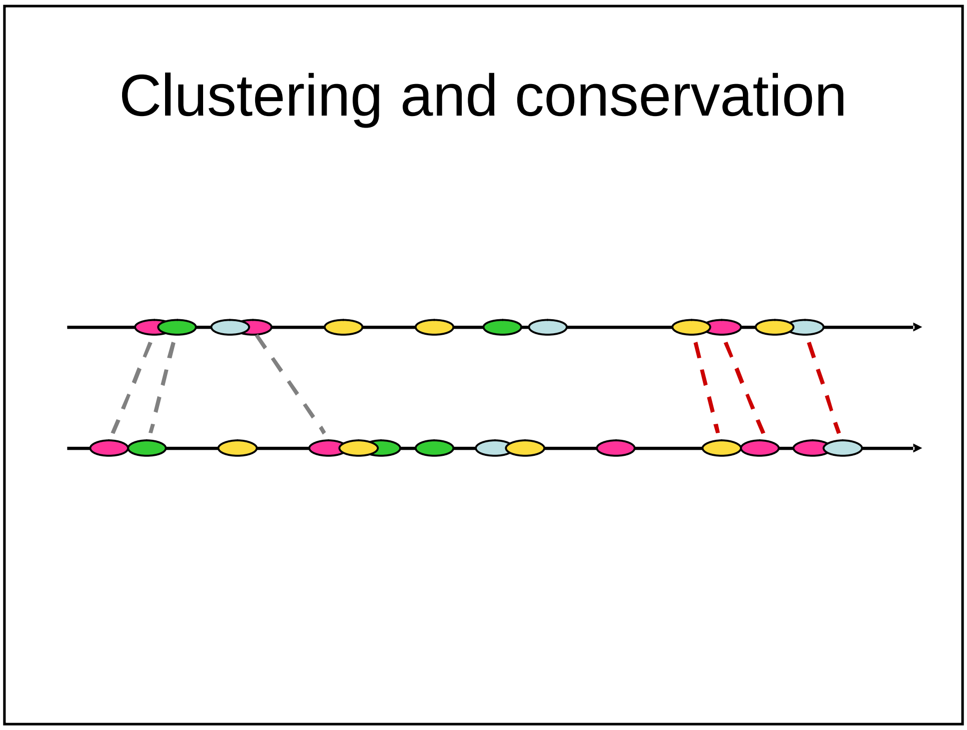
- looking at one (human) genome gives too many positives
- comparative approach (phylogenetic footprinting):
  - take (say) the 200 kbp regions surrounding the same genes (paralogs and orthologs) of different organisms: human, mouse, chicken, …
  - find conserved clusters (subsequence motifs) of binding sites
- Smith-Waterman type algorithm with a novel scoring function

Good binding sites

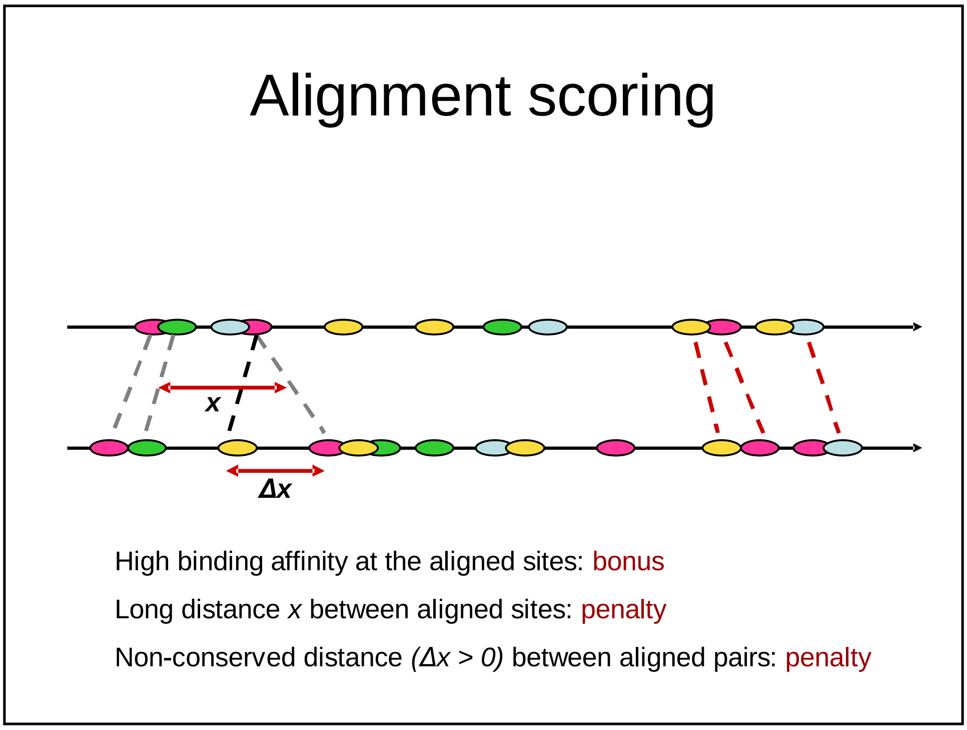




Clustering and conservation



Clustering and conservation

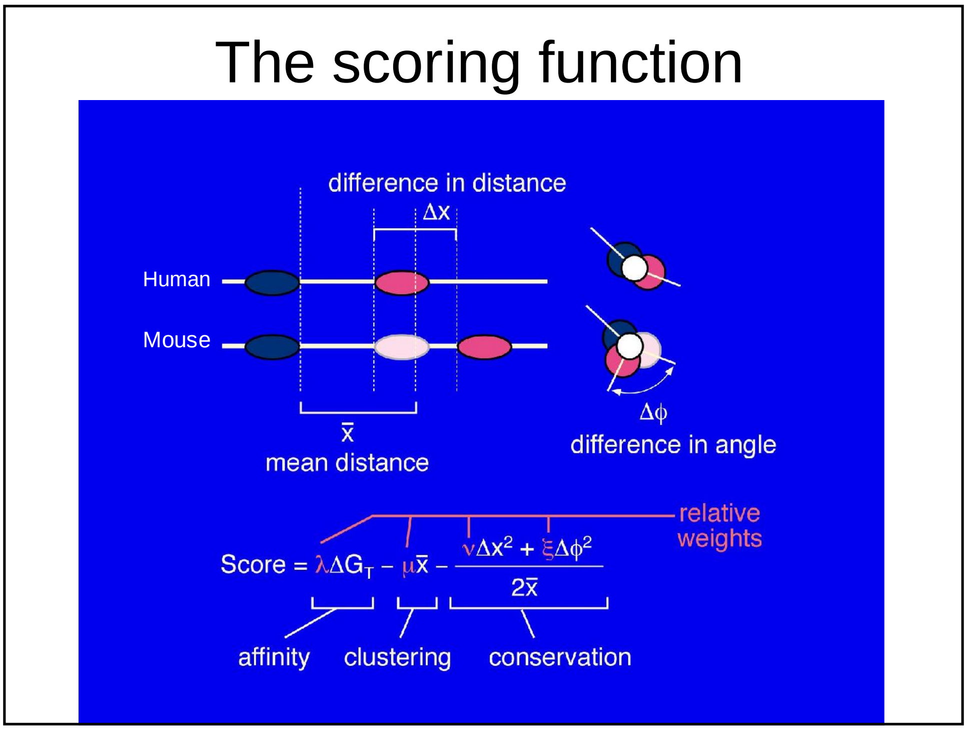


Alignment scoring

High binding affinity at the aligned sites: bonus

Long distance *x* between aligned sites: penalty

Non-conserved distance *(Δx > 0)* between aligned pairs: penalty

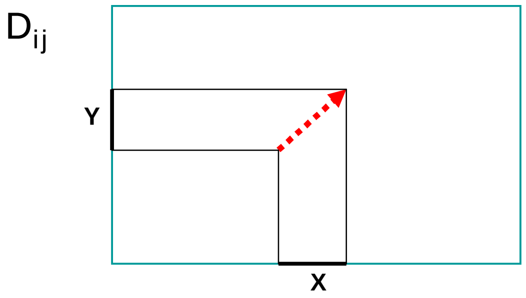


The scoring function

$$\text{Score} = \underbrace{\lambda \Delta G_T}_{\text{affinity}} - \underbrace{\mu \bar{x}}_{\text{clustering}} - \underbrace{\frac{\nu \Delta x^2 + \xi \Delta \phi^2}{2\bar{x}}}_{\text{conservation}}$$

Smith-Waterman

- find the best local alignment of strings A and B: *subsequence* X of A and *subsequence* Y of B such that X and Y have the best scoring pairwise alignment



Dynamic programming

$$D_{ij} = \begin{cases} \max\{\lambda w_{ij}, D_{k,l} + \lambda w_{ij} - F(p_i - q_k\ , p'_j - q'_l) \mid \\ \qquad 0 < p_i - q_k < 1000,\ 0 < p'_j - q'_l < 1000\}, \\ \qquad \text{if } f_i = f'_j \text{ (i.e., the same TF aligned)} \\ -\infty, \quad \text{otherwise} \end{cases}$$

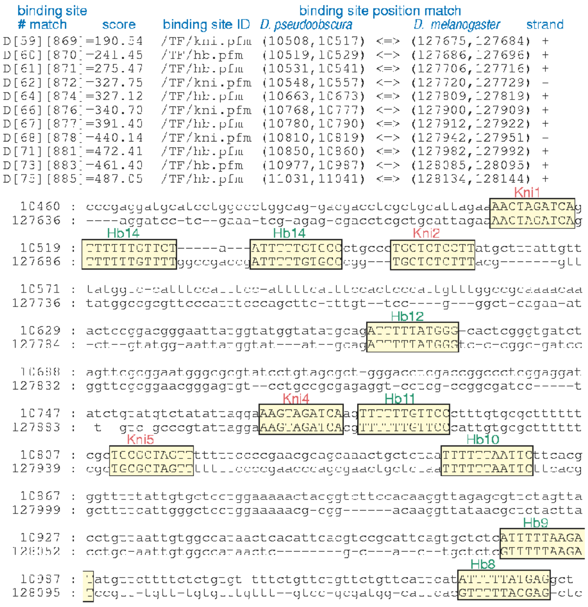
$O(n^4)$

w_{ij} = sum of the binding affinities of the sites of the TF at i and j in the two sequences

$F(\Delta i, \Delta j)$ = penalty for the non-conservation and the length of the distances between adjacent sites

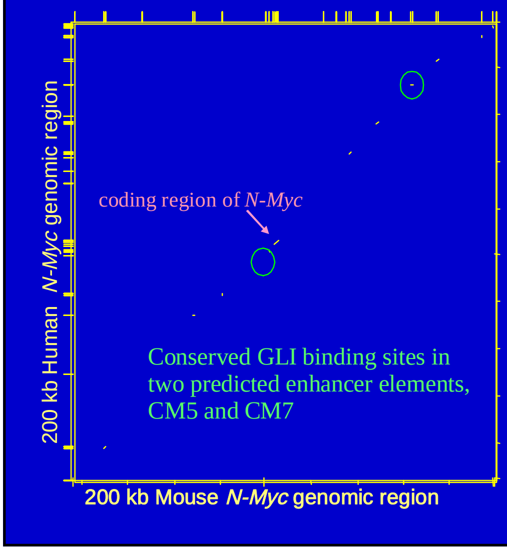
Drosophila enhancer

- **Output from EEL (Enhancer Element Locator) program**
- *Drosophila* even-skipped gene stripe 2 enhancer
- Score = 487.05



```
binding site                                      binding site position match
# match       score     binding site ID  D. pseudoobscura         D. melanogaster   strand
D[59][869]=190.34  /TF/kni.pfm (10508,10517) <=> (127675,127684) +
D[60][870]=241.45  /TF/hb.pfm  (10519,10529) <=> (127686,127696) +
D[61][871]=275.47  /TF/hb.pfm  (10531,10541) <=> (127706,127716) +
D[62][872]=327.75  /TF/kni.pfm (10548,10557) <=> (127720,127729) -
D[64][874]=327.12  /TF/hb.pfm  (10663,10673) <=> (127809,127819) +
D[66][876]=340.70  /TF/kni.pfm (10768,10777) <=> (127900,127909) +
D[67][877]=391.40  /TF/hb.pfm  (10780,10790) <=> (127912,127922) +
D[68][878]=440.14  /TF/kni.pfm (10810,10819) <=> (127942,127951) -
D[71][881]=472.41  /TF/hb.pfm  (10850,10860) <=> (127982,127992) +
D[73][883]=461.40  /TF/hb.pfm  (10977,10987) <=> (128085,128095) +
D[75][885]=487.05  /TF/hb.pfm  (11031,11041) <=> (128134,128144) +

                                                              Kni1
 10460 : cccgaggatgcatcctggcccctggcag-gacgacctcgctgcattagaaAACTAGATCAg
127636 : ----aggatcc-tc--gcaa-tcg-agag-cgacctcgctgcattagaaAACTAGATCAg
               Hb14               Hb14              Kni2
 10519 : TTTTTTGTTCT-----a---ATTTTTGTCCCctgcccTGCTCTCCTTatgcttttattgtt
127686 : TTTTTTGTTTTggccgaccgATTTTTGTGCCcgg---TGCTCTCTTTacg-------gtt

 10571 : tatgg-c-catttcca--tcc-attttca-ttccactccca-tgtttggccgcaaaacaa
127736 : tatggccgcgttcccatttcccagcttc-tttgt--tcc----g---ggct-cagaa-at
                                                   Hb12
 10629 : actccggacgggaattatggtatggtatatgcagATTTTTATGGG-cactcgggtgatct
127784 : -ct--gtatgg-aattatggtat---at--gcagATTTTTATGGGtc-c-cggc-gatcc

 10688 : agttcgcggaatgggcgcgtatcctgtagcgct-gggacctcgaccggccctcggaggat
127832 : ggttcgcggaacgggagtgt--cctgccgcgagaggt-cctcg-ccggcgatcc-----t
                                   Kni4        Hb11
 10747 : atctgtatgtctatattaggaAAGTAGATCAagTTTTTTGTTCCctttgtgcgcttttt
127883 :  t  gtc gcccgtattaggaAAGTAGATCAcgTTTTTTGTTCCcattgtgcgcttttt
             Kni5                                        Hb10
 10807 : cgcTCCGCTAGTTttttttccccgaacgcagcaaactgctcttaTTTTTTAATTCttcacg
127939 : cgcTGCGCTAGTTttttcccccgaacccagcgaactgctcttaTTTTTTAATTCttcacg

 10867 : ggtttttattgtgctccttggaaaaactacggtcttccacaaggttagagcgttctagtta
127999 : gcttttcattgggctccttggaaaaacg-cgg------acaaggttataacgctctactta
                                                              Hb9
 10927 : cctgttaattgtggccataactcacattcacgtccgcattcagtgctctcATTTTTAAGA
128052 : cctgc-aattgtggccataactc--------g-c---a--c--tgctctcGTTTTTAAGA
                                                      Hb8
 10987 : tatgttcttttctctgtgt tttctgttctgttctgttcattcatATTTTTATGAGgct
128095 : tccg---tgtt-tgtg--tgttt--gtcc-gcgatgg-ca--catGTTTTTACGAG-ctc
```

Enhancer prediction for *N-myc*

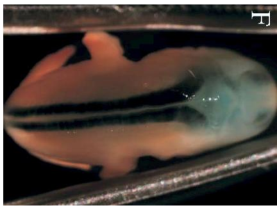


coding region of *N-Myc*

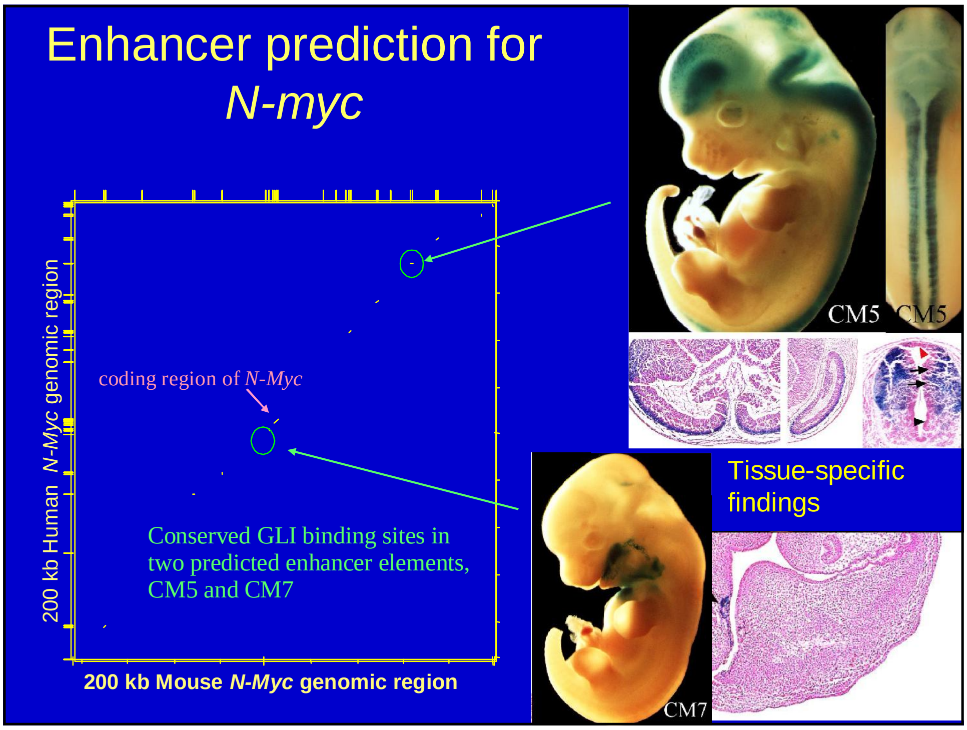
Conserved GLI binding sites in two predicted enhancer elements, CM5 and CM7

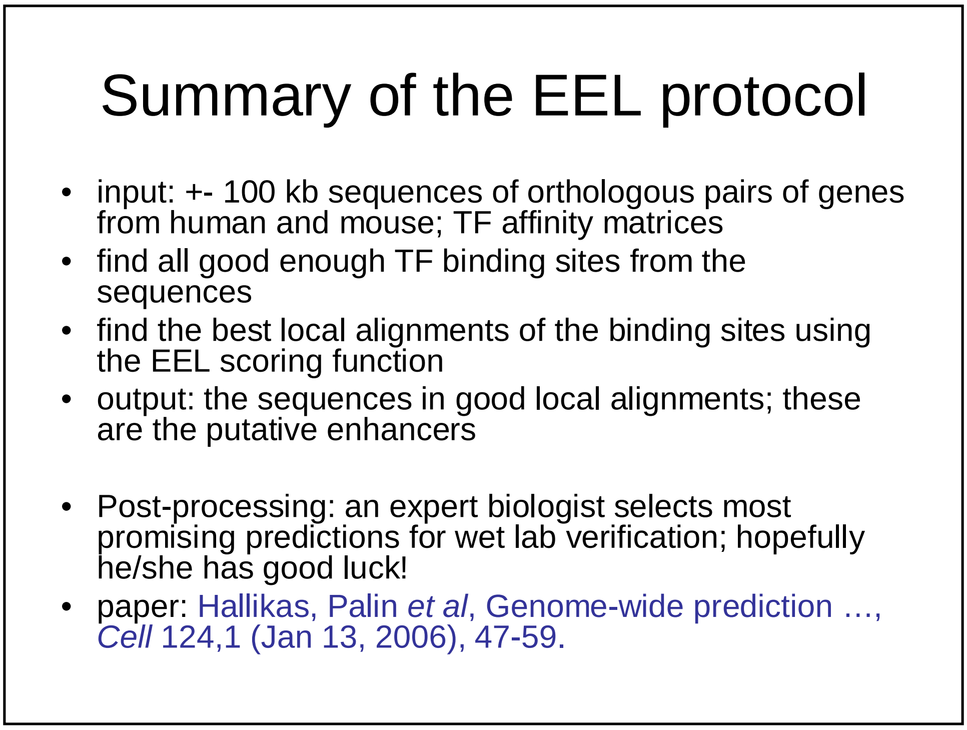
Wet-lab verification

- Selected predicted cis-modules for wet-lab verification
- Fused 1kb DNA segment containing the predicted enhancer to a marker gene (LacZ) with a minimal promoter and generated transgenic embryos.



Enhancer prediction for *N-myc*


200 kb Human *N-Myc* genomic region
200 kb Mouse *N-Myc* genomic region
coding region of *N-Myc*
Conserved GLI binding sites in two predicted enhancer elements, CM5 and CM7
CM5
CM5
CM7
Tissue-specific findings



Summary of the EEL protocol

- input: +- 100 kb sequences of orthologous pairs of genes from human and mouse; TF affinity matrices
- find all good enough TF binding sites from the sequences
- find the best local alignments of the binding sites using the EEL scoring function
- output: the sequences in good local alignments; these are the putative enhancers

- Post-processing: an expert biologist selects most promising predictions for wet lab verification; hopefully he/she has good luck!
- paper: Hallikas, Palin *et al*, Genome-wide prediction …, *Cell* 124,1 (Jan 13, 2006), 47-59.