More Motifs

WMM, log odds scores, Neyman-Pearson, background;
Greedy & EM for motif discovery

Neyman-Pearson

- Given a sample $x_1, x_2, ..., x_n$, from a distribution $f(...|\Theta)$ with parameter Θ, want to test hypothesis $\Theta = \theta_1$ vs $\Theta = \theta_2$.
- Might as well look at *likelihood ratio:*

$$\frac{f(x_1, x_2, ..., x_n|\theta_1)}{f(x_1, x_2, ..., x_n|\theta_2)} > \tau$$

What's best WMM?

- Given 20 sequences $s_1, s_2, ..., s_k$ of length 8, assumed to be generated at random according to a WMM defined by 8 x (4-1) parameters θ, what's the best θ?
- E.g., what MLE for θ given data $s_1, s_2, ..., s_k$?
- Answer: count frequencies per position.

Weight Matrix Models

8 Sequences:

ATG
ATG
ATG
ATG
ATG
GTG
GTG
TTG

Freq.	Col 1	Col 2	Col3
A	.625	0	0
C	0	0	0
G	.250	0	1
T	.125	1	0

Log-Likelihood Ratio:

$$\log_2 \frac{f_{x_i,i}}{f_{x_i}}, \; f_{x_i} = \frac{1}{4}$$

LLR	Col 1	Col 2	Col 3
A	1.32	-∞	-∞
C	-∞	-∞	-∞
G	0	-∞	2.00
T	-1.00	2.00	-∞

Non-uniform Background

- *E. coli* - DNA approximately 25% A, C, G, T
- *M. jannaschi* - 68% A-T, 32% G-C

LLR from previous example, assuming

$$f_A = f_T = 3/8$$
$$f_C = f_G = 1/8$$

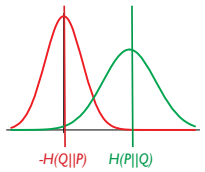
LLR	Col 1	Col 2	Col 3
A	.74	-∞	-∞
C	-∞	-∞	-∞
G	1.00	-∞	3.00
T	-1.58	1.42	-∞

e.g., G in col 3 is 8 x more likely via WMM than background, so ($\log_2$) score = 3 (bits).

WMM: How "Informative"? Mean score of site vs bkg?

- For any fixed length sequence *x*, let
  P(x) = Prob. of *x* according to WMM
  Q(x) = Prob. of *x* according to background
- Recall Relative Entropy:

$$H(P||Q) = \sum_{x \in \Omega} P(x) \log_2 \frac{P(x)}{Q(x)}$$



- *H(P||Q)* is *expected log likelihood score* of a sequence randomly chosen from *WMM*; *-H(Q||P)* is expected score of *Background*

For WMM, you can show (based on the assumption of independence between columns), that :

$$H(P||Q) = \sum_i H(P_i||Q_i)$$

where P_i and Q_i are the WMM/background distributions for column i.

WMM Example, cont.

Freq.	Col 1	Col 2	Col3
A	.625	0	0
C	0	0	0
G	.250	0	1
T	.125	1	0

Uniform

LLR	Col 1	Col 2	Col 3	
A	1.32	-∞	-∞	
C	-∞	-∞	-∞	
G	0	-∞	2.00	
T	-1.00	2.00	-∞	
RelEnt	.70	2.00	2.00	4.70

Non-uniform

LLR	Col 1	Col 2	Col 3	
A	.74	-∞	-∞	
C	-∞	-∞	-∞	
G	1.00	-∞	3.00	
T	-1.58	1.42	-∞	
RelEnt	.51	1.42	3.00	4.93

Pseudocounts

- Are the $-\infty$'s a problem?
  - Certain that a given residue *never* occurs in a given position? Then $-\infty$ just right
  - Else, it may be a small-sample artifact
- Typical fix: add a *pseudocount* to each observed count—small constant (e.g., .5, 1)
- Sounds *ad hoc*; there is a Bayesian justification

How-to Questions

- Given aligned motif instances, build model?
  - Frequency counts (above, maybe with pseudocounts)
- Given a model, find (probable) instances?
  - Scanning, as above
- Given unaligned strings thought to contain a motif, find it? (e.g., upstream regions for co-expressed genes from a microarray experiment)
  - Hard... next few lectures.

Motif Discovery: 3 example approaches

- Greedy search
- Expectation Maximization
- Gibbs sampler

Note: finding a site of max relative entropy in a set of unaligned sequences is NP-hard (Akutsu)

Greedy Best-First Approach
[Hertz & Stormo]

Input:
- Sequence $s_1, s_2, ..., s_k$; motif length l; "breadth" d

Algorithm:
- create singleton set with each length l subsequence of each $s_1, s_2, ..., s_k$
- for each set, add each possible length l subsequence not already present
- compute relative entropy of each
- discard all but d best
- repeat until all have k sequences

usual "greedy" problems

Expectation Maximization
[MEME, Bailey & Elkan, 1995]

Input (as above):

- Sequence s_1, s_2, ..., s_k; motif length l; background model; again assume one instance per sequence (variants possible)

Algorithm: EM

- Visible data: the sequences
- Hidden data: where's the motif

$$Y_{i,j} = \begin{cases} 1 & \text{if motif in sequence } i \text{ begins at position } j \\ 0 & \text{otherwise} \end{cases}$$

- Parameters θ: The WMM

MEME Outline

Typical EM algorithm:

- Given parameters θ^t at t^{th} iteration, use them to estimate where the motif instances are (the hidden variables)
- Use those estimates to re-estimate the parameters θ to maximize likelihood of observed data, giving θ^{t+1}
- Repeat

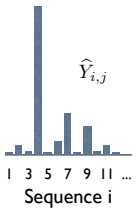
Expectation Step
(where are the motif instances?)

$$\begin{aligned} \widehat{Y}_{i,j} &= E(Y_{i,j} \mid s_i, \theta^t) \\ &= P(Y_{i,j} = 1 \mid s_i, \theta^t) \\ &= P(s_i \mid Y_{i,j} = 1, \theta^t) \frac{P(Y_{i,j}=1|\theta^t)}{P(s_i|\theta^t)} \\ &= cP(s_i \mid Y_{i,j} = 1, \theta^t) \\ &= c' \prod_{k=1}^{l} P(s_{i,j+k-1} \mid \theta^t) \end{aligned}$$

$E = 0 \cdot P(0) + 1 \cdot P(1)$

Bayes

where c' is chosen so that $\sum_j \widehat{Y}_{i,j} = 1$.



Maximization Step
(what is the motif?)

Find θ maximizing expected value:

$$\begin{aligned} Q(\theta \mid \theta^t) &= E_{Y \sim \theta^t}[\log P(s, Y \mid \theta)] \\ &= E_{Y \sim \theta^t}[\log \textstyle\prod_{i=1}^{k} P(s_i, Y_i \mid \theta)] \\ &= E_{Y \sim \theta^t}[\textstyle\sum_{i=1}^{k} \log P(s_i, Y_i \mid \theta)] \\ &= E_{Y \sim \theta^t}[\textstyle\sum_{i=1}^{k} \sum_{j=1}^{|s_i|-l+1} Y_{i,j} \log P(s_i, Y_{i,j} = 1 \mid \theta)] \\ &= E_{Y \sim \theta^t}[\textstyle\sum_{i=1}^{k} \sum_{j=1}^{|s_i|-l+1} Y_{i,j} \log (P(s_i \mid Y_{i,j} = 1, \theta) P(Y_{i,j} = 1 \mid \theta))] \\ &= \textstyle\sum_{i=1}^{k} \sum_{j=1}^{|s_i|-l+1} E_{Y \sim \theta^t}[Y_{i,j}] \log P(s_i \mid Y_{i,j} = 1, \theta) + C \\ &= \textstyle\sum_{i=1}^{k} \sum_{j=1}^{|s_i|-l+1} \widehat{Y}_{i,j} \log P(s_i \mid Y_{i,j} = 1, \theta) + C \end{aligned}$$

M-Step (cont.)

$$Q(\theta \mid \theta^t) \quad = \quad \sum_{i=1}^{k} \sum_{j=1}^{|s_i|-l+1} \widehat{Y}_{i,j} \log P(s_i \mid Y_{i,j} = 1, \theta) + C$$

Exercise: Show this is maximized by "counting" letter frequencies over all possible motif instances, with counts weighted by $\widehat{Y}_{i,j}$, again the "obvious" thing.

s_1 : ACGGATT...

...

s_k : GC...TCGGAC

$\widehat{Y}_{1,1}$	ACGG
$\widehat{Y}_{1,2}$	CGGA
$\widehat{Y}_{1,3}$	GGAT
⋮	⋮
$\widehat{Y}_{k,l-1}$	CGGA
$\widehat{Y}_{k,l}$	GGAC

Initialization

1. Try *every* motif-length substring, and use as initial θ a WMM with, say 80% of weight on that sequence, rest uniform
2. Run a few iterations of each
3. Run best few to convergence

(Having a supercomputer helps)