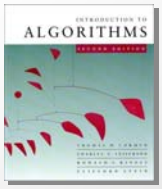
Introduction to Algorithms

6.046J/18.401J



LECTURE 18

Computational Biology

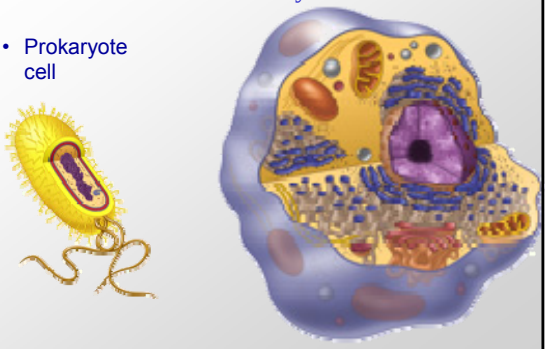
- Bio intro: Regulatory Motifs
- Combinatorial motif discovery
  - Median string finding
- Probabilistic motif discovery
  - Expectation maximization
- Comparative genomics

Prof. Manolis Kellis

April 15, 2008

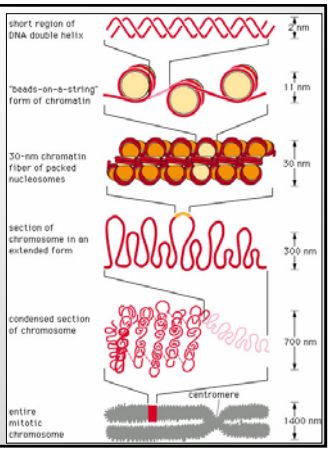
Chromosomes inside the cell

- Eukaryote cell
- Prokaryote cell



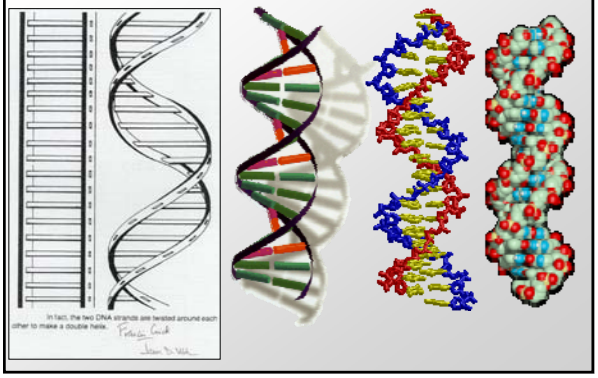
DNA packaging

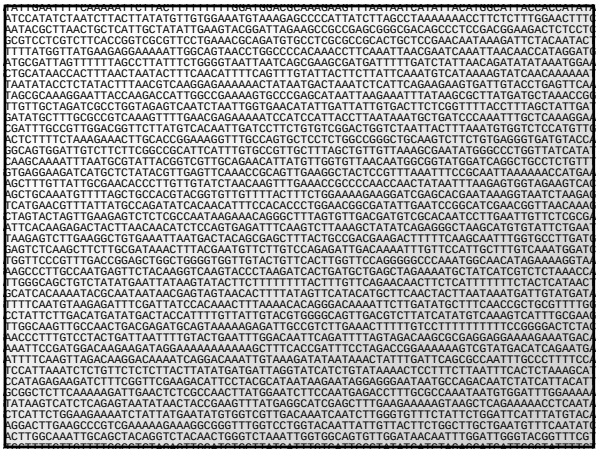
- Why packaging
  - DNA is very long
  - Cell is very small
- Compression
  - Chromosome is 50,000 times shorter than extended DNA
- Using the DNA
  - Before a piece of DNA is used for anything, this compact structure must open locally

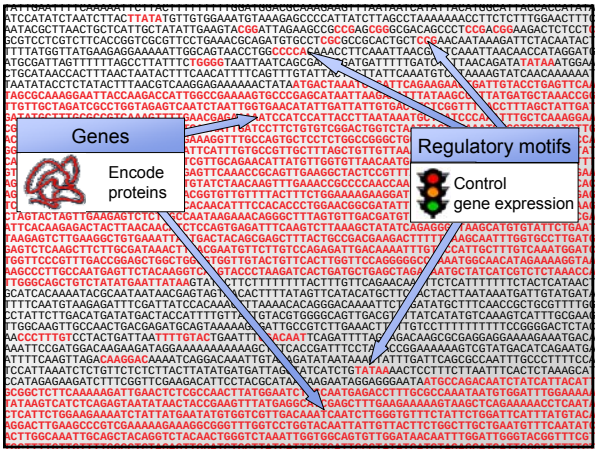


DNA: The double helix

- The most noble molecule of our time

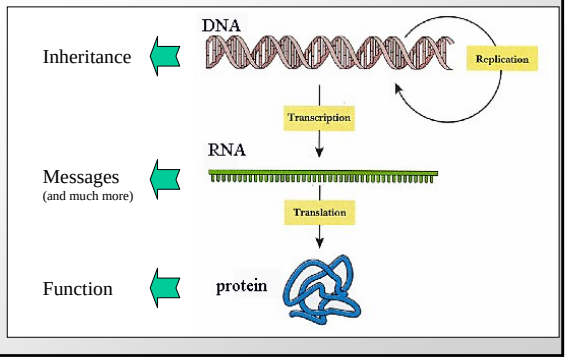




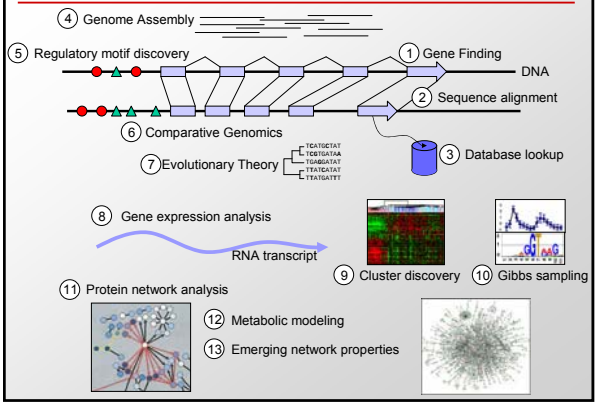


The Central Dogma of Biology

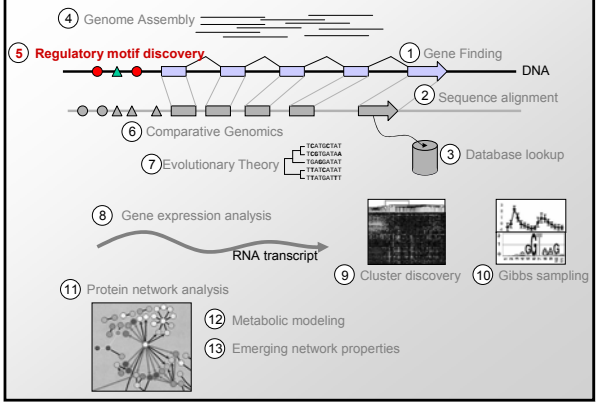
DNA makes RNA makes Protein



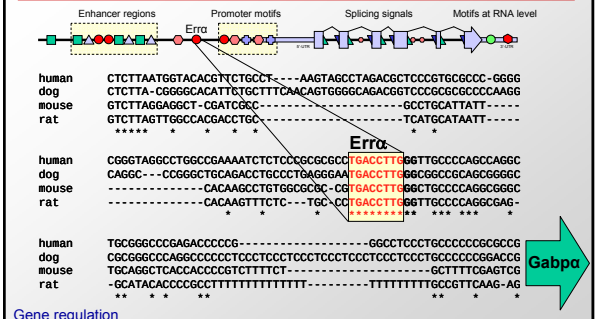
Challenges in Computational Biology



Challenges in Computational Biology



The regulatory code



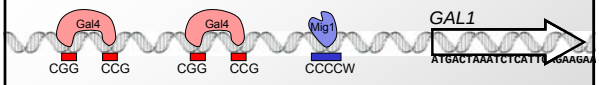
Gene regulation

Cells respond to environment and change during development.
These events are minutely controlled by short patterns within the DNA.

Regulatory motifs:

Sequence patterns that control gene usage, recognized by specific regulators.
General, short (~6-12 letters), possibly degenerate, act at varying distances.

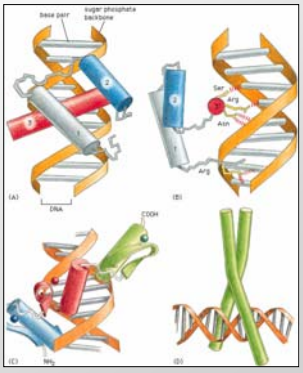
Regulatory motif discovery



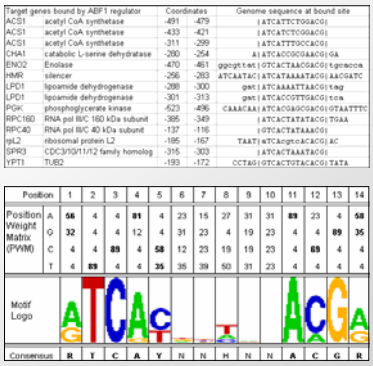
- Regulatory motifs (summary)
  - Genes are turned on / off in response to changing environments
  - No direct addressing: subroutines (genes) contain sequence tags (motifs)
  - Specialized proteins (transcription factors) recognize these tags
- What makes motif discovery hard?
  - Motifs are short (6-8 bp), sometimes degenerate
  - Can contain any set of nucleotides (no ATG or other rules)
  - Act at variable distances upstream (or downstream) of target gene
- How can we discover them?

Three-dimensional contacts of regulators and DNA

- Protein-DNA interactions
  - "**Feeling**" chemical properties of the bases
  - **DO NOT open** DNA (not by base complementarity)
- Sequence specificity
  - **Topology** of 3D contact dictates sequence specificity of binding
  - Some positions are **fully constrained**; other positions are **degenerate**
  - "Ambiguous / degenerate" positions are **loosely** contacted by the transcription factor



Motifs capture regulator sequence specificity



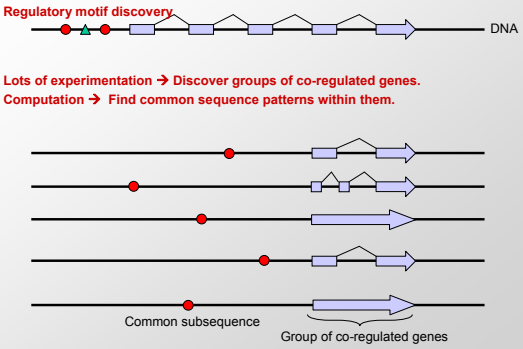
- Summarize information
- Building blocks of gene regulation
- Underlying code linking regulatory networks together
- How do we go about discovering them?
- Motif vs. motif instance!

Three settings for motif discovery

Three settings for motif discovery

- Combinatorial solutions
  - Exhaustive search
  - Greedy motif clustering
  - Wordlets and motif refinement
- Probabilistic solutions
  - Expectation maximization
  - Gibbs sampling
- Comparative genomics
  - Genome-wide conservation
  - Evolutionary signatures

Computational motif discovery (traditional)



Problem Definition

Given a collection of promoter sequences $s_1,\ldots, s_N$ of genes with common expression

Combinatorial

Motif M: $m_1 \ldots m_W$
Some of the m_i's blank

- Find M that occurs in all s_i with $\leq$ k differences
- Or, Find M with smallest total hamming dist

Probabilistic

Motif: M_{ij}; $1 \leq i \leq W$
$1 \leq j \leq 4$
M_{ij} = Prob[letter j, pos i]

Find best M, and positions $p_1,\ldots,$ p_N in sequences

Three settings for motif discovery

- Combinatorial solutions
  - Exhaustive search
  - Greedy motif clustering
  - Wordlets and motif refinement
- Probabilistic solutions
  - Expectation maximization
  - Gibbs sampling
- Comparative genomics
  - Genome-wide conservation
  - Evolutionary signatures

Discrete Formulations

Given sequences S = {x^1, …, x^n}

- A motif W is a consensus string $w_1 \ldots w_K$
- Find motif W* with "best" match to x^1, …, x^n

Definition of "best":

d(W, x^i) = min hamming dist. between W and any word in x^i

d(W, S) = $\sum_i$ d(W, x^i)

Exhaustive Searches

1. Pattern-driven algorithm:

For W = AA…A to TT…T (4^K possibilities)
Find d(W, S)
Report W* = argmin(d(W, S))

Running time: O(K N 4^K)
(where N = Σ_i |x^i|)

Advantage: Finds provably "best" motif W
Disadvantage: Time

Exhaustive Searches

2. Sample-driven algorithm:

For W = every K-long word occurring in some x^i
Find d(W, S)

Report W* = argmin(d(W, S))
or, Report a local improvement of W*

Running time: O(K N^2)

Advantage: Time

Disadvantage: If the true motif is weak and does not occur in data

then a random motif may score better than any instance of true motif

Overview

- Introduction
  - Bio review: Where do ambiguities come from?
  - Computational formulation of the problem
- Combinatorial solutions
  - Exhaustive search
  - **Greedy motif clustering**
  - Wordlets and motif refinement
- Probabilistic solutions
  - Expectation maximization
  - Gibbs sampling

Greedy motif clustering (CONSENSUS)

Algorithm:

Cycle 1:
For each word W in S (of fixed length!)
For each word W' in S
Create alignment (gap free) of W, W'

Keep the C_1 best alignments, A_1, …, A_{C1}

```
ACGGTTG  ,  CGAACTT  ,  GGGCTCT …
ACGCCTG  ,  AGAACTA  ,  GGGGTGT …
```

Greedy motif clustering (CONSENSUS)

Algorithm:

Cycle t:
For each word W in S
For each alignment A_j from cycle t-1
Create alignment (gap free) of W, A_j

Keep the C_t best alignments A_1, …, A_{Ct}

```
ACGGTTG  ,  CGAACTT  ,  GGGCTCT …
ACGCCTG  ,  AGAACTA  ,  GGGGTGT …
…        ,  …        ,  …
ACGGCTC  ,  AGATCTT  ,  GGCGTCT …
```

Greedy motif clustering (CONSENSUS)

- $C_1, \ldots, C_n$ are user-defined heuristic constants
  - N is sum of sequence lengths
  - n is the number of sequences

Running time:

$O(N^2) + O(N C_1) + O(N C_2) + \ldots + O(N C_n)$

$= O(N^2 + NC_{total})$

Where $C_{total} = \Sigma_i C_i$, typically O(nC), where C is a big constant

Overview

- Introduction
  - Bio review: Where do ambiguities come from?
  - Computational formulation of the problem
- Combinatorial solutions
  - Exhaustive search
  - Greedy motif clustering
  - **Wordlets and motif refinement**
- Probabilistic solutions
  - Expectation maximization
  - Gibbs sampling

Motif Refinement and wordlets (MULTIPROFILER)

- Extended sample-driven approach

Given a K-long word W, define:

$N_\alpha(W)$ = words W' in S s.t. $d(W,W') \leq \alpha$

Idea:
Assume W is occurrence of true motif W^*
Will use $N_\alpha(W)$ to correct "errors" in W

Motif Refinement and wordlets (MULTIPROFILER)

Assume W differs from true motif W^* in at most L positions

Define:

A wordlet G of W is a L-long pattern with blanks, differing from W
- L is smaller than the word length K

Example:

K = 7; L = 3

```
W  =  ACGTTGA
G  =  --A--CG
```

Motif Refinement and wordlets (MULTIPROFILER)

Algorithm:

For each W in S:
  For L = 1 to L_{max}
  1. Find the α-neighbors of W in S → $N_\alpha(W)$
  2. Find all "strong" L-long wordlets G in $N_\alpha(W)$
  3. For each wordlet G,
     1. Modify W by the wordlet G → W'
     2. Compute d(W', S)

Report W^* = argmin d(W', S)

Step 1 above: Smaller motif-finding problem;
Use exhaustive search

Three settings for motif discovery

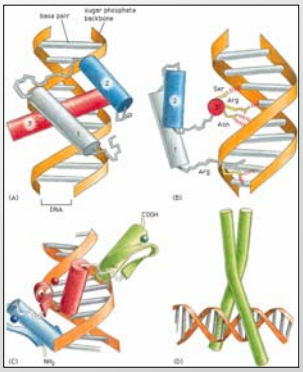
- Combinatorial solutions
  - Exhaustive search
  - Greedy motif clustering
  - Wordlets and motif refinement
- Probabilistic solutions
  - Expectation maximization
  - Gibbs sampling
- Comparative genomics
  - Genome-wide conservation
  - Evolutionary signatures

Overview

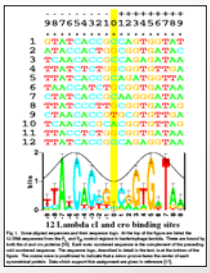
- Introduction
  - Bio review: Where do ambiguities come from?
  - Computational formulation of the problem
- Combinatorial solutions
  - Exhaustive search
  - Greedy motif clustering
  - Wordlets and motif refinement
- Probabilistic solutions
  - **Expectation maximization**
  - Gibbs sampling

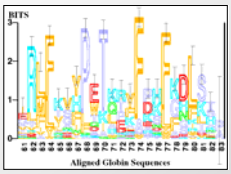
Where do *ambiguous bases* come from ?

- Protein-DNA interactions
  - Proteins read DNA by "feeling" the chemical properties of the bases
  - Without opening DNA (not by base complementarity)
- Sequence specificity
  - Topology of 3D contact dictates sequence specificity of binding
  - Some positions are fully constrained; other positions are degenerate
  - "Ambiguous / degenerate" positions are loosely contacted by the transcription factor



Representing motif ambiguities



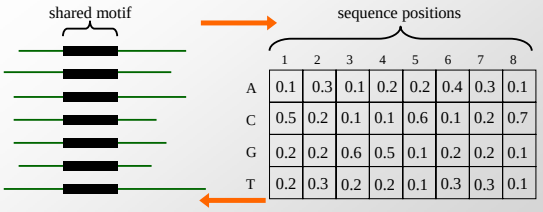


entropy - *n*
1: (communication theory) a numerical measure of the uncertainty of an outcome; "the signal contained thousands of bits of information" [information, selective information]
2: (thermodynamics) a thermodynamic quantity representing the amount of energy in a system that is no longer available for doing mechanical work; "entropy increases as matter and energy in the universe degrade to an ultimate state of inert uniformity" [randomness]

- Entropy at pos'n I, H(i) $= -\Sigma_{\{\text{letter x}\}} \text{freq}(x, i) \log_2 \text{freq}(x, i)$
- Height of x at pos'n i, L(x, i) = freq(x, i) (2 – H(i))
  - Examples:
    - freq(A, i) = 1; H(i) = 0; L(A, i) = 2
    - A: ½; C: ¼; G: ¼; H(i) = 1.5; L(A, i) = ¼; L(not T, i) = ¼

Starting positions ⇔ Motif matrix

- given aligned sequences ➔ easy to compute profile matrix



	1	2	3	4	5	6	7	8
A	0.1	0.3	0.1	0.2	0.2	0.4	0.3	0.1
C	0.5	0.2	0.1	0.1	0.6	0.1	0.2	0.7
G	0.2	0.2	0.6	0.5	0.1	0.2	0.2	0.1
T	0.2	0.3	0.2	0.2	0.1	0.3	0.3	0.1

given profile matrix

- easy to find starting position probabilities

Key idea: Iterative procedure for estimating both, given uncertainty (learning problem with hidden variables: the starting positions)

Basic Iterative Approach

Given: length parameter *W*, training set of sequences
set initial values for **motif**
do
➔ re-estimate ***starting-positions*** from ***motif***
➔ re-estimate ***motif*** from ***starting-positions***
until convergence (change < ε)
return: ***motif, starting-positions***

Representing Motif (p_{ck}) and Background (p_{c0})

- Assume motif has fixed width, *W*
- Motif represented by matrix of probabilities: p_{ck}
  the probability of character ***c*** in column ***k***

$$p = \begin{array}{c|ccc} & 1 & 2 & 3 \\ A & 0.1 & 0.5 & 0.2 \\ C & 0.4 & 0.2 & 0.1 \\ G & 0.3 & 0.1 & 0.6 \\ T & 0.2 & 0.2 & 0.1 \end{array}$$ (~CAG)

- Background represented by p_{c0}, frequency of each base

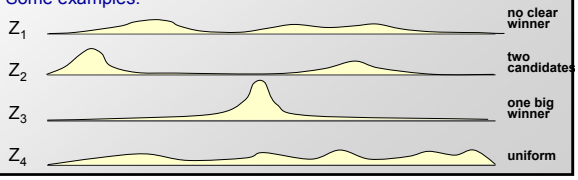
$$p_0 = \begin{array}{c|c} & 0 \\ A & 0.26 \\ C & 0.24 \\ G & 0.23 \\ T & 0.27 \end{array}$$ (near uniform)
(see also: di-nucleotide etc)

Representing the starting position probabilities (Z_{ij})

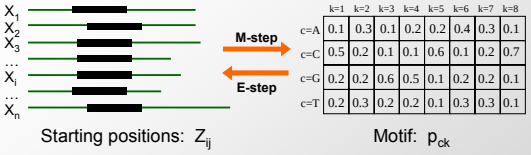
- the element Z_{ij} of the matrix Z represents the probability that the motif starts in position *j* in sequence *I*

$$Z = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \text{seq1} & 0.1 & 0.1 & 0.2 & 0.6 \\ \text{seq2} & 0.4 & 0.2 & 0.1 & 0.3 \\ \text{seq3} & 0.3 & 0.1 & 0.5 & 0.1 \\ \text{seq4} & 0.1 & 0.5 & 0.1 & 0.3 \end{array}$$

Some examples:



Starting positions (Z_{ij}) ⇔ Motif matrix (p_{ck})

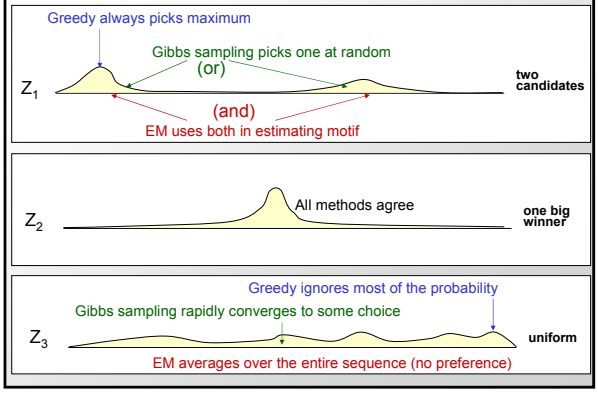


Starting positions: Z_{ij} Motif: p_{ck}

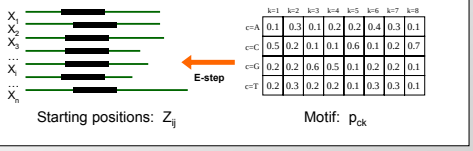
- Z_{ij}: Probability that on sequence i, motif start at position j
- p_{ck}: Probability that k^{th} character of motif is letter c

- Computing Z_{ij} matrix from p_{ck} is straightforward
  - At each position, evaluate start probability by multiplying across the matrix
- Three variations for re-computing motif p_{ck} from Z_{ij} matrix
  - Expectation maximization → All starts weighted by Z_{ij} prob distribution
  - Gibbs sampling → Single start for each seq X_i by sampling Z_{ij}
  - Greedy approach → Best start for each seq X_i by maximum Z_{ij}

Three examples of Greedy, Gibbs Sampling, EM



E-step: Calculating Z_{ij} from motif



Starting positions: Z_{ij} Motif: p_{ck}

Calculating $P(X_i)$ when motif position is known

- Probability of training sequence X_i, given hypothesized start position j

$$\Pr(X_i \mid Z_{ij} = 1, p) = \underbrace{\prod_{k=1}^{j-1} p_{c_k,0}}_{\text{before motif}} \underbrace{\prod_{k=j}^{j+W-1} p_{c_k,k-j+1}}_{\text{motif}} \underbrace{\prod_{k=j+W}^{L} p_{c_k,0}}_{\text{after motif}}$$

- Example:

X_i = G C **T G T** A G

$$p = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \text{A} & 0.25 & 0.1 & 0.5 & 0.2 \\ \text{C} & 0.25 & 0.4 & 0.2 & 0.1 \\ \text{G} & 0.25 & 0.3 & 0.1 & 0.6 \\ \text{T} & 0.25 & 0.2 & 0.2 & 0.1 \end{array}$$

$$\Pr(X_i \mid Z_{i3} = 1, p) =$$

$$p_{G,0} \times p_{C,0} \times p_{T,1} \times p_{G,2} \times p_{T,3} \times p_{A,0} \times p_{G,0} =$$

$$0.25 \times 0.25 \times \boxed{0.2 \times 0.1 \times 0.1} \times 0.25 \times 0.25$$

Calculating the Z vector (using P(Xi))

- To estimate the starting positions in *Z* at step *t*

$$\Pr(Z_{ij}=1 \mid X_i, p) = \frac{\Pr(X_i \mid Z_{ij}=1, p)\, \Pr(Z_{ij}=1)}{\Pr(X_i)} \quad \text{(Bayes' rule)}$$

- At iteration t, calculate $Z_{ij}^{(t)}$ based on $p^{(t)}$
  - We just saw how to calculate $\Pr(X_i \mid Z_{ij}=1, p)$
  - To obtain total probability $\Pr(X_i)$, sum over all starting positions

$$Z_{ij}^{(t)} = \frac{\Pr(X_i \mid Z_{ij} = 1, p^{(t)})\, \cancel{\Pr(Z_{ij} = 1)}}{\sum_{k=1}^{L-W+1} \Pr(X_i \mid Z_{ik} = 1, p^{(t)})\, \cancel{\Pr(Z_{ik} = 1)}}$$

- Assume uniform priors (motif equally likely to start at any position)

Calculating the Z vector: Example

$X_i =$ G C T G T A G

	0	1	2	3
A	0.25	0.1	0.5	0.2
C	0.25	0.4	0.2	0.1
G	0.25	0.3	0.1	0.6
T	0.25	0.2	0.2	0.1

$p =$

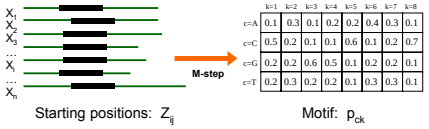
$$Z_{i1} = 0.3 \times 0.2 \times 0.1 \times 0.25 \times 0.25 \times 0.25 \times 0.25$$

$$Z_{i2} = 0.25 \times 0.4 \times 0.2 \times 0.6 \times 0.25 \times 0.25 \times 0.25$$

$\vdots$

- then normalize so that $\sum_{j=1}^{L-W+1} Z_{ij} = 1$

M-step: Calculating motif from Z_{ij}



Starting positions: Z_{ij} Motif: p_{ck}

The M-step: Estimating the motif *p*

- recall $p_{c,k}$ represents the probability of character *c* in position *k*; values for position 0 represent the background

$$p_{c,k}^{(t+1)} = \frac{n_{c,k} + d_{c,k}}{\sum_b (n_{b,k} + d_{b,k})}$$ pseudo-counts

$$n_{c,k} = \begin{cases} \sum_i \sum_{\{j | X_{i,j+k-1} = c\}} Z_{ij} & k > 0 \quad \text{motif} \\ n_c - \sum_{j=1}^{W} n_{c,j} & k = 0 \quad \text{background} \end{cases}$$

total # of c's in data set

M-step example: Estimating p_{ck} from Z_{ij}

$X_1 =$ A C A G C A
$Z_1 =$ 0.1 0.7 0.1 0.1

$X_2 =$ A G G C A G
$Z_2 =$ 0.4 0.1 0.1 0.4

$X_3 =$ T C A G T C
$Z_3 =$ 0.2 0.6 0.1 0.1

$$p_{A,1} = \frac{Z_{1,1} + Z_{1,3} + Z_{2,1} + Z_{3,3} + 1}{Z_{1,1} + Z_{1,2} \ldots + Z_{3,3} + Z_{3,4} + 4}$$

Gibbs sampling: Pick one
Greedy: Pick max

- EM: sum over full probability
  - $n_{A,1}$= 0.1+0.1+0.4+0.1 = 0.7
  - $n_{C,1}$= 0.7+0.4+0.6 = 1.7
  - $n_{G,1}$= 0.1+0.1+0.1+0.1= 0.4
  - $n_{T,1}$= 0.2 = 0.2
  - Total: T=0.7+1.7+0.4+0.2 = 3.0
- Normalize and add pseudo-counts
  - $P_{A,1}$ = (0.7+1)/(T+4) = 1.7/7=0.24
  - $P_{C,1}$ = (1.7+1)/(T+4) = 2.7/7=0.39
  - $P_{G,1}$ = (0.4+1)/(T+4) = 1.4/7=0.2
  - $P_{T,1}$ = (0.2+1)/(T+4) = 1.2/7=0.17
- P_{ck} =

	1	2	3
A	0.24	0.39	0.21
C	0.39	0.21	0.18
G	0.2	0.24	0.44
T	0.17	0.16	0.16

The EM Algorithm

- EM converges to a local maximum in the likelihood of the data given the model:

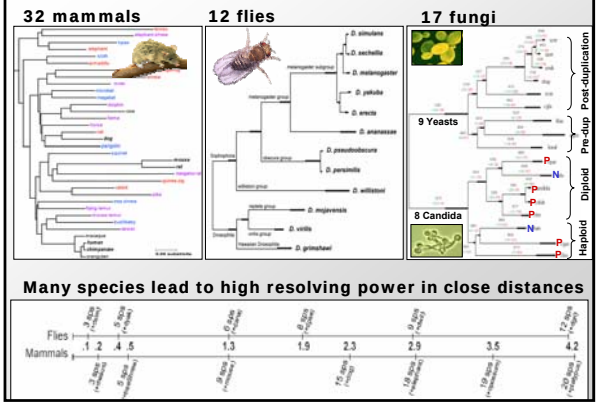
$$\prod_i \Pr(X_i \mid p)$$

- usually converges in a small number of iterations
- sensitive to initial starting point (i.e. values in *p*)

Three settings for motif discovery

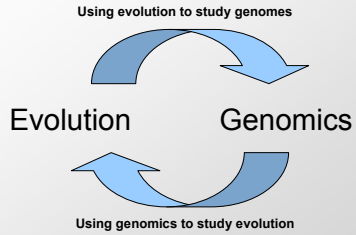
- Combinatorial solutions
  - Exhaustive search
  - Greedy motif clustering
  - Wordlets and motif refinement
- Probabilistic solutions
  - Expectation maximization
  - Gibbs sampling
- Comparative genomics
  - Genome-wide conservation
  - Evolutionary signatures

Resolving power in mammals, flies, fungi

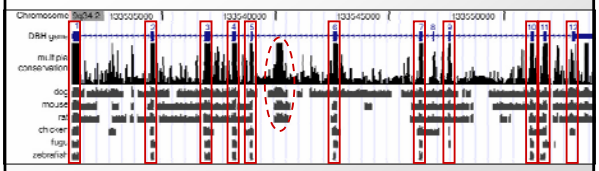


Many species lead to high resolving power in close distances

Comparative Genomics

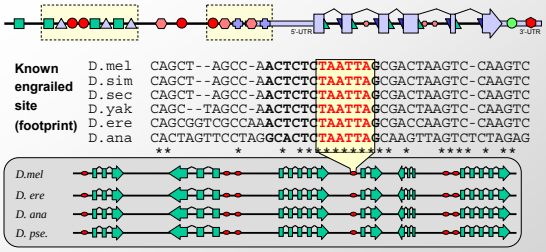


Comparative genomics and evolutionary signatures



- Comparative genomics can reveal functional elements
  - For example: exons are deeply conserved to mouse, chicken, fish
  - Many other elements are also strongly conserved: exons / regulatory?
- Can we also pinpoint specific functions of each region? Yes!
  - Patterns of change distinguish different types of functional elements
  - Specific function ⇔ Selective pressures ⇔ Patterns of mutation/inse/del
- Develop evolutionary signatures characteristic of each function

Evolutionary signal for regulatory motif discovery



- Individual motif instances are preferentially conserved
- Measure conservation across entire genome
  - Over thousands of motif instances → Increased discovery power
  - Couple to rapid enumeration and rapid string search

→ De novo discovery of regulatory motifs

Kellis *et al*, Nature 2003
Xie *et al*. Nature 2005
Stark *et al*, Nature 2007

Framing the problem computationally

- How do we find all instances of a motif in a genome?
  - Naïve algorithm: Search every position
- How do we count all instances of every 6-mer in a genome
  - Naïve algorithm: Scan the genome for each motif
  - Improvement: Scan genome once, filling a table
- How do we count all instances of every 50-mer in a genome
  - Table is no longer feasible, most entries empty
  - Use a hash table
- How do we search a new motif in a known genome
  - Pre-processing of the database
- How do we deal with motif degeneracy and ambiguities
  - Hash in multiple places, increase alphabet size, partial hashing

Computational approaches for motif discovery

- Method #1: Enumerate all motifs
  - Combinatorial search
- Method #2: Randomly sample the genome
  - Statistical approach
- Method #3: Enumerate motif seeds + refinement
  - Hill-climbing
- Method #4: Content-based addressing
  - Hashing

Evaluating genome-wide motif conservation

ATTAGCCAGTAGCGCAGTGCATGCATGCACGACTGCAAGTGCATGCATGCTAGCTACGTAGCTAGCCGCGCATGCTGTGACTGCTAG

	cons	total	ratio
AGTGAA	20	4000	
AGTGAC	20	4000	
AGTGAG	20	4000	
AGTGAT	20	4000	
AGTGCA	50	200	
AGTGCC	20	4000	
AGTGCG	20	4000	
AGTGCT	20	4000	

- Genome-wide conservation reveals real motifs
  - Count conserved instances → Not informative
  - Count total instances → Not informative
  - Evaluate conserved/total instances → Real motifs!
- Motif enumeration
  - Perfectly conserved motif instances
  - Each motif is a fully-specified 6-mer
- Algorithmic speed-up
  - Do not search entire genome for every motif
  - Scan genome once, fill in table of motif instances
  - Content-based indexing
- What's missing: Motif collapsing

Power of evolutionary signatures for motif discovery

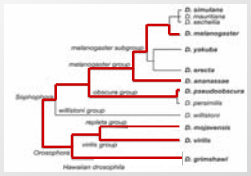
	Consensus	MCS	Matches to known	Expression enrichment	Promoters	Enhancers
1	CTAATTAAA	65.6	engrailed (en)		25.4	2
2	TTKCAATTAA	57.3	reversed-polarity (repo)		5.8	4.2
3	WATTRATTK	54.9	araucan (ara)		11.7	2.6
4	AAATTTATGCK	54.4	paired (prd)		4.5	16.5
5	GCAATAAA	51	ventral veins lacking (vvl)		13.2	0.3
6	DTAATTTRYNR	46.7	Ultrabithorax (Ubx)		16	3.3
7	TGATTAAT	45.7	apterous (ap)		7.1	1.7
8	YMATTAAAA	43.1	abdominal A (abd-A)		7	2.2
9	AAACNNGTT	41.2			20.1	4.3
10	RATTKAATT	40			3.9	0.7
11	GCACGTGT	39.5	fushi tarazu (ftz)		17.9	
12	AACASCTG	38.8	broad-Z3 (br-Z3)		10.7	
13	AATTRMATTA	38.2			19.5	1.2
14	TATGCWAAT	37.8			5.8	2
15	TAATTATG	37.5	Antennapedia (Antp)		14.1	5.4
16	CATNAATCA	36.9			1.8	1.7
17	TTACATAA	36.9			5.4	
18	RTAAATCAA	36.3			3.2	2.8
19	AATKNMATTT	36			3.6	0
20	ATGTCAAHT	35.6			2.4	4.6
21	ATAAAYAAA	35.5			57.2	-0.5
22	YYAATCAAA	33.9			5.3	0.6
23	WTTTTATG	33.8	Abdominal B (Abd-B)		6.3	6
24	TTTYMATTA	33.6	extradenticle (exd)		6.7	1.7
25	TGTMAATA	33.2			8.9	1.6
26	TAAYGAG	33.1			4.7	2.7
27	AAAKTGA	32.9			7.6	0.3
28	AAANNAAA	32.9			449.7	0.8
29	RTAAWTTAT	32.9	gooseberry-neuro (gsb-n)		11	0.8
30	TTATTTAYR	32.9	Deformed (Dfd)		30.7	

Ability to discover full dictionary of regulatory motifs *de novo*

Stark *et al*, Nature, 2007

5. Evolutionary signatures of motif instances

- Allow for motif movements
  - Sequencing/alignment errors
  - Loss, movement, divergence
- Measure branch-length score
  - Sum evidence along branches
  - Close species little contribution



BLS: 25% **Mef2:YTAWWWWTAR** BLS: 83%

Three settings for motif discovery

- Combinatorial solutions
  - Exhaustive search
  - Greedy motif clustering
  - Wordlets and motif refinement
- Probabilistic solutions
  - Expectation maximization
  - Gibbs sampling
- Comparative genomics
  - Genome-wide conservation
  - Evolutionary signatures

Challenges in Computational Biology