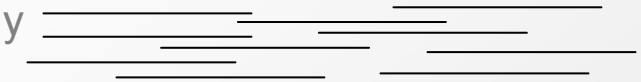
Motif finding in groups of related sequences

Challenges in Computational Biology

4 Genome Assembly



5 **Regulatory motif discovery**

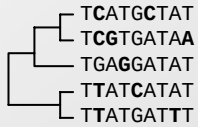
1 Gene Finding

DNA

2 Sequence alignment

6 Comparative Genomics

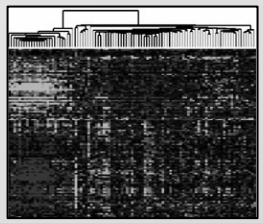
7 Evolutionary Theory



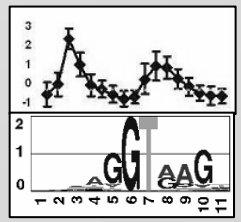
3 Database lookup

8 Gene expression analysis

RNA transcript

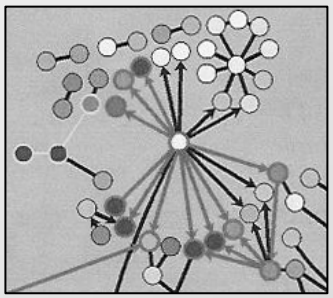


9 Cluster discovery



10 Gibbs sampling

11 Protein network analysis

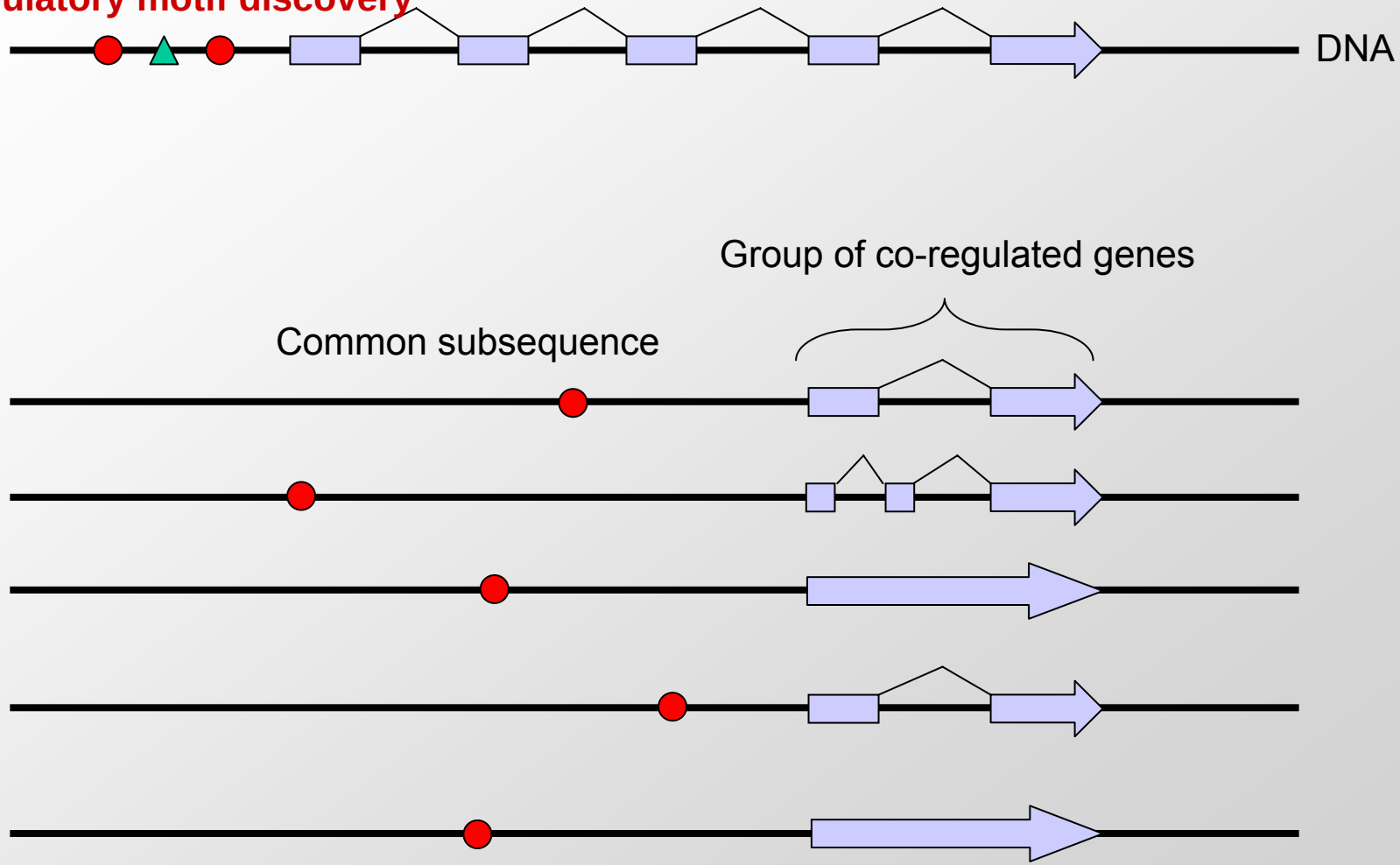


12 Regulatory network inference

13 Emerging network properties

Challenges in Computational Biology

5 **Regulatory motif discovery**



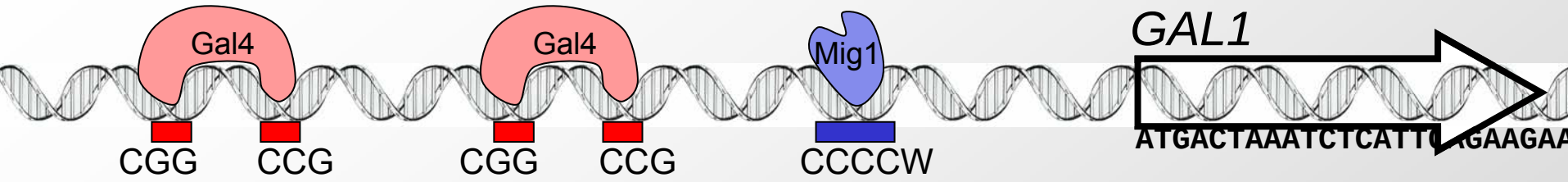
Overview

- 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

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

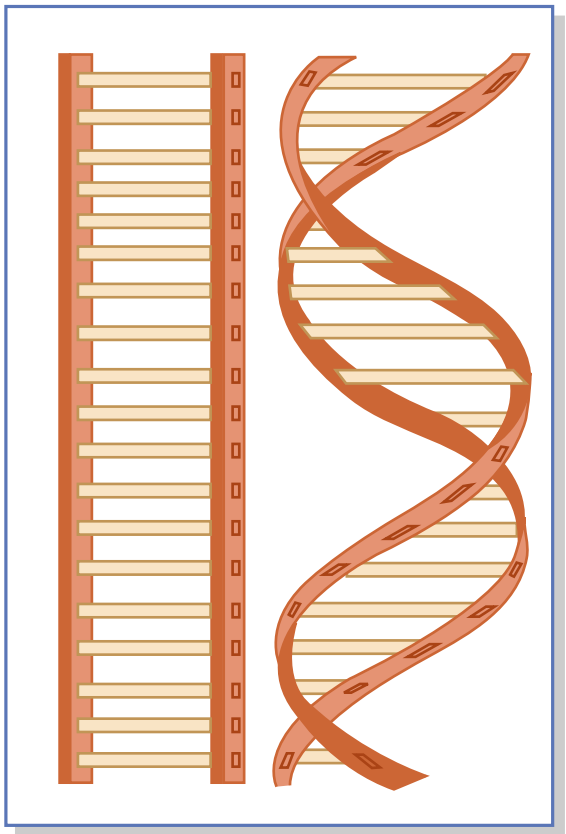
Regulatory motif discovery



- Regulatory motifs
  - 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

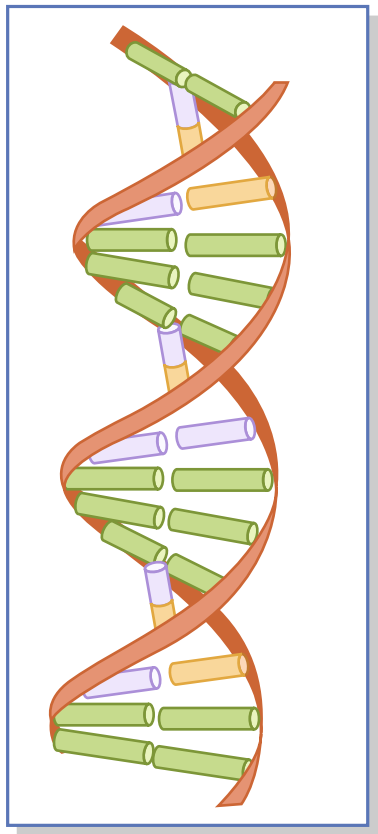
- What makes motif discovery hard?
  - Motifs are short (6-8 bp), sometimes degenerate
  - Can contain any set of nucleotides (no ATG or other rules)
  - Act at variable distances upstream (or downstream) of target gene

Sticks and backbones

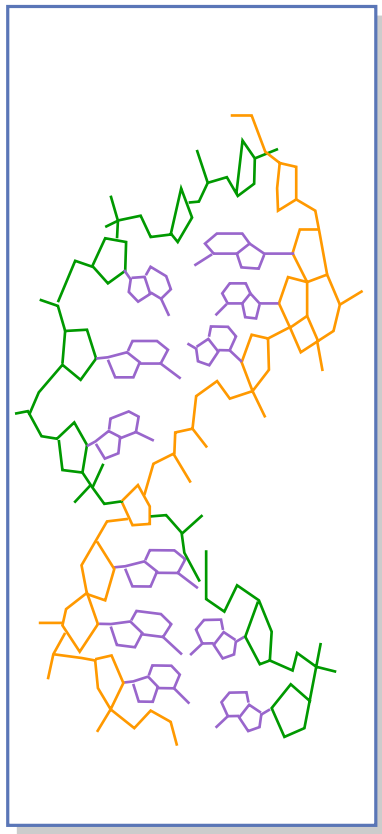


In fact, the two DNA strands are twisted around each other to make a double helix.

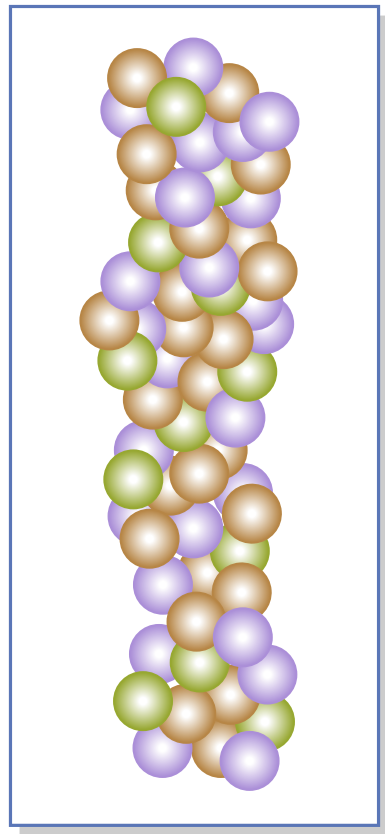
Traditional



Fancy



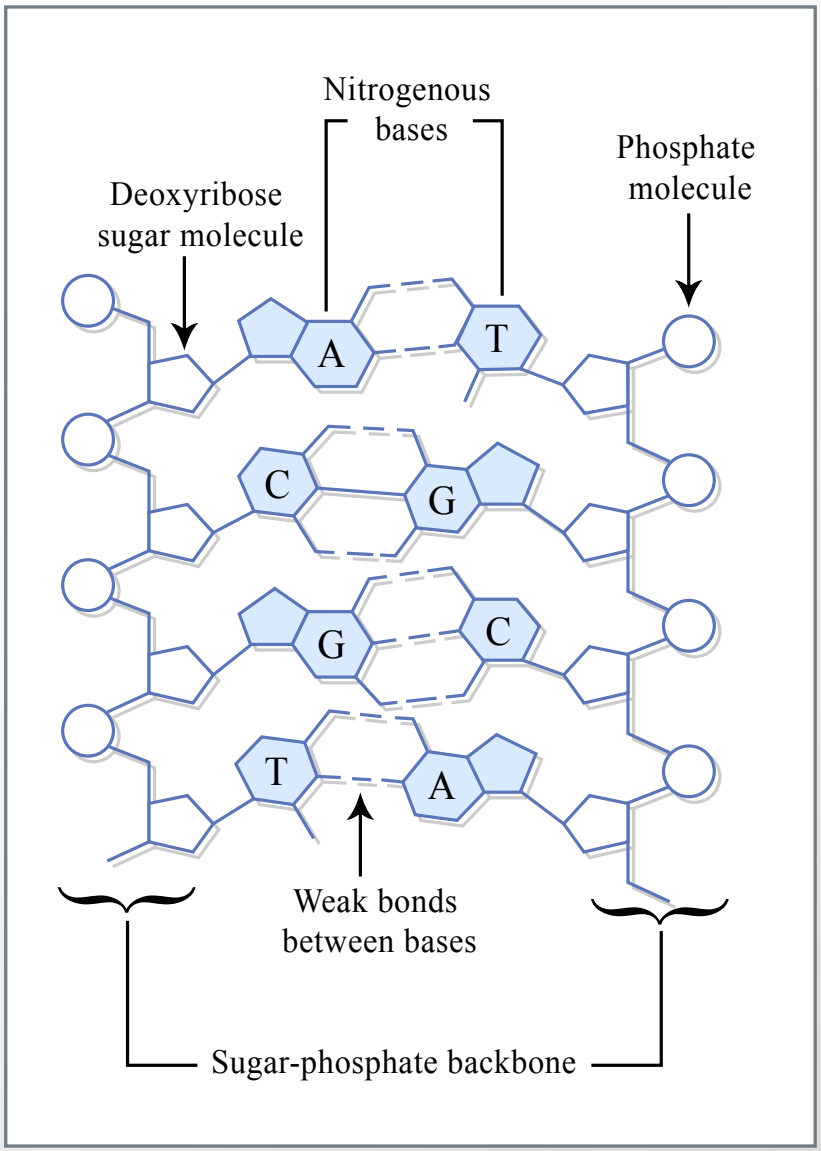
Chemical

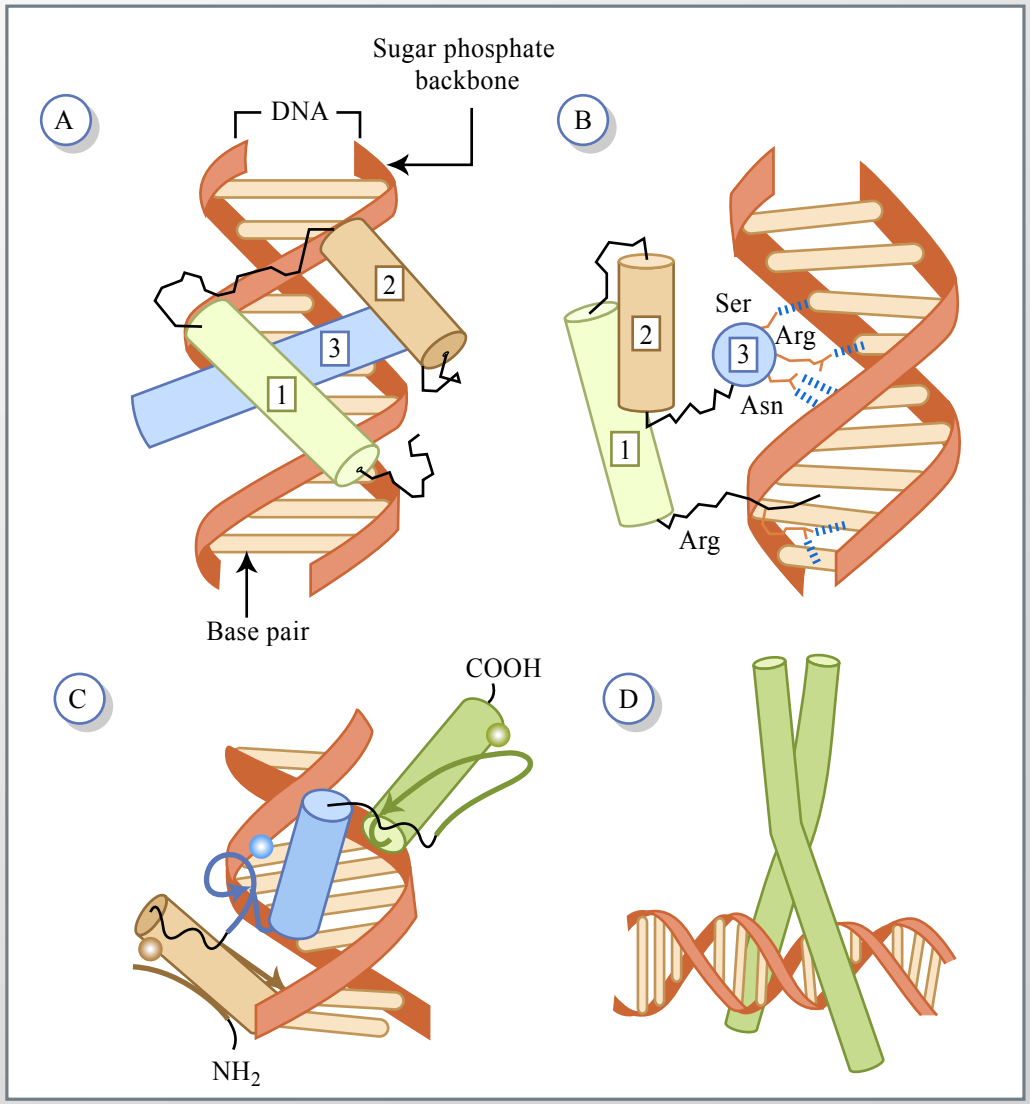


Atomic

Figure by MIT OCW.

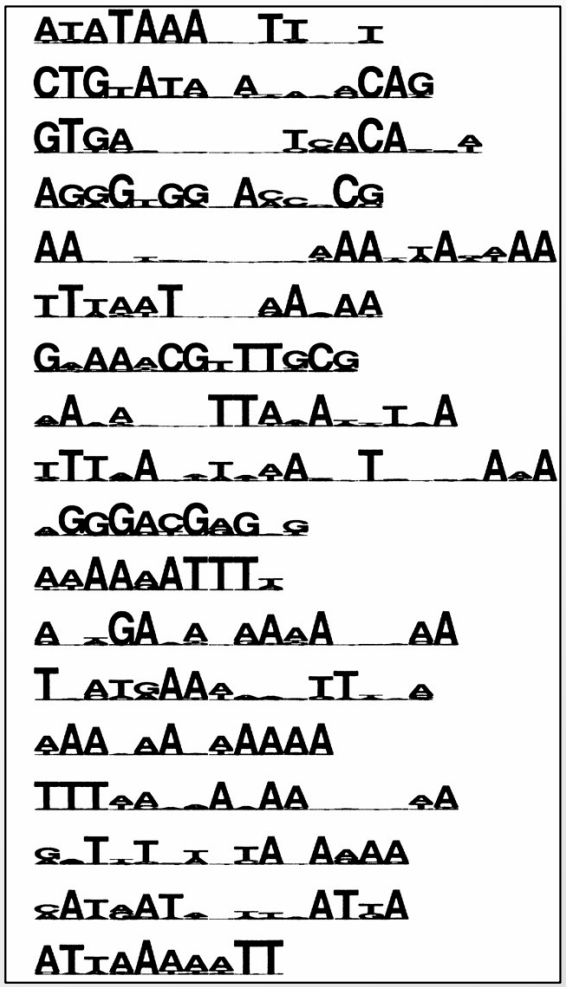
Where do *ambiguous bases* come from ?





Figures by MIT OCW.

Characteristics of Regulatory Motifs



- Tiny
- Highly Variable
- ~Constant Size
  - Because a constant-size transcription factor binds
- Often repeated
- Low-complexity-ish

Sequence Logos

Image removed due to copyright restrictions.

Image removed due to copyright restrictions.

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, $H(i) = -\sum_{\{\text{letter } x\}} \text{freq}(x, i) \log_2 \text{freq}(x, i)$
- Height of x at pos'n i, $L(x, i) = \text{freq}(x, i) (2 - H(i))$
  - 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

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

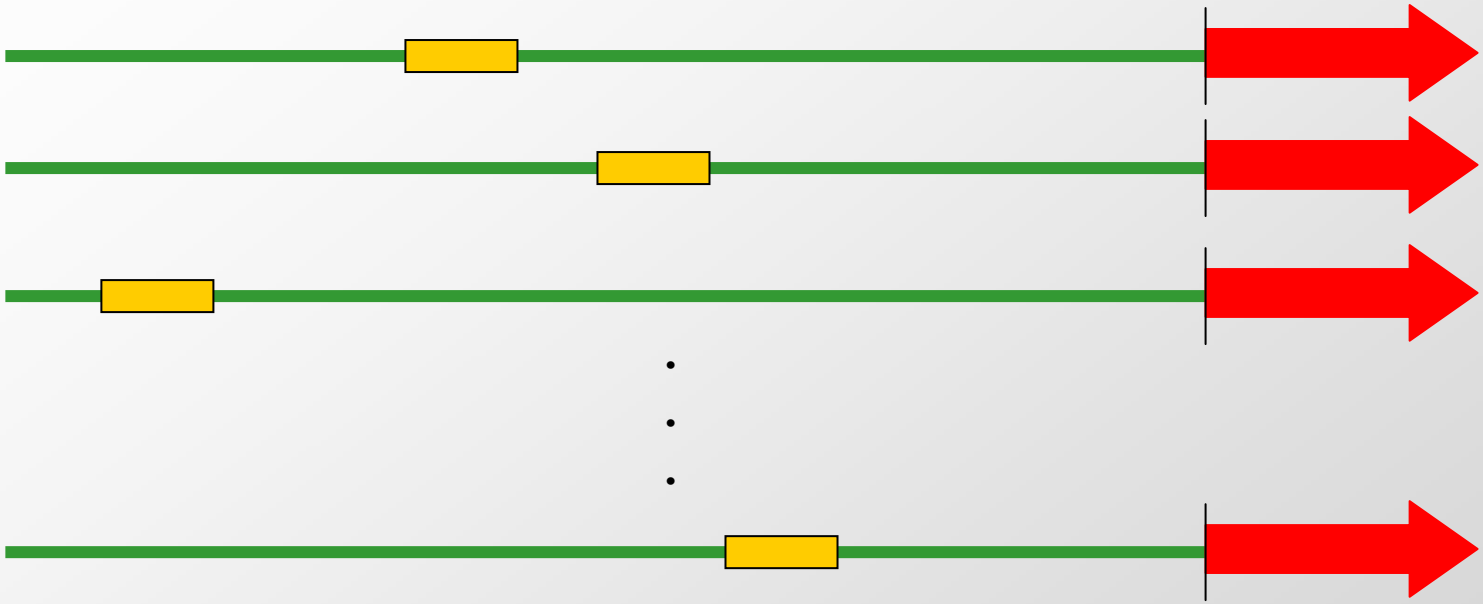
Motif: M_{ij}; $1 \leq i \leq W$

$1 \leq j \leq 4$

M_{ij} = Prob[letter j, pos i]

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

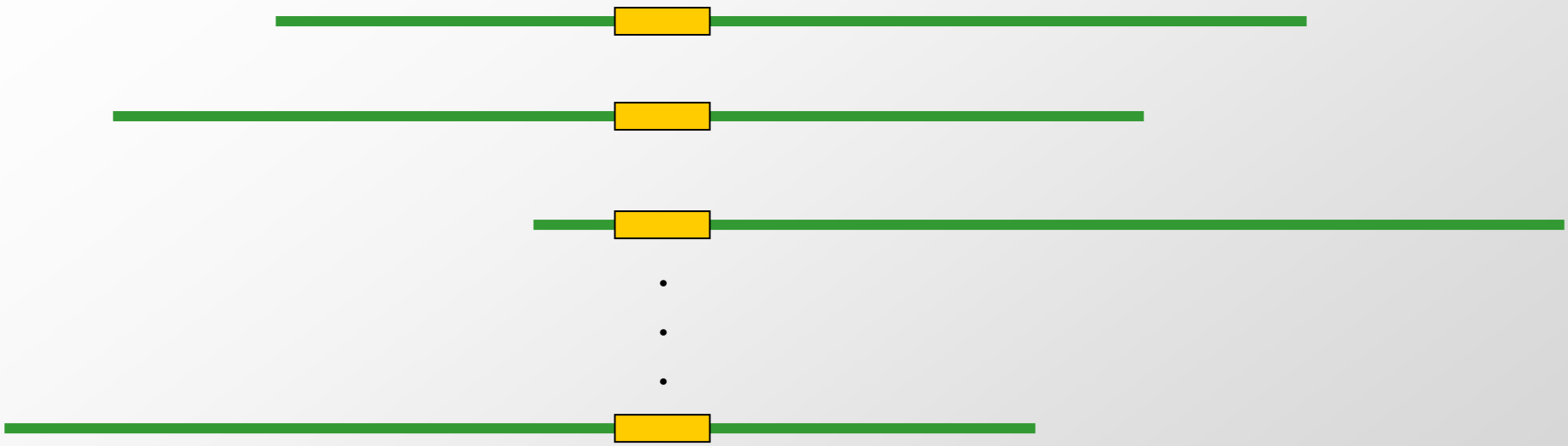
Finding Regulatory Motifs



Given a collection of genes bound by a transcription factor,

Find the TF-binding motif in common

Essentially a Multiple Local Alignment



- Find “best” multiple local alignment
- Alignment score defined differently in probabilistic/combinatorial cases

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

Discrete Formulations

Given sequences $S = \{x^1, \ldots, x^n\}$

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

Definition of "best":

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

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

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

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 = any 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_l best alignments A_1, …, A_{Ct}

ACGGTTG , CGAACTT , GGGCTCT …

ACGCCTG , AGAACTA , GGGGTGT …

… … …

ACGGCTC , AGATCTT , GGCGTCT …

Greedy motif clustering (CONSENSUS)

- C_1, …, 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:

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

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 $N_a(W)$
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