Motif finding : Lecture 2

CS 498 CXZ

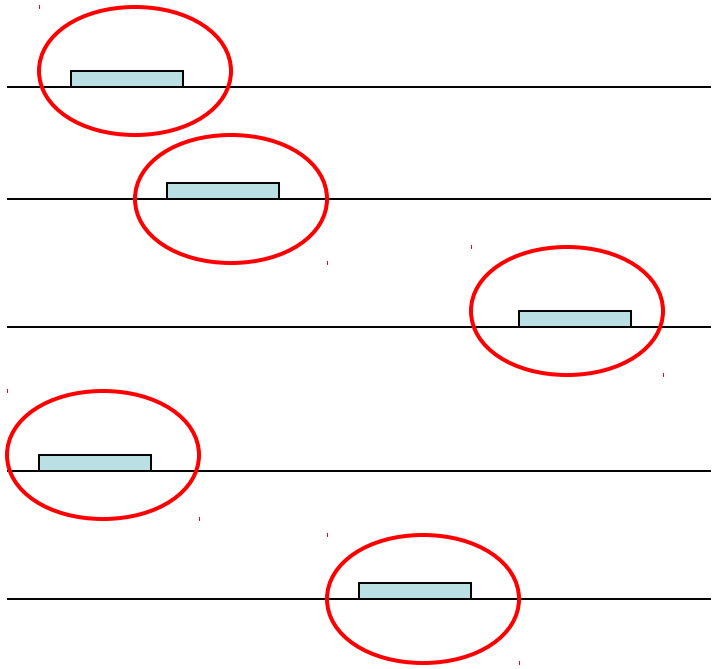
Recap

- Problem 1: Given a motif, finding its instances
- Problem 2: Finding motif ab initio.
  - Paradigm: look for over-represented motifs
  - Gibbs sampling

Ab initio motif finding: Gibbs sampling

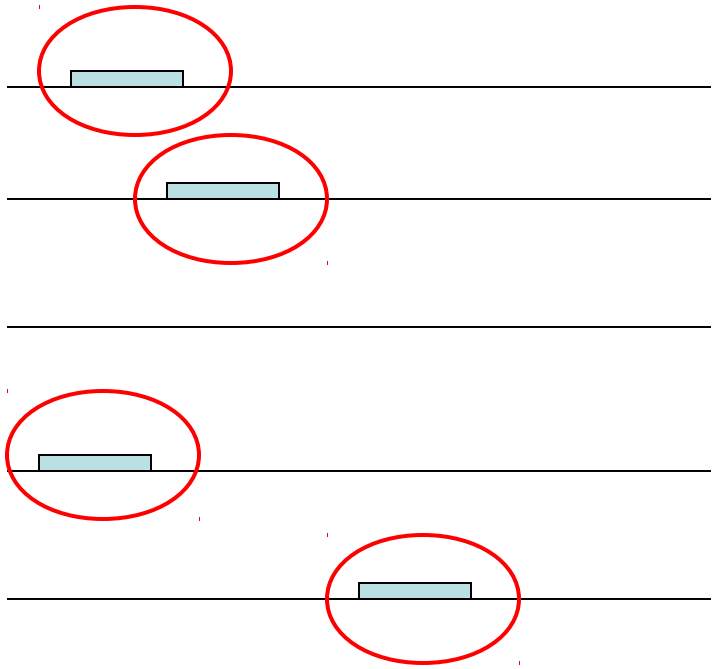
- Popular algorithm for motif discovery
- Motif model: Position Weight Matrix
- Local search algorithm

Gibbs sampling: basic idea



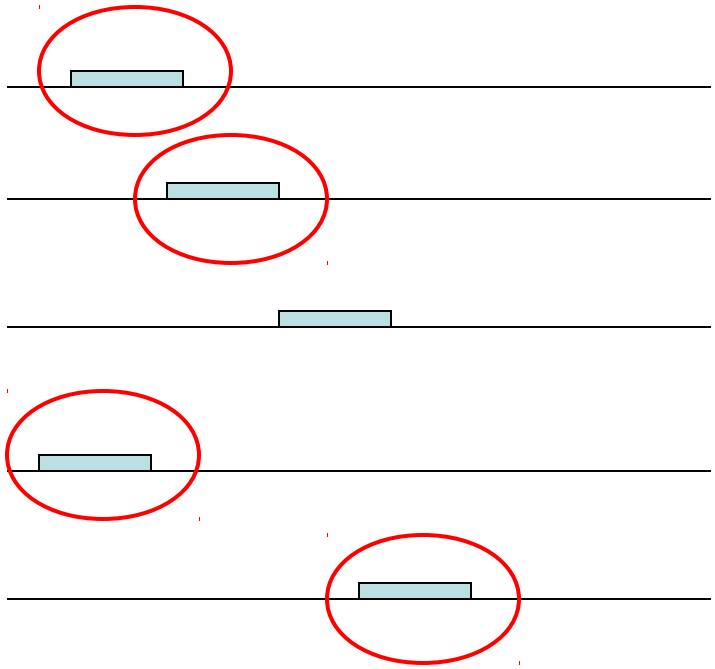
Current motif = PWM formed by circled substrings

Gibbs sampling: basic idea



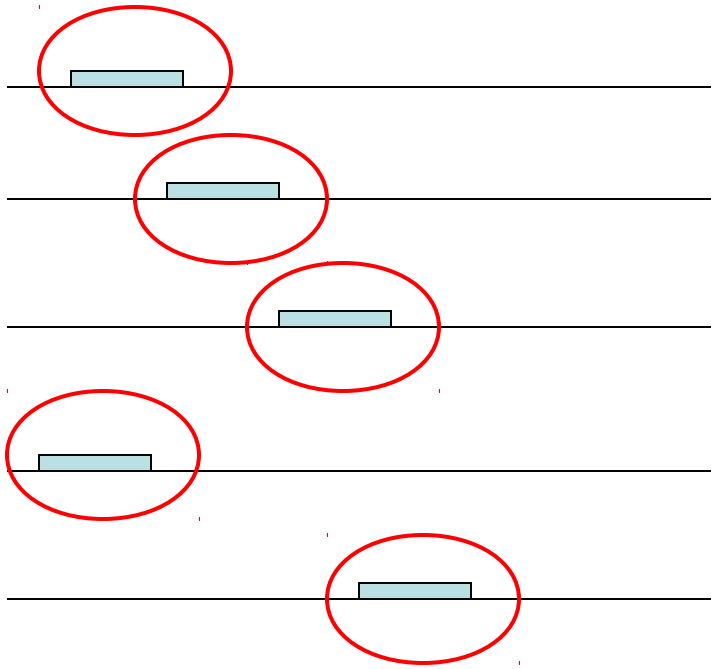
Delete one substring

Gibbs sampling: basic idea



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

Gibbs sampling: basic idea



New motif

Ab initio motif finding: Expectation Maximization

- Popular algorithm for motif discovery
- Motif model: Position Weight Matrix
- Local search algorithm
  - Move from current choice of motif to a new similar motif, so as to improve the score
  - Keep doing this until no more improvement is obtained : Convergence to local optima

How is a motif evaluated ?

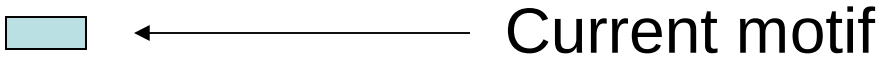
- Let W be a PWM. Let S be the input sequence.
- Imagine a process that randomly picks different strings matching W, and threads them together, interspersed with random sequence
- Let Pr(S|W) be the probability of such a process ending up generating S.

How is a motif evaluated ?

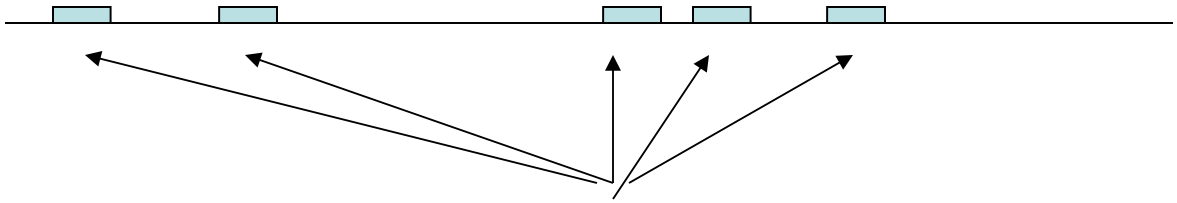
- Find W so as to maximize Pr(S|W)
- Difficult optimization
- Special technique called “Expectation-Maximization” or E-M.
- Iteratively finds a new motif W that improves Pr(S|W)

Basic idea of iteration

1. PWM



2. Scan sequence for good matches to the current motif.



3. Build a new PWM out of these matches, and make it the new motif

Guarantees

- The basic idea can be formalized in the language of probability
- Provides a formula for updating W, that guarantees an improvement in Pr(S|W)

MEME

- Popular motif finding program that uses Expectation-Maximization
- Web site

http://meme.sdsc.edu/meme/website/meme.html

Ab initio motif finding: CONSENSUS

- Popular algorithm for motif discovery, that uses a greedy approach
- Motif model: Position Weight Matrix
- Motif score: Information Content

Information Content

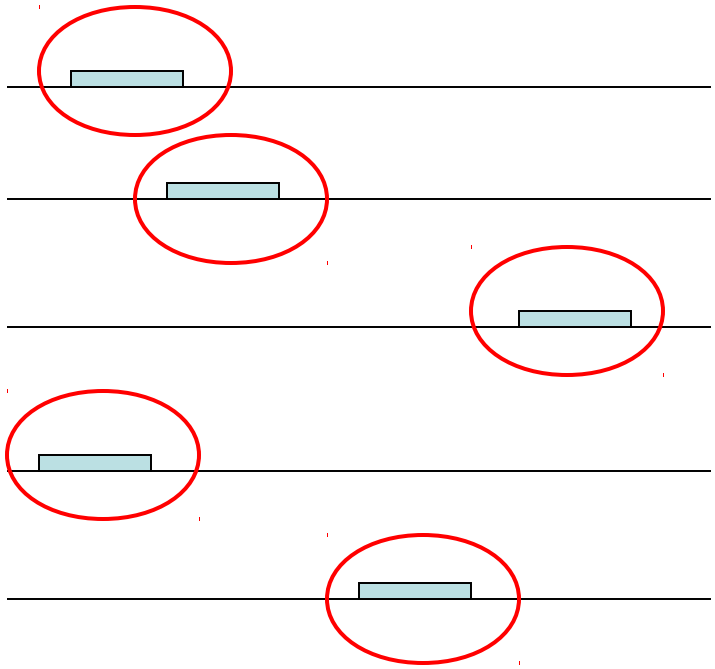
- PWM W:
- $W_{\beta k}$ = frequency of base β at position k
- q_β = frequency of base β by chance
- Information content of W:

$$\sum_{k} \sum_{b \in \{A,C,G,T\}} W_{bk} \log \frac{W_{bk}}{q_b}$$

Information Content

- If $W_{\beta k}$ is always equal to q_{β}, i.e., if W is similar to random sequence, information content of W is 0.
- If W is different from q, information content is high.

CONSENSUS: basic idea

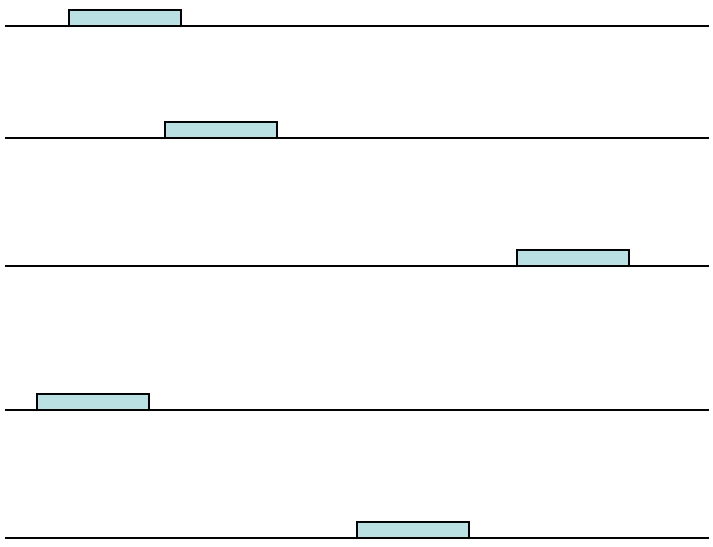


Final goal: Find a set of substrings, one in each input sequence

Set of substrings define a PWM. Goal: This PWM should have high information content.

High information content means that the motif “stands out”.

CONSENSUS: basic idea



Start with a substring in one input sequence

Build the set of substrings incrementally, adding one substring at a time

Until the entire set is built

CONSENSUS: the greedy heuristic

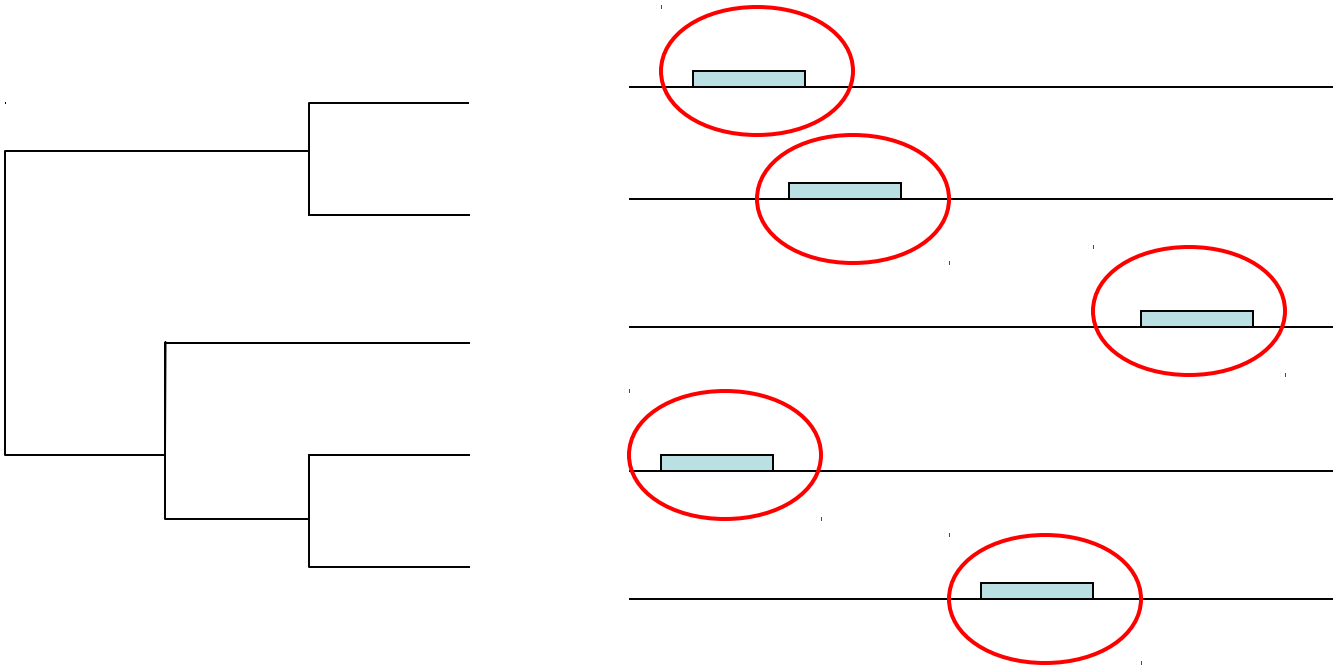
- Suppose we have built a partial set of substrings $\{s_1, s_2, \ldots, s_i\}$ so far.
- Have to choose a substring s_{i+1} from the input sequence S_{i+1}
- Consider each substring s of S_{i+1}
- Compute the score (information content) of the PWM made from $\{s_1, s_2, \ldots, s_i, s\}$
- Choose the s that gives the PWM with highest score, and assign $s_{i+1} \leftarrow s$

Phylogenetic footprinting

- So far, the input sequences were the “promoter” regions of genes believed to be “co-regulated”
- A special case: the input sequences are promoter regions of the same gene, but from multiple species.
  - Such sequences are said to be “orthologous” to each other.

Phylogenetic footprinting

Input sequences



Related by an evolutionary tree

Find motif

Phylogenetic footprinting: formally speaking

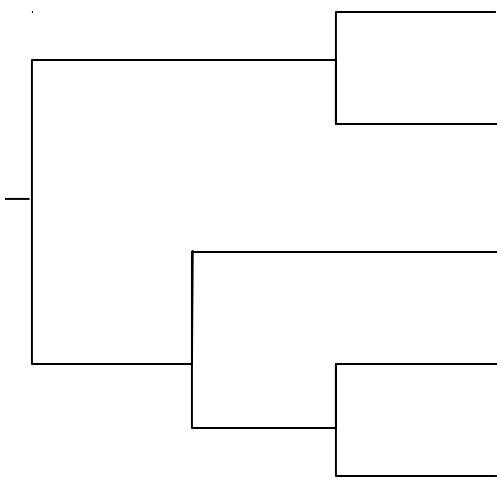
Given:

- phylogenetic tree T,
- set of orthologous sequences at leaves of T,
- length k of motif
- threshold d

Problem:

- Find each set S of k-mers, one k-mer from each leaf, such that the "parsimony" score of S in T is at most d.

Small Example



AGTCGTACGTGAC... (Human)

AGTAGACGTGCCG... (Chimp)

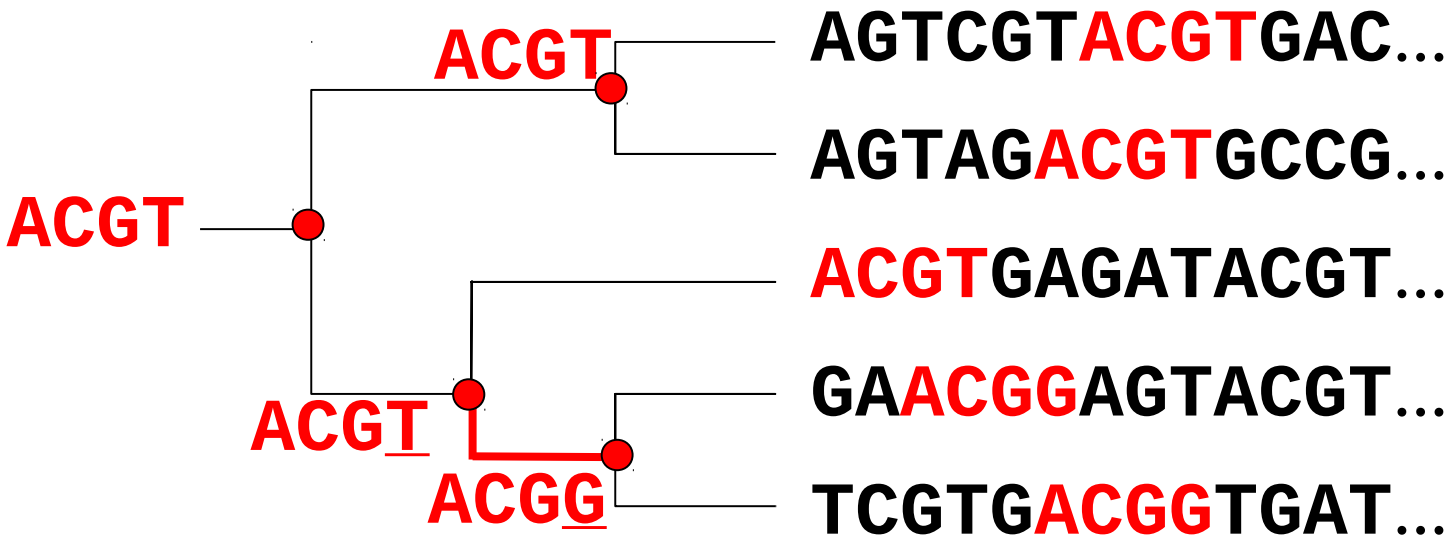
ACGTGAGATACGT... (Rabbit)

GAACGGAGTACGT... (Mouse)

TCGTGACGGTGAT... (Rat)

Size of motif sought: $k = 4$

Solution

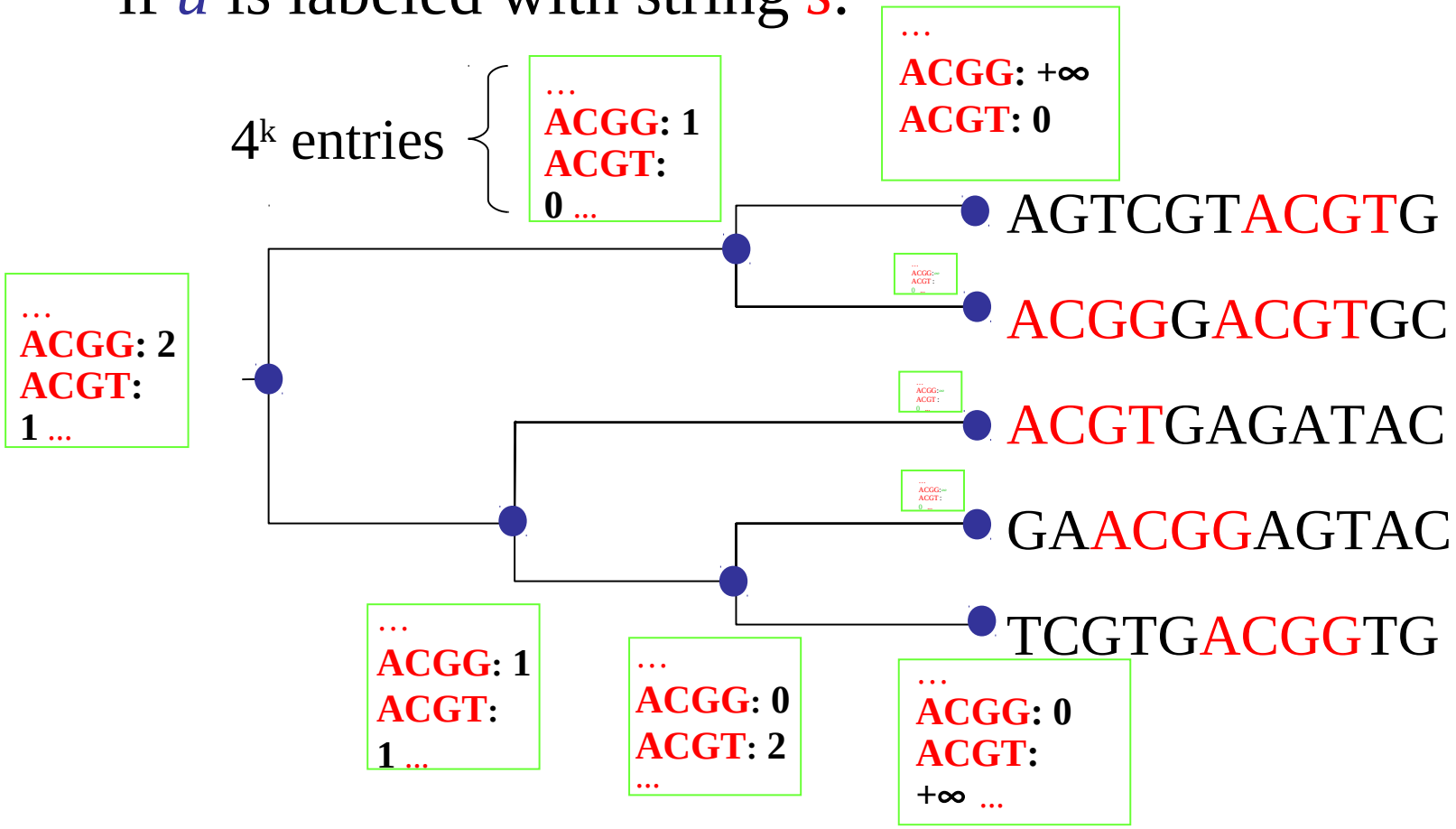


Parsimony score: 1 mutation

An Exact Algorithm

(Blanchette's algorithm)

$W_u[s]$ = best parsimony score for subtree rooted at node u, if u is labeled with string s.



Recurrence

$$W_u[s] = \sum_{v:\ \text{child}} \min_t \left(W_v[t] + d(s, t) \right)$$

of u

Running Time

$$W_u[s] = \sum_{v:\ \text{child}} \min_t \left(W_v[t] + d(s, t) \right)$$

of u

$O(k \cdot 4^{2k})$
time per node

What after motif finding ?

- Experiments to confirm results
- DNaseI footprinting & gel-shift assays
- Tells us which substrings are the binding sites

Before motif finding

- How do we obtain a set of sequences on which to run motif finding ?
- In other words, how do we get genes that we believe are regulated by the same transcription factor ?
- Two high-throughput experimental methods: ChIP-chip and microarray.

Before motif finding: ChIP-chip

- Take a particular transcription factor TF
- Take hundreds or thousands of promoter sequences
- Measure how strongly TF binds to each of the promoter sequences
- Collect the set to which TF binds strongly, do motif finding on these

Before motif finding: Microarray

- Take a large number of genes (mRNA) in the form of a soup
- Make a “chip” with thousands of slots, one for each gene
- Pour the gene (mRNA) soup on the chip
- Slots for genes that are highly expressed in the soup light up !

Before motif finding: Microarray

- Hence measure activity of thousands of genes in parallel
- Collect set of genes with similar expression (activity) profiles and do motif finding on these.