

# Gene Regulation and Microarrays

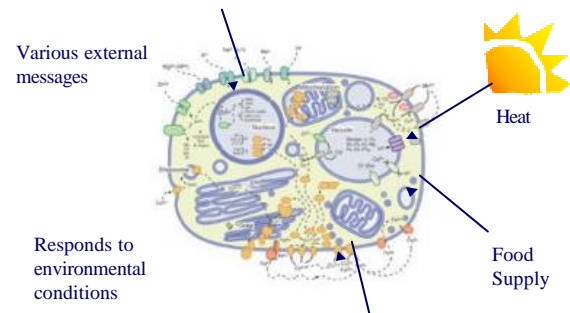
...after which we come back to multiple alignments for finding regulatory motifs

## Overview

- A. Gene Expression and Regulation
- B. Measuring Gene Expression: Microarrays
- C. Finding Regulatory Motifs

## A. Regulation of Gene Expression

## Cells respond to environment



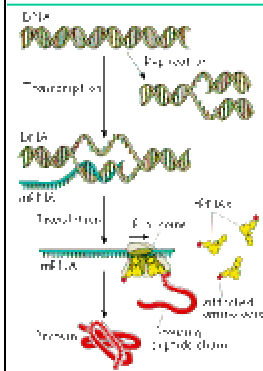
## Genome is fixed – Cells are dynamic

- A genome is static
  - Every cell in our body has a copy of same genome
- A cell is dynamic
  - Responds to external conditions
  - Most cells follow a **cell cycle** of division
- Cells differentiate during development

## Gene regulation

- ... is responsible for the dynamic cell
- Gene expression varies according to:
  - Cell type
  - Cell cycle
  - External conditions
  - Location

## Where gene regulation takes place

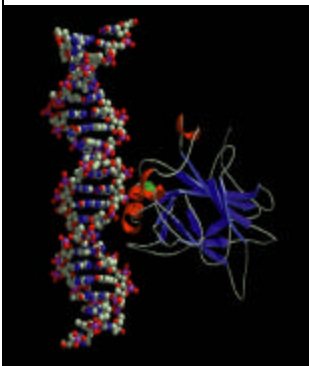


- Opening of chromatin
- Transcription
- Translation
- Protein stability
- Protein modifications

## Transcriptional Regulation

- **Strongest** regulation happens during transcription
- **Best** place to regulate:  
No energy wasted making intermediate products
- However, **slowest** response time  
After a receptor notices a change:
  1. Cascade message to nucleus
  2. Open chromatin & bind transcription factors
  3. Recruit RNA polymerase and transcribe
  4. Splice mRNA and send to cytoplasm
  5. Translate into protein

## Transcription Factors Binding to DNA



Transcription regulation:

Certain transcription factors bind DNA

Binding recognizes DNA substrings:

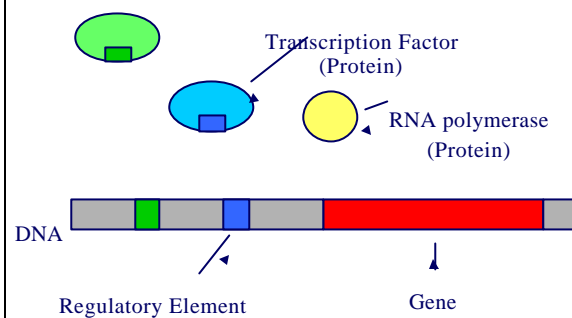
Regulatory motifs

## Promoter and Enhancers

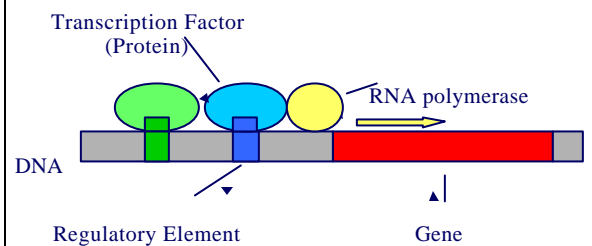


- Promoter necessary to start transcription
- Enhancers can affect transcription from afar

## Regulation of Genes

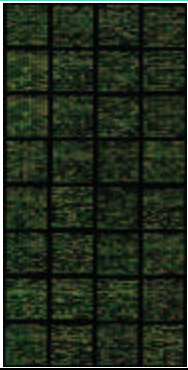


## Regulation of Genes





## What is a microarray (2)



- A 2D array of DNA sequences from thousands of genes
- Each spot has many copies of same gene
- Allow mRNAs from a sample to hybridize
- Measure number of hybridizations per spot

## How to make a microarray

- Method 1: Printed Slides (Stanford)
  - Use PCR to amplify a 1Kb portion of each gene
  - Apply each sample on glass slide
- Method 2: DNA Chips (Affymetrix)
  - Grow oligonucleotides (20bp) on glass
  - Several words per gene (choose unique words)

If we know the gene sequences,  
Can sample all genes in one experiment!

## Goal of Microarray Experiments

- Measure level of gene expression across many different conditions:

– Expression Matrix M: {genes}x{conditions}:

$$M_{ij} = \text{[gene]} \text{ in condition}$$

- Deduce gene function
- Deduce gene regulatory networks – **parts and connections**-level description of biology

## Steps Towards Achieving this Goal

1. Removing noise from gene expression levels
2. Feature Extraction
3. Clustering of genes/conditions
4. Analysis
  - a. Statistical significance of clusters
  - b. **Finding regulatory sequence motifs**
  - c. Building regulatory networks
  - d. Experimental verification

## 1. Removing Noise from Gene Expression Levels

- Expression levels vary with time, labs, concentrations, chemicals used
- Noise model:  $M_{ij} = c_i(a_{ij} g_i T_i + \epsilon_{ij})$ 
  - $M_{ij}$ ,  $T_{ij}$ : observed and true level gene<sub>i</sub>, chip<sub>j</sub>
  - $g_i$ ,  $c_j$ : mult. error constant for gene<sub>i</sub>, chip<sub>j</sub>
  - $a_{ij}$ ,  $\epsilon_{ij}$ : error terms
- Parameter Estimation
  - $c_j$ : spike in control probes
  - $g_i$ : control experiment of known concentration
  - $\epsilon_{ij}$ ,  $a_{ij}$ : minimize according to normal distribution

## 2. Feature Extraction

- Sample Correlation
  - Expression level can be different, but genes related; or similar, but genes unrelated

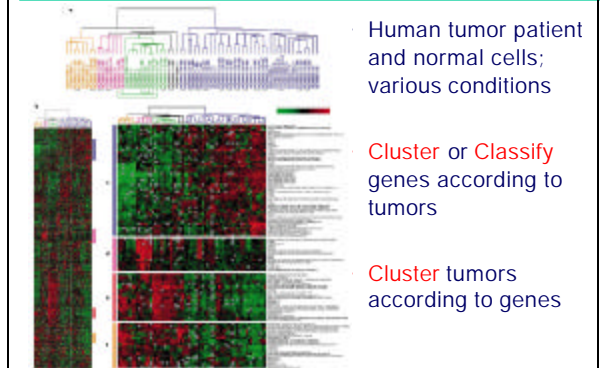
$$s(x, y) = \frac{\sum_{i=1}^{\#chips} (x_i - \hat{x})(y_i - \hat{y})}{\sqrt{\sum_{i=1}^{\#chips} (x_i - \hat{x})^2 \sum_{i=1}^{\#chips} (y_i - \hat{y})^2}}$$

- Select most relevant features
  - In clustering genes, most meaningful chips
  - In clustering conditions, most meaningful genes

### 3. Clustering of Genes and Conditions

- Unsupervised:
  - Hierarchical clustering
  - K-means clustering
  - Self Organizing Maps (SOMs)
  - Singular Value Decomposition (SVD)
- Supervised:
  - Support Vector Machines
    - Could be useful to separate patient from non-patient genes and samples

### Results of Clustering Gene Expression



### 4. Analysis of Clustered Data

- Statistical Significance of Clusters
- Regulatory motifs responsible for common expression
- Regulatory Networks
- Experimental Verification

### C. Finding Regulatory Motifs

Tiny Multiple Local Alignments of Many Sequences

### Finding Regulatory Motifs



Given a collection of genes with common expression,  
Find the TF-binding motif in common

### Characteristics of Regulatory Motifs

```

ATAAATAA  TTTT
CTGATAA  A...CAAG
GTGA      T...CA...
AGGAGG  AG...CG
AA      ...AA...AA
TTT...T  ...AA...
G...AA...CG...TTGCG
...A...  TT...A...T...A
xTT...A  T...A...T...A...A
...GGAGAGG...
...AA...ATTT...
A...GA...AA...AA
T...AT...AA...TT...
...AA...AA...AAAA
TTT...A...AA...AA
...T...T...T...AA...AA
...AT...AT...T...AT...A
AT...AA...TT
  
```

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

## Problem Definition

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

### Probabilistic

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

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

### Combinatorial

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

**Find**  $M$  that occurs in all  $s_i$  with  $\leq k$  differences

## Essentially a Multiple Local Alignment



- **Find** "best" multiple local alignment

Alignment score defined differently in probabilistic/combinatorial cases

## Algorithms

- Probabilistic
  1. Expectation Maximization:  
MEME
  2. Gibbs Sampling:  
AlignACE, BioProspector
- Combinatorial  
CONSENSUS, TEIRESIAS, SP-STAR, others

## Discrete Approaches to Motif Finding

## Discrete Formulations

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

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

Definition of "best":

$d(W, x) = \text{min hamming dist. between } W \text{ and a word in } x$

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

## Approaches

- Exhaustive Searches
- CONSENSUS
- MULTIPROFILER, TEIRESIAS, SP-STAR, WINNOWER

## Exhaustive Searches

### Pattern-driven algorithm:

For  $W = AA...A$  to  $TT...T$  ( $4^K$  possibilities)  
Find  $d(W, S)$   
Report  $W^* = \text{argmin}(d(W, S))$

Running time:  $O(K N 4^K)$   
(where  $N = \sum_i |x^i|$ )

## Exhaustive Searches (2)

### 2. Sample-driven algorithm:

For  $W =$  a  $K$ -long word in some  $x^i$   
Find  $d(W, S)$   
Report  $W^* = \text{argmin}(d(W, S))$   
**OR** Report a local improvement of  $W^*$

Running time:  $O(K N^2)$

## Exhaustive Searches (3)

- Problem with sample-driven approach:
- If:
  - True motif does not occur in data, and
  - True motif is “weak”
- Then,
  - random strings may score better than any instance of true motif

## CONSENSUS (1)

### Algorithm:

#### Cycle 1:

For each word  $W$  in  $S$   
For each word  $W'$  in  $S$   
Create alignment (gap free) of  $W, W'$   
Keep the  $C_1$  best alignments,  $A_1, \dots, A_{C_1}$

ACGGTTG , CGAACTT , GGGCTCT ...  
ACGCCTG , AGAACTA , GGGGTGT ...

## CONSENSUS (2)

### Algorithm (cont'd):

#### Cycle I:

For each word  $W$  in  $S$   
For each alignment  $A_j$  from cycle  $I-1$   
Create alignment (gap free) of  $W, A_j$   
Keep the  $C_1$  best alignments  $A_1, \dots, A_{C_1}$

## CONSENSUS (3)

- $C_1, \dots, C_n$  are user-defined heuristic constants

Running time:

$$O(N^2) + O(N C_1) + O(N C_2) + \dots + O(N C_n) \\ = O(N^2 + N C_{\text{total}})$$

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

## MULTIPROFILER

- Extended sample-driven approach

Given a K-long word  $W$ , define:

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

Idea:

Assume  $W$  is occurrence of true motif  $W^*$

Will use  $N_a(W)$  to correct "errors" in  $W$

## MULTIPROFILER (2)

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  $W$

Example:  $K = 7$ ;  $L = 3$

$W$	=	ACGTTGA
$G$	=	--A--CG

## MULTIPROFILER (2)

Algorithm:

For each  $W$  in  $S$ :

For  $L = 1$  to  $L_{\max}$

- Find all "strong"  $L$ -long wordlets  $G$  in  $N_a(W)$
- Modify  $W$  by the wordlet  $G$   $\rightarrow W'$
- Compute  $d(W', S)$

Report  $W^* = \text{argmin } d(W', S)$

Step 1: Smaller motif-finding problem;

Use exhaustive search

## Expectation Maximization in Motif Finding

## Expectation Maximization (1)

- The MM algorithm, part of MEME package uses Expectation Maximization

Algorithm (sketch):

- Given genomic sequences find all  $K$ -long words
- Assume each word is **motif** or **background**
- Find **likeliest** motif & background models, and classification of words

## Expectation Maximization (2)

- Given sequences  $x^1, \dots, x^N$ ,
- Find all  $k$ -long words  $X_1, \dots, X_n$
- Define motif model:  
 $M = (M_1, \dots, M_k)$   
 $M_i = (M_{i1}, \dots, M_{i4})$  (assume  $\{A, C, G, T\}$ )  
where  $M_{ij} = \text{Prob}[\text{motif position } i \text{ is letter } j]$
- Define background model:  
 $B = B_1, \dots, B_4$   
 $B_i = \text{Prob}[\text{letter } j \text{ in background sequence}]$



### Expectation Maximization (3)

- Define  
 $Z_{i0} = \{ 1, \text{ if } X_i \text{ is motif;} \}$   
 $0, \text{ otherwise } \}$   
 $Z_{i1} = \{ 0, \text{ if } X_i \text{ is motif;} \}$   
 $1, \text{ otherwise } \}$
- Given a word  $X_i = a[1] \dots a[K]$ ,  
 $P[X_i, Z_{i0}=1] = \lambda M_{1a[1]} \dots M_{Ka[K]}$   
 $P[X_i, Z_{i1}=1] = (1 - \lambda) B_{a[1]} \dots B_{a[K]}$

### Expectation Maximization (4)

Define:

Parameter space  $\theta = (M, B)$

Objective:

Maximize log likelihood of model:

$$\begin{aligned} \log P(X_1 \dots X_n, Z | \mathbf{q}, \mathbf{I}) &= \sum_{i=1}^n \sum_{j=1}^2 Z_{ij} \log(I_j P(X_i | \mathbf{q}_j)) \\ &= \sum_{i=1}^n \sum_{j=1}^2 Z_{ij} \log P(X_i | \mathbf{q}_j) + \sum_{i=1}^n \sum_{j=1}^2 Z_{ij} \log I_j \end{aligned}$$

### Expectation Maximization (5)

- Maximize expected likelihood, in iteration of two steps:

Expectation:

Find expected value of log likelihood:

$$E[\log P(X_1 \dots X_n, Z | \mathbf{q}, \mathbf{I})]$$

Maximization:

Maximize expected value over  $\theta, \lambda$

### Expectation Maximization (6): E-step

Expectation:

Find expected value of log likelihood:

$$\begin{aligned} E[\log P(X_1 \dots X_n, Z | \mathbf{q}, \mathbf{I})] &= \\ \sum_{i=1}^n \sum_{j=1}^2 E[Z_{ij}] \log P(X_i | \mathbf{q}_j) &+ \sum_{i=1}^n \sum_{j=1}^2 E[Z_{ij}] \log I_j \end{aligned}$$

where expected values of Z can be computed as follows:

$$Z_{ij} = \frac{I_j P(X_i | \mathbf{q}_j)}{\sum_{k=1}^2 I_k P(X_i | \mathbf{q}_k)}$$

### Expectation Maximization (7): M-step

Maximization:

Maximize expected value over  $\theta$  and  $\lambda$  independently

For  $\lambda$ , this is easy:

$$I_j^{NEW} = \arg \max_{I_j} \sum_{i=1}^n E[Z_{ij}] \log I_j = \sum_{i=1}^n \frac{Z_{ij}}{n}$$

### Expectation Maximization (8): M-step

- For  $\theta = (M, B)$ , define

$c_{jk} = E[\text{# times letter k appears in motif position j}]$

$c_{0k} = E[\text{# times letter k appears in background}]$

It easily follows:

$$M_{jk}^{NEW} = \frac{c_{jk}}{\sum_{k=1}^4 c_{jk}} \quad B_k^{NEW} = \frac{c_{0k}}{\sum_{k=1}^4 c_{0k}}$$

to not allow any 0's, add pseudocounts

## Initial Parameters Matter!

Consider the following "artificial" example:

$x^1, \dots, x^N$  contain:

- $2^k$  patterns  $A\dots A, A\dots AT, \dots, T\dots T$
- $2^k$  patterns  $C\dots C, C\dots CG, \dots, G\dots G$
- $D \ll 2^k$  occurrences of K-mer  $ACTG\dots ACTG$

Some local maxima:

$\lambda \approx 1/2; \quad B = 1/2C, 1/2G; \quad M_i = 1/2A, 1/2T, i = 1, \dots, K$

$\lambda \approx D/2^{k+1}; \quad B = 1/4A, 1/4C, 1/4G, 1/4T;$   
 $M_1 = 100\% A, M_2 = 100\% C, M_3 = 100\% T, \text{ etc.}$

## Overview of EM Algorithm

1. Initialize parameters  $\theta = (M, B), \lambda$ :
  - Try different values of  $\lambda$  from  $N^{-1/2}$  upto  $1/(2K)$
2. Repeat:
  - a. Expectation
  - b. Maximization
3. Until change in  $\theta = (M, B), \lambda$  falls below  $\epsilon$
4. Report results for several "good"  $\lambda$

## Conclusion

- One iteration running time:  $O(NK)$ 
  - Usually need  $< N$  iterations for convergence, and  $< N$  starting points.
  - Overall complexity: unclear – typically  $O(N^2K)$  -  $O(N^3K)$
- EM is a local optimization method
- Initial parameters matter
- MEME: Bailey and Elkan, ISMB 1994.

## Gibbs Sampling in Motif Finding

## Gibbs Sampling (1)

- **Given:**
    - $x^1, \dots, x^N$ ,
    - motif length  $K$ ,
    - background  $B$ ,
  - **Find:**
    - Model  $M$
    - Locations  $a_1, \dots, a_N$  in  $x^1, \dots, x^N$
- Maximizing log-odds likelihood ratio:

$$\sum_{i=1}^N \sum_{k=1}^K \log \frac{M(k, x_{a_i+k}^i)}{B(x_{a_i+k}^i)}$$

## Gibbs Sampling (2)

- AlignACE: first statistical motif finder
- BioProspector: improved version of AlignACE

### Algorithm (sketch):

1. Initialization:
  - a. Select random locations in sequences  $x^1, \dots, x^N$
  - b. Compute an initial model  $M$  from these locations
2. Sampling Iterations:
  - a. Remove one sequence  $x^i$
  - b. Recalculate model
  - c. Pick a new location of motif in  $x^i$  according to probability the location is a motif occurrence

## Gibbs Sampling (3)

### Initialization:

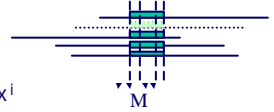
- Select random locations  $a_1, \dots, a_N$  in  $x^1, \dots, x^N$
- For these locations, compute M:

$$M_{kj} = \frac{1}{N} \sum_{i=1}^N (x_{a_i+k} = j)$$

- That is,  $M_{kj}$  is the number of occurrences of letter  $j$  in motif position  $k$ , over the total

## Gibbs Sampling (4)

### Predictive Update:



- Select a sequence  $x = x^i$
- Remove  $x^i$ , recompute model:

$$M_{kj} = \frac{1}{(N-1)+B} (b_j + \sum_{s=1, s \neq i}^N (x_{a_s+k} = j))$$

where  $\beta_j$  are pseudocounts to avoid 0s,  
and  $B = \sum_j \beta_j$

## Gibbs Sampling (5)

### Sampling:

For every K-long word  $x_j, \dots, x_{j+K-1}$  in  $x$ :

$$Q_j = \text{Prob}[\text{word} \mid \text{motif}] = M(1, x_j) \times \dots \times M(K, x_{j+K-1})$$

$$P_i = \text{Prob}[\text{word} \mid \text{background}] = B(x_i) \times \dots \times B(x_{i+K-1})$$

Let 
$$A_j = \frac{Q_j / P_j}{\sum_{j=1}^{|x|-K+1} Q_j / P_j}$$



Sample a random new position  $a_i$  according to the probabilities  $A_1, \dots, A_{|x|-K+1}$ .

## Gibbs Sampling (6)

### Running Gibbs Sampling:

1. Initialize
2. Run until convergence
3. Repeat 1,2 several times, report common motifs

## Advantages / Disadvantages

- Very similar to EM

### Advantages:

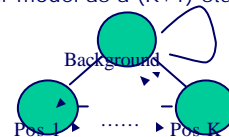
- Easier to implement
- Less dependent on initial parameters
- More versatile, easier to enhance with heuristics

### Disadvantages:

- More dependent on all sequences to exhibit the motif
- Less systematic search of initial parameter space

## Gibbs Sampling vs. Viterbi Training

- Consider model as a (K+1)-state HMM:



### Viterbi Training:

1. Find best  $\pi^* = \text{argmax}(\text{Prob}[x, \pi])$  in all sequences
2. Recalculate parameters

- Gibbs: one sequence, sample from  $\text{Prob}[x, \pi]$

## Repeats, and a Better Background Model

- Repeat DNA can be confused as motif
  - Especially low-complexity CACACA... AAAAA, etc.

Solution: more elaborate background model

0<sup>th</sup> order:  $B = \{ p_A, p_C, p_G, p_T \}$

1<sup>st</sup> order:  $B = \{ P(A|A), P(A|C), \dots, P(T|T) \}$

...

K<sup>th</sup> order:  $B = \{ P(X | b_1 \dots b_K); X, b_i \in \{A, C, G, T\} \}$

Has been applied to EM and Gibbs (up to 3<sup>rd</sup> order)

## Applications

### Application 1: Motifs in Yeast

Group:

Tavazoie et al. 1999, G. Church's lab, Harvard

Data:

- Microarrays on 6,220 mRNAs from yeast Affymetrix chips (Cho et al.)
- 15 time points across two cell cycles

### Processing of Data

#### 1. Selection of 3,000 genes

Genes with most variable expression were selected

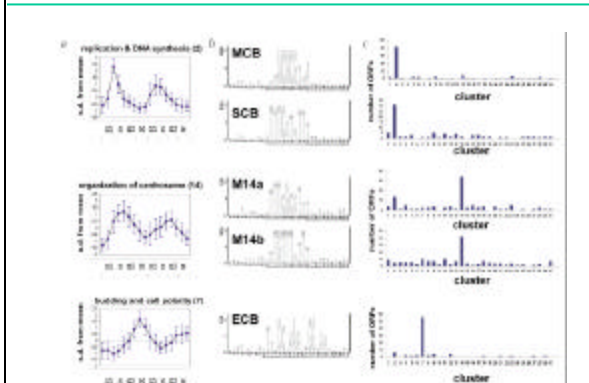
#### 2. Clustering according to common expression

- K-means clustering
- 30 clusters, 50-190 genes/cluster
- Clusters correlate well with known function

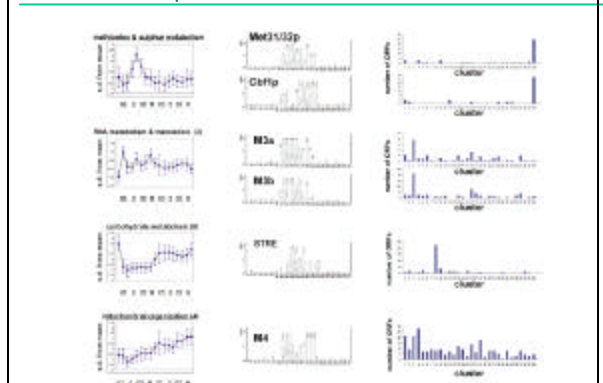
#### 3. AlignACE motif finding

- 600-long upstream regions
- 50 regions/trial

### Motifs in Periodic Clusters



### Motifs in Non-periodic Clusters



## Application 2: Discovery of Heat Shock Motif in *C. Elegans*

### Group:

GuhaThakurta et al. 2002, C.D. Link's lab & colleagues

### Data:

- Microarrays on 11,917 genes from *C. Elegans*
- Isolated genes upregulated in heat shock

## Processing of Data, and Results

- Isolated 28 genes upregulated in heat shock during 5 separate experiments
- Motif finding with CONSENSUS and ANNSpec on 500-long upstream regions
- 2 motifs found:
  - TTCTAGAA: known heat shock factor (HSF)
  - GGGTGTC: previously unreportedConserved in comparison with *C. Briggsae*
- Validation by in vitro mutagenesis of a GFP reporter

## Phylogenetic Footprinting

(Slides by Martin Tompa)

## Phylogenetic Footprinting

(Tagle et al. 1988)

Functional sequences evolve slower than nonfunctional ones

- Consider a set of *orthologous* sequences from different species
- Identify unusually well conserved regions

## Substring Parsimony Problem

### Given:

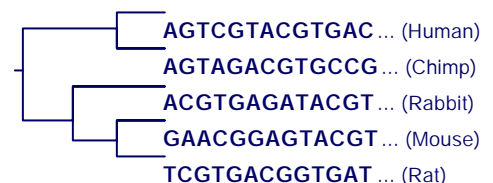
- phylogenetic tree  $T$ ,
- set of orthologous sequences at leaves of  $T$ ,
- length  $k$  of motif
- threshold  $d$

### 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  $d$ .

This problem is *NP*hard.

## Small Example



Size of motif sought:  $k = 4$



# Improvements

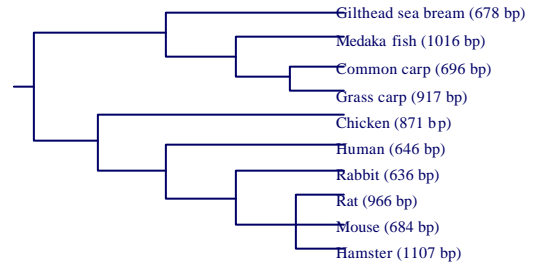
- Better algorithm reduces time from  $O(n k (4^{2k} + l))$  to  $O(n k (4^k + l))$
- By restricting to motifs with parsimony score at most  $d$ , greatly reduce the number of table entries computed (exponential in  $d$ , polynomial in  $k$ )
- Amenable to many useful extensions (e.g., allow insertions and deletions)

- Better algorithm reduces time from  $O(n k (4^{2k} + l))$  to  $O(n k (4^k + l))$
- By restricting to motifs with parsimony score at most  $d$ , greatly reduce the number of table entries computed (exponential in  $d$ , polynomial in  $k$ )
- Amenable to many useful extensions (e.g., allow insertions and deletions)

## Application to *b-actin* Gene

A phylogenetic tree showing the evolutionary relationships between the *b-actin* gene in various species. The tree is rooted on the left and branches out to the right. The species and their corresponding *b-actin* gene lengths in base pairs (bp) are listed on the right side of the tree.

- Gilthead sea bream (678 bp)
- Medaka fish (1016 bp)
- Common carp (696 bp)
- Grass carp (917 bp)
- Chicken (871 bp)
- Human (646 bp)
- Rabbit (636 bp)
- Rat (966 bp)
- Mouse (684 bp)
- Hamster (1107 bp)

[illegible]

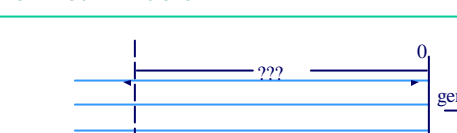
ACGGACTGTGTACCACTTCAACGGCACTCAACTGCGACAGAGAAATCTCAAGACAGCAATGGCATGGCTTTGTTATTTTGGCGTGACTCAG  
GATCTAAAACTGGAACTGGAGGAGGAGGCACTGTTGGCAATCAATCCCGAAGTCTCAAGATGCACTGGAGGCACTGTTATTTT  
TTTTTTCTTGATTCGATTCCTAAATGTTGTTGAAGGCTGTGCGCAACTTAACTGCTAGGATGAGGCTGGCCAGATGGGCACTTAACATGTGATGATATG  
TGTAAATATGTCAAAACCAATGACTGGGTTTGTGACTTTCAGCGCTTAATCTGGGTTTTTTTTTTTTTTTGGTCAAAACCAAGCTTACCATCAGATGATG  
AAGGTTTCTATCCCGTGGCATATGAAAAAGCTGTGGAGCGTGGCGTGCAGACATCTGCTGGGGCCACCTGTACATCAATAAAA  
TGCAGATCAATGCACTACTCTGCTGTGTTGTTGTTGTTGCTGAGCACTGCTGATCACTGCTGTTAGTGACGCTCTTAATAAAGAGTCTCTGCTTAAGGTG

[illegible][illegible][illegible]

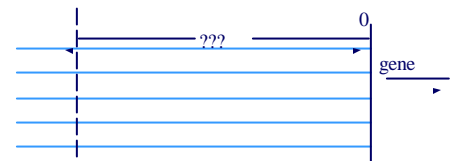
CGGACGATGACTTACTGGTGTACACCTTCTTGCCAAAACCTAACTGCGCAGAAACAG ATGAG **ATTGCGATGCGCT**TAFTTGTGTTTTTTGTGTTTGT  
TTGTTGTTTTTTTTTTTGTGTTG**CTGACTCAGTA****TA****AAAACTGGAA****CG**GTGAGGATGAGCAGCGTGTGTGGCGAGGACATCCCCCAAAAT  
CACAATGCGGCCGAGGAGCTTTGATTTGCTTTGTTT**TAAT****GTGATTCCTCAAA**ATTAGATGATGCTTTGTACAGGAAGCTCCCTGGCATCTAAAGAACCCACCC  
CTCTTCAAGGAGATAGCGCAGGCTGCTCCAGCATCCACACAGGGGAGGTGATGACGTTCTTG **TGTAATATGT****ATGT**AGCAAAATTTTTTTTAAATCTGCGGCTT  
ATATCACTTTTGTGTTTGTGTTT**TAAT**GTAGATGATGAGTGTGCGCCGCCCCCTCCCCCTTTTG TCCGCCAACTGAGATGATGAGGAGGAGCTTTGGTCTGCGGGAGGTG

Parsimony score over 10 vertebrates: 0 1 2

# Limits of Motif Finders



- Given upstream regions of coregulated genes:
  - Increasing length makes motif finding **harder** – random motifs clutter the true ones
  - Decreasing length makes motif finding **harder** – true motif missing in some sequences



- Given upstream regions of coregulated genes:
  - Increasing length makes motif finding harder – random motifs clutter the true ones
  - Decreasing length makes motif finding harder – true motif missing in some sequences

# Limits of Motif Finders

A (k,d)-motif is a k-long motif with d random differences per copy

Motif Challenge problem:

Find a (15,4) motif in N sequences of length L

CONSENSUS, MEME, AlignACE, & most other programs fail for  $N = 20$ ,  $L = 1000$

A (k,d)-motif is a k-long motif with d random differences per copy

Motif Challenge problem:

Find a (15,4) motif in N sequences of length L

CONSENSUS, MEME, AlignACE, & most other programs  
fail for  $N = 20$ ,  $L = 1000$