

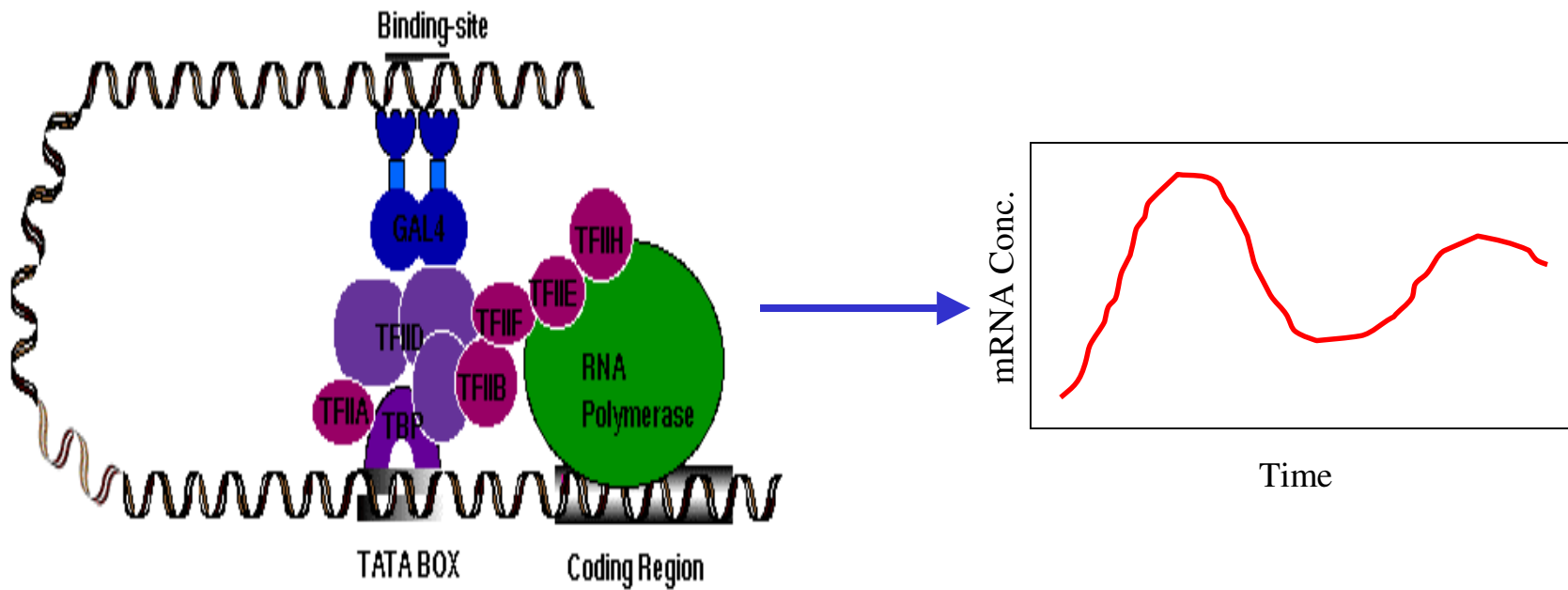
# Motif Finding: Summary of Approaches

4/26/03

# Lecture Outline

- Flashback: Gene regulation, the cis-region, and tying function to sequence
- Motivation
- Representation
  - Simple motifs
  - weight matrices
- Problem: Finding motifs in sequences
- Approaches
  - enumerative (combinatorial)
  - statistical
- Comparison of approaches
- Higher Order Motifs and Approaches

# Gene Regulation

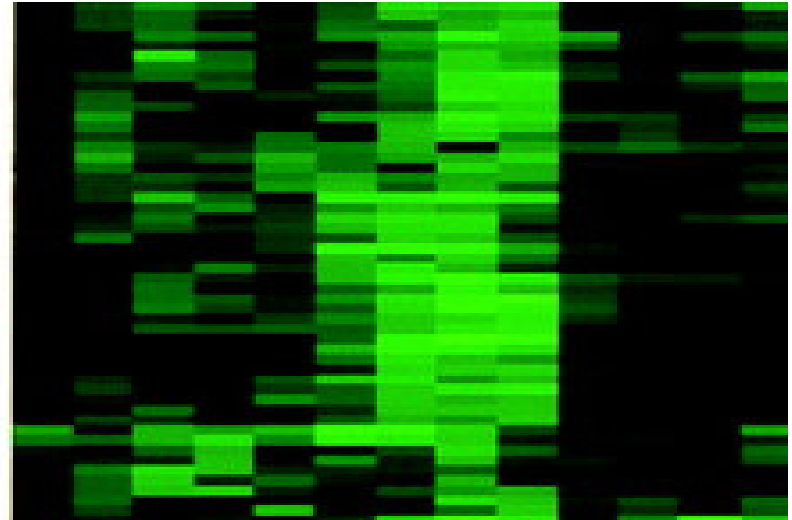


Sequence

Function

# Motif Finding Motivation

Clustering genes based on their expressions groups co-expressed genes



Assuming co-expressed genes are co-regulated, we look in their promoter regions to find conserved motifs, confirming that the same TF binds to them


# Co-expressed Genes Share Motifs

GTGGCTGCACCACGTGTATGC . . . ACGATGTCTC  
ACATCGCATCACGTGACCAGT . . . GACATGGACG  
CCTCGCACGTGGTGGTACAGT . . . AACATGACTA  
CTCGTTAGGACCATCACGTGA . . . ACAATGAGAG  
GCTAGCCACGTGGATCTTGT . . . AGAATGGCCT



# Co-expressed Genes Share Motifs

GGCTGCAC**CACGTGT**ATGC . . . ACG**ATGTCTCGC**  
ATCGCAT**CACGTG**ACCAGT . . . GAC**ATGGACGGC**  
TCG**CACGTGGTGGT**ACAGT . . . AAC**ATGACTAAA**  
CGTTAGGACCAT**CACGTGA** . . . ACA**ATGAGAGCG**  
TAGCC**CACGTGGATCTTGT** . . . AGA**ATGGCCTAT**



# Co-expressed Genes Share Motifs

TCTGCAC**CACGTGT**ATGC . . . ACG**ATGTCTCGC**  
ATCGCAT**CACGTG**ACCAGT . . . GAC**ATGGACGGC**  
GCCTCG**CACGTG**GTGGTACAGT . . . AAC**ATGAC**  
GGACCAT**CACGTGA** . . . ACA**ATGAGAGCG**  
GCTAGCC**CACGTG**GATCTTGT . . . AGA**ATGGCC**

↓  
Protein binding

# Multi-site Motif

- Two-site: Dimer, dyad
- Gapped Motif
- In general, a motif is an ordered set of binding sites

**Table 3 • Dimer alignment  
for MCM1 binding site**

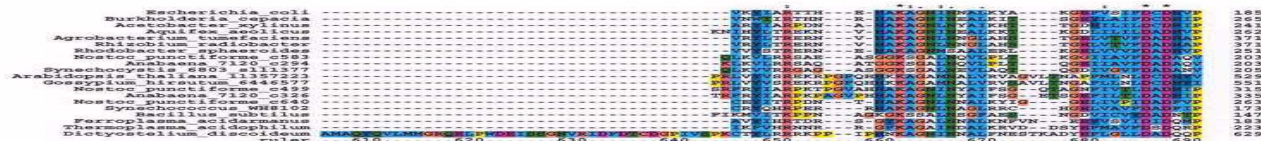
```
.ACC.....AGGA.  
.ACC.....GGAA  
..CCTA...AGGA.  
.ACCT...AAGG..  
..CCT.....GGAA  
..CCTA....GGAA  
TACC....AAGG..  
.ACCT.....GGA.  
.ACCT....AGGA.  
TACC.....GGA.  
TACC....AGGA.  
.ACCT.....GGAA  
TACC.....GGAA
```




# Motif Finding Problem

Given  $n$  sequences, find a motif present in many

- This is essentially multiple alignment
- Difference: multiple alignment is global
  - longer overlaps
  - constant site sizes and gaps
  - NP-complete!



# Definition and Representation

- Motifs: Short sequences
- IUPAC notation 
- Regular Expressions
  - consensus motif  
**ACGGGTA**
  - degenerate motif  
**RCGGGTM**  
**{G|A}CGGGT{A|C}**

Symbol	Meaning
G	G
A	A
T	T or U
C	C
U	U or T
R	G or A
Y	T, U or C
M	A or C
K	G, T or U
S	G or C
W	A, T or U
H	A, C, T or U
B	G, T, U or C
V	G, C or A
D	G, A, T or U
N	G, A, T, U or C

# Single Site Motif Finding

- Methods based on Position Weight Matrices (alignment)
  - Gibbs Sampling
  - Expectation Maximization
- Other Methods
  - HMMs
  - Bayesian methods
  - enumerative (combinatorial)

# Popular Software:

- MEME (EM)  
<http://meme.sdsc.edu/meme/website/intro.html>
- AlignACE (Gibbs)  
<http://atlas.med.harvard.edu/>
- Cister (HMM)  
<http://zlab.bu.edu/~mfrith/cister.shtml>
- YMF (combinatorial)  
<http://www.cs.washington.edu/homes/blanchem/software.html>
- MITRA (combinatorial)  
<http://www.cs.columbia.edu/compbio/mitra/>

# Overall Idea

- Enumerate motifs
- Score motifs base on their overrepresentation in all sequences
- The highest scoring ones, if occurring at surprising rates, are meaningful

Problems:

- How to enumerate?
- How to score motifs?
- What is surprise?

# PWM, main idea

- Capture the data in PWM
- Enumerate and score all patterns,  $w$ 
  - suffix trees used to save space
- Update the PWM
- Scoring: overrepresentation

$$S = \text{observed frequency} / \text{expected frequency}$$

$w$  in given sequences

$w$  in genome

# Position Specific Information

Seqs.

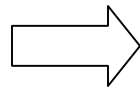
ACGGG

ATCGT

AAACC

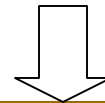
TTAGC

ATGCC



Alignment Matrix

Pos	A	C	G	T
1	4	0	0	1
2	1	1	0	4
3	2	1	2	0
4	0	2	3	0
5	0	3	1	1



Frequency Weight Matrix

Pos	A	C	G	T	Conse
1	0.8	0	0	0.2	A
2	0.2	0.2	0	0.6	T
3	0.4	0.2	0.4	0	A G
4	0	0.4	0.6	0	G
5	0	0.6	0.2	0.2	C

# Calculating the Joint Distribution

## Frequency Weight Matrix

Pos	A	C	G	T	Conse
1	0.8	0	0	0.2	A
2	0.2	0.2	0	0.6	T
3	0.4	0.2	0.4	0	A G
4	0	0.4	0.6	0	G
5	0	0.6	0.2	0.2	C

Given AAATC and the Weight Matrix of the data and for the background (i.e. prior), we want to calculate the joint probability

In general this is a lot of work, because of all possible ways a motif can depend on its sub-words.

E.g. TATTA=TAT.TA|TA.T.TA|T.A.T.T.A, etc.



# MEME

- Use Expectation-Maximization Algorithm to fit a two-component mixture model to the sequence data
- Component 1 is the motif
- Component 2 is the background

## Algorithm:

1. For each sequence  $s_i$ , (out of  $n$ )
2. Start with a random PWM,  $P_i$  (i.e. alignment)
3. Score every segment of  $s_i$  with  $P_i$
4.  $P_i = \text{Sum}$  all the scores with appropriate weights
5. Perform EM until there is a convergence

The best 100 scoring motifs are kept overall

# Gibbs Sampler

- Use a simple leave-one-out sampling strategy

## Algorithm

1. Given  $n$  sequences,  $s_1, s_2, \dots, s_n$
  2. Randomly initialize PWM (i.e. align)
  3. For each sequence  $s_i$ , take it out from the PWM
    - score each segment of  $s_i$  with the rest of the sequences
    - put the sequence back
- Important feature: convergence

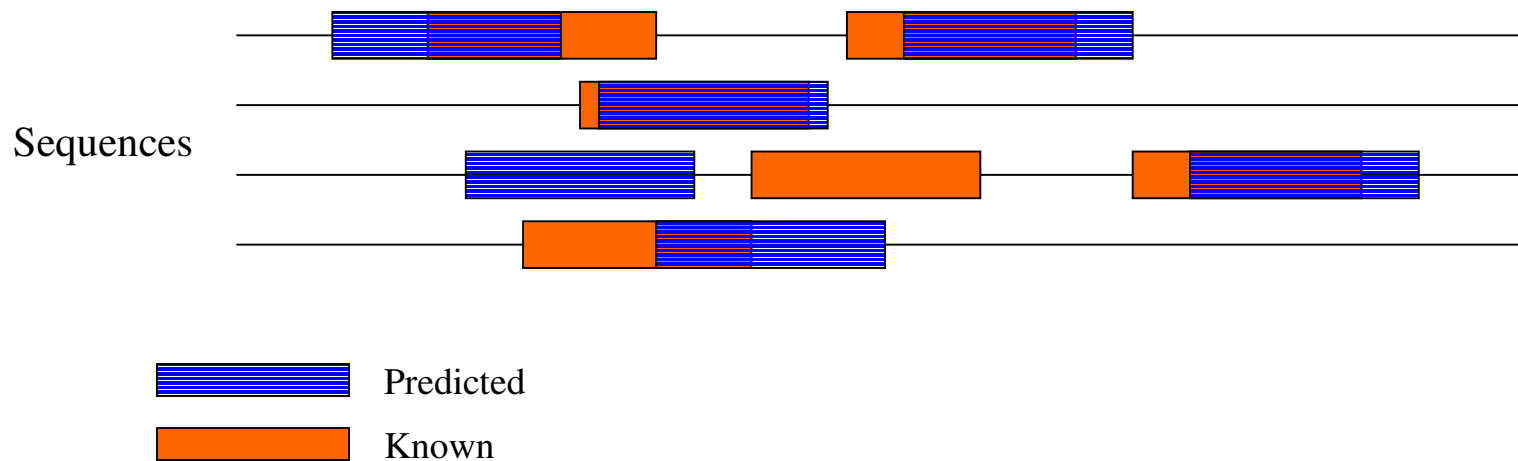
# Enumeration

- Use a consensus model of motifs based on IUPAC alphabet
- Score motifs based on their significance of occurrence (vs. random)
- Clean up the found motifs to remove redundant motifs

# Comparing the Methods

- Sinha and Tompa (2003)
- Scored motif finders: MEME, AlignACE and YMF
- Used synthetic sequences with planted motifs and yeast sequences
- Scored methods based on overlap of known and reported motifs

# Scoring Method Performance



$$\text{Score} = \text{Total overlap} / \text{Total span} \quad (\text{Pevzner \& Sze 2000})$$

Score = 1, if span = overlap

Score = 0, if overlap = 0

# Results

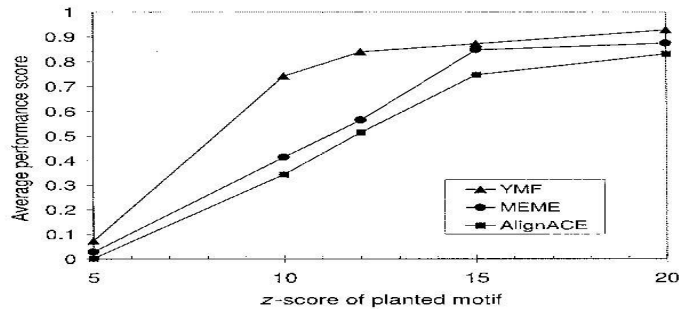


Figure 1: Performance of three motif-finding algorithms (YMF, MEME, and AlignACE) on 10 sequences of length 1000 each, with planted consensus motifs. Each point represents the average of the performance scores for a particular algorithm and for a specific z-score of the planted motif, the average being over 100 experiments, each using a different planted motif.

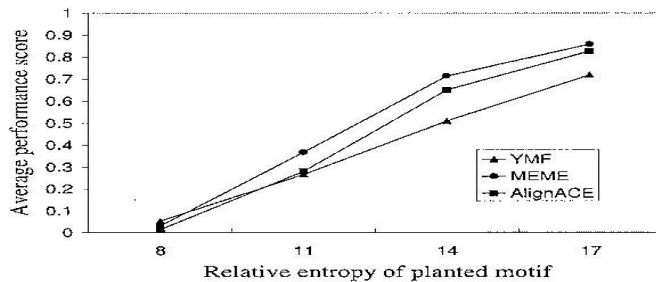


Figure 2: Performance of YMF, MEME, and AlignACE on 10 sequences of length 1000 each, with planted weight matrix motifs. Each point represents the average of the performance scores for a particular algorithm and for a specific strength of the planted motif, the average being over 50 experiments, each using a different planted motif.

Table 1: Performance comparison of different motif finders on yeast regulons. “Size” is the number of genes in the regulon. The columns labelled “time” report the time to completion for each algorithm, in seconds.

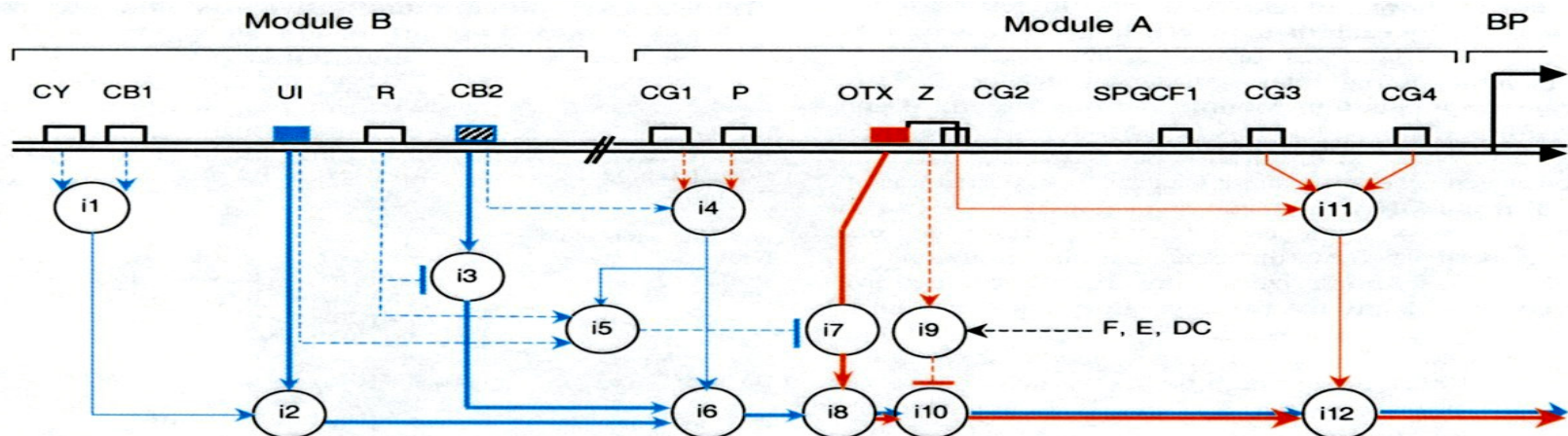
Regulon	Size	YMF		MEME		AlignACE	
		$\Phi_y$	time	$\Phi_m$	time	$\Phi_a$	time
ABF1	19	<u>0.33</u>	171	0.01	1645	0.00	28
BAS1	6	0.02	176	0.03	246	0.02	16
CAR1	12	<u>0.31</u>	151	0.25	771	0.20	18
CPF1	3	<u>0.62</u>	110	0.49	86	0.02	7
CSRE	4	0.28	125	<u>0.32</u>	149	0.25	7
GAL4	6	0.61	176	<u>0.66</u>	232	0.61	17
GATA	4	<u>0.57</u>	128	0.19	149	0.54	12
GCN	38	<u>0.25</u>	414	0.00	8523	0.00	64
GCR1	6	0.05	196	0.20	252	<u>0.31</u>	7
GLN3	3	0.00	129	0.00	83	0.00	10
HAP1	5	<u>0.15</u>	139	0.12	208	0.10	11
HAP2	4	0.00	93	0.00	150	0.02	9
HSE	6	<u>0.39</u>	158	0.23	247	0.31	21
MATA1	3	0.19	101	<u>0.20</u>	85	0.11	7
MATA2	7	0.06	197	<u>0.36</u>	359	0.03	17
MCB	6	0.54	122	0.15	238	<u>0.55</u>	22
MCM1	23	0.32	557	<u>0.51</u>	3532	0.50	24
MIG1	9	0.28	188	0.00	505	<u>0.29</u>	23
FDR3	7	<u>0.73</u>	174	0.43	357	0.47	13
PHO2	3	0.00	126	0.00	84	0.00	8
PHO4	5	<u>0.26</u>	161	0.05	209	0.22	12
RAP1	16	0.09	645	<u>0.31</u>	2036	0.23	26
REB1	14	<u>0.39</u>	396	0.34	1628	0.01	16
ROX1	3	0.00	90	0.03	83	0.00	2
RPA	3	<u>0.20</u>	99	0.15	80	0.00	8
SCB	3	0.60	137	0.61	85	<u>0.84</u>	7
SFF	3	0.00	136	0.00	80	0.05	11
STE12	4	0.60	176	0.02	144	<u>0.71</u>	12
TBP	17	0.00	379	0.00	2253	0.00	27
UASCAR	3	0.02	178	<u>0.13</u>	85	0.06	7
UASH	18	0.00	180	0.01	2301	0.00	39
UASPHR	17	0.01	556	0.02	2205	0.06	30
UIS	3	0.01	124	<u>0.43</u>	82	0.20	10
URS1H	13	0.57	388	<u>0.73</u>	1386	0.42	19
Wins		11		9		5	
#scores $\geq 0.2$		18		16		16	
#scores $\geq 0.33$		11		9		8	
#scores $\geq 0.5$		8		4		6	

# Results, contd.

- Results are a mixed bag
- YMF wins more often than not
- Each wins when motifs are specific to that algorithm
- Each algorithm wins on an exclusive set of motifs
- Take home lesson: use all on the same data

# Higher Order Motifs

- Nature of course is more complicated...



- Combinatorial motifs: combinations of binding sites to which an interacting group of TFs binds
- More realistic, but difficult to look for
- Sinha, 2002

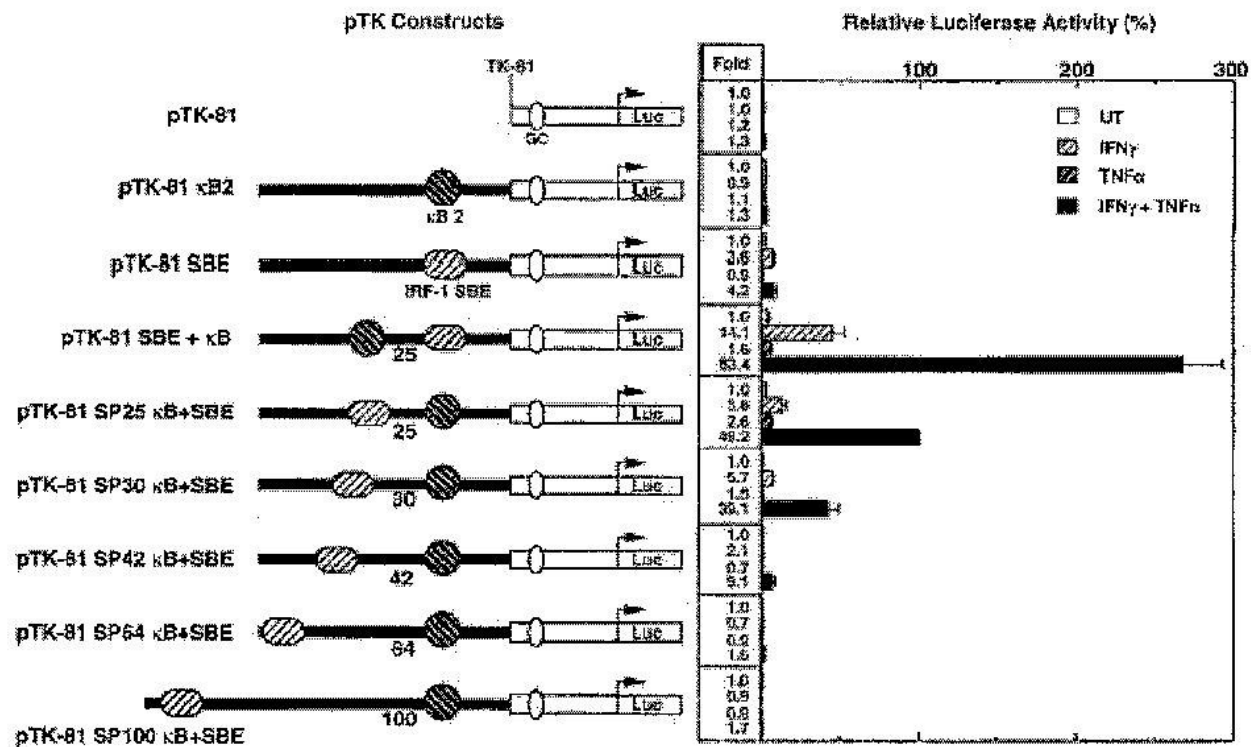


# What is Nature Like?

Now that we are talking about realistic motifs, what is it that we know about them from biology?

- Combinatorial motifs are sets of simple motifs separated by a stretch of DNA
- Changing the order of the simple motifs within it doesn't kill transcription, but changes it
- Changing the distance between the simple motifs usually kills transcription
- The distances between motifs are usually small (<20bp)
- The distance restriction is sometimes strict, and other times not
- Randomly distributed simple motifs do not activate transcription

# Dependence of Simple Motif Pairs on Distance and Order Between Them



Ohmori et al., 1997

# Finding Higher Order Motifs

Sinha (2002) reviews methods for finding higher order motifs, and groups the approaches based on their general relationship to simple motif finders

- find simple motifs and discover patterns made of these
- start with simple motifs and build higher order ones
- find higher order motifs from scratch (e.g. Marsan and Sagot, 2000)

# Models of Higher Order Motifs

- The set model  $\{M_1, M_2, \dots, M_k\}$
- Tuples with distance constraints  
( $M_1, M_2, d_{12}$ )
- Hidden Markov Model
- Boolean Combinations

Usually two step approaches:

- Enumerate the motif models
- Determine significance (Monte Carlo experiments)

# Tricky Business

- All these models have a lot of parameters (e.g. distances between motifs)
- They depend on the initial choice of parameters and/or an initial set of simple motifs
- Using these tools is more of an art than science so far

# Conclusions

- PWMs do well for simple motifs
- Combinatorial methods are probably doing better
- Should use all available tools to determine strong simple motifs
- Higher order motifs:
  - positive: knowing your biochemistry helps
  - negative: nobody knows the biochemistry fully!

# References:

- Saurabh Sinha, Ph.D. thesis, U of Washington, 2002
- Sinha and Tompa, *Performance Comparison of Algorithms for Finding Transcription Factor Binding Sites*, BIBE 2003
- Marsan and Sagot, *Algorithms for Extracting Structured Motifs Using a Suffix Tree*, JCB, v.7, 2000, 345-362
- Ohmori et al., *Journal of Biological Chemistry*, 1997