A Statistical Method for Finding Transcription Factor Binding Sites

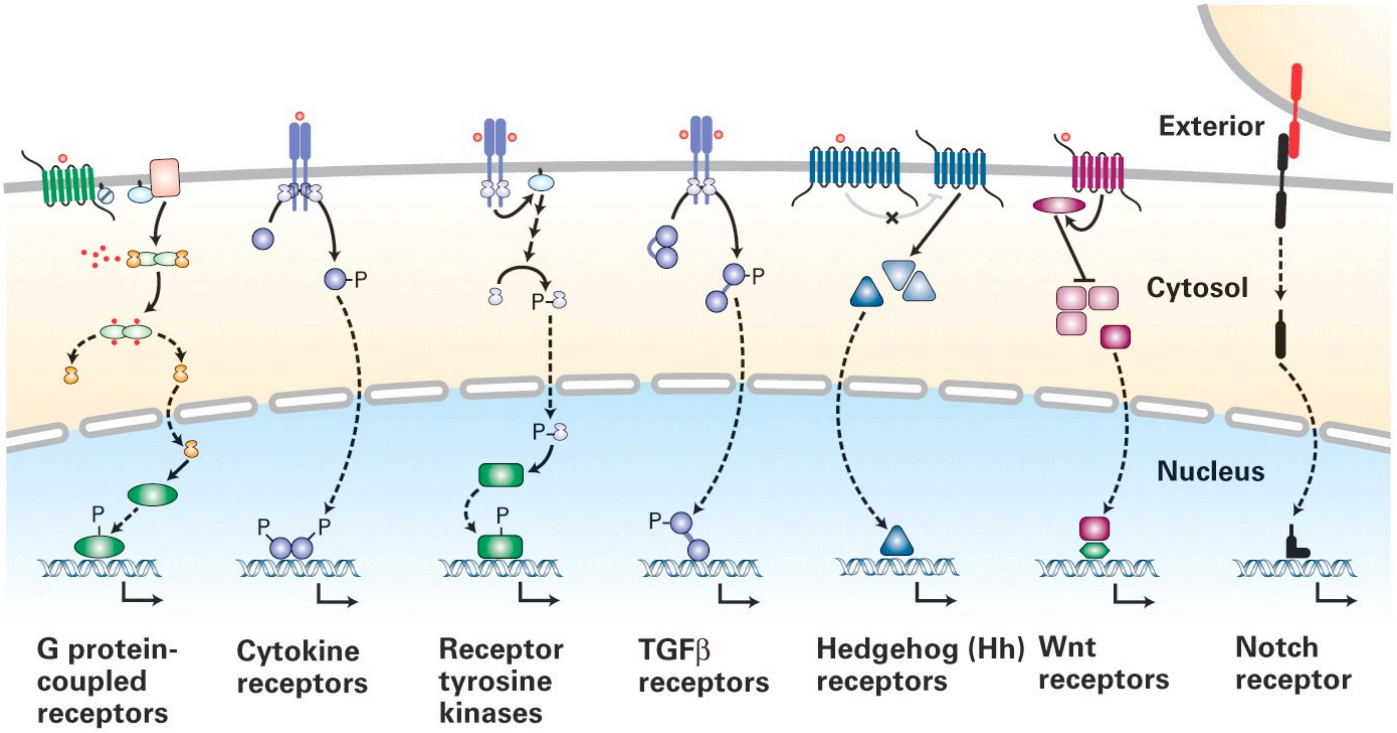
Saurabh Sinha and Martin Tompa

2000

Presented by Harley Cooper, Rohan Shah, and Robert Vincent
March 20 2006

Eukaryotic Regulatory Sequences

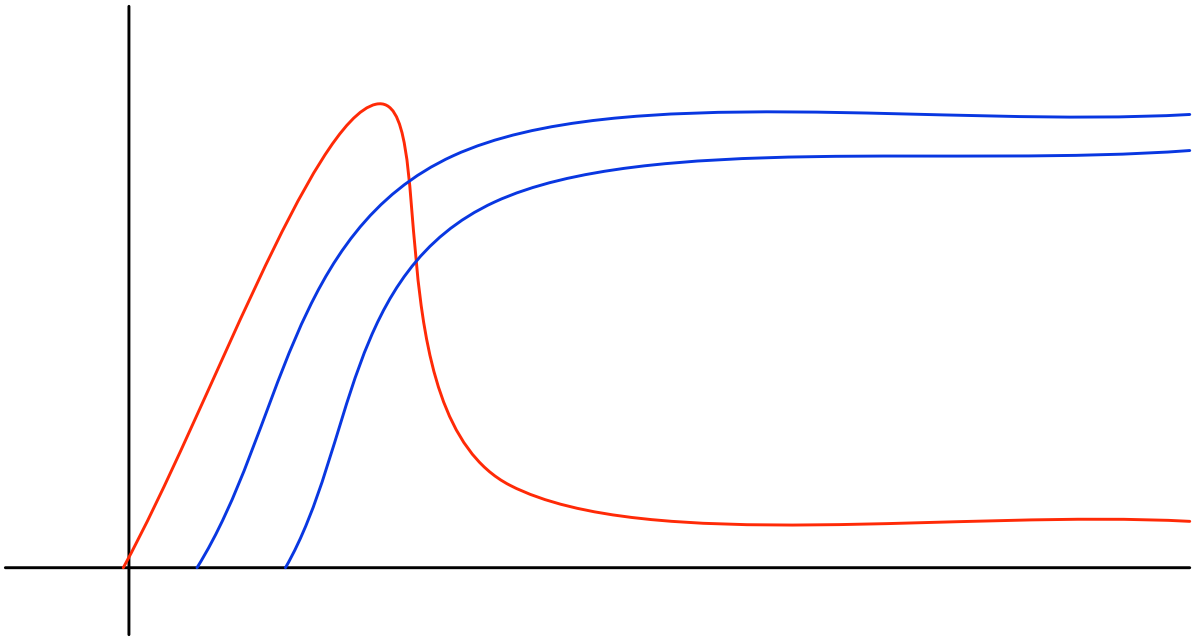
- Regulation of gene expression is a complex set of biochemical pathways



- The action of every transcription factor is regulated by many different chemical reactions throughout the cell

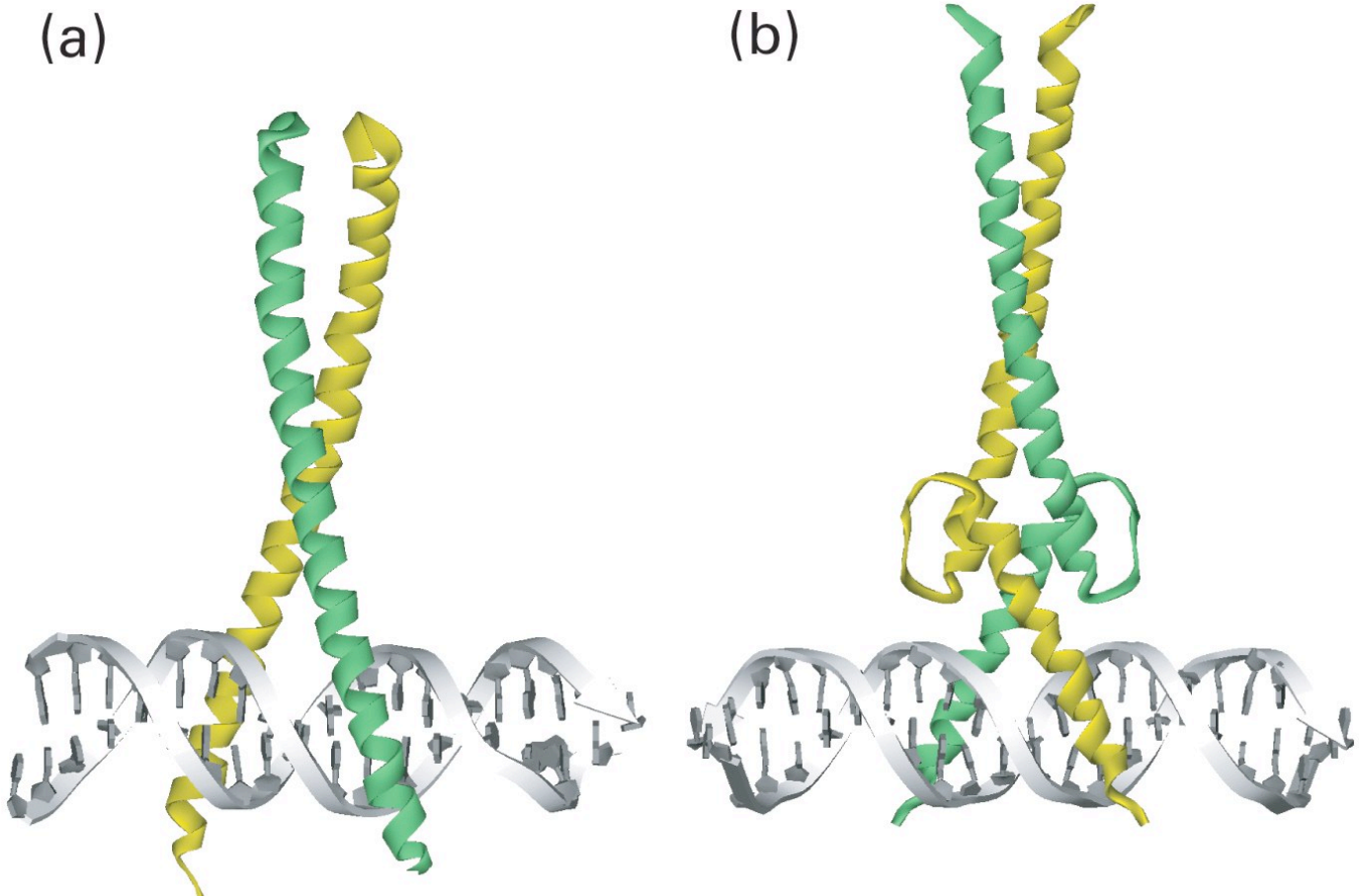
Coregulation

- Many studies have grouped genes into coregulated sets
- Genes are coregulated if their expression is governed by the same sets of transcription factors



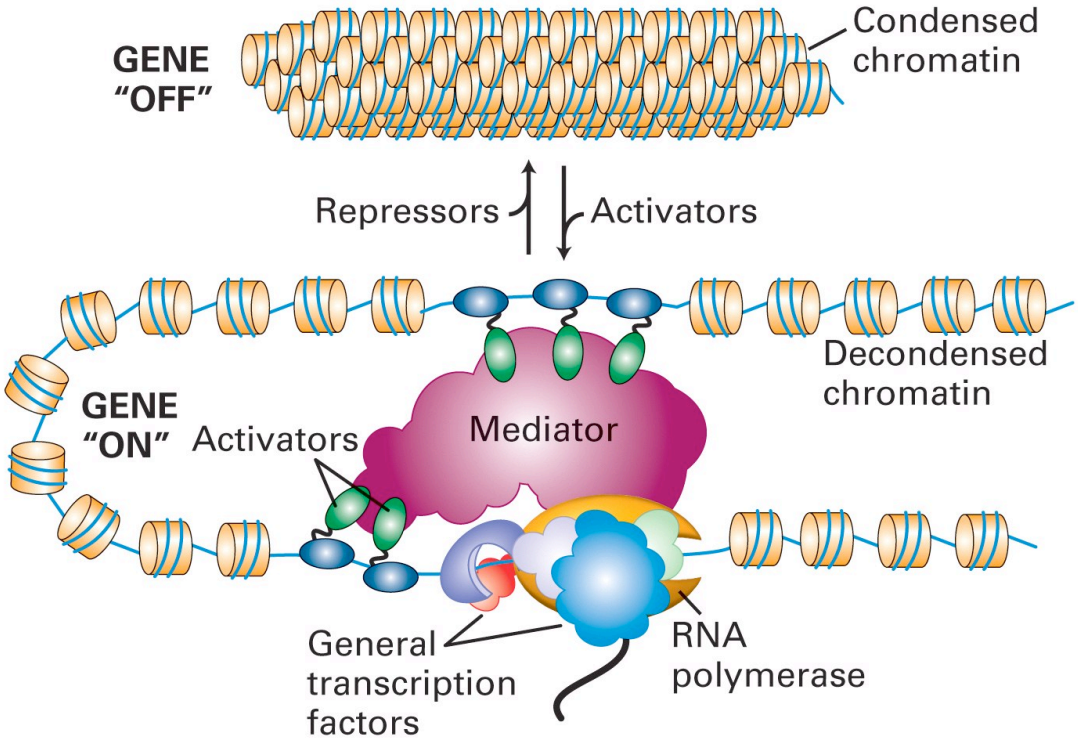
Binding Sites

- The key to the action of transcription factors is where they bind to the DNA
- The idea is that coregulated genes should have the same binding sites, and through the binding sites we can find these genes' common transcription factors



Difficulties – I

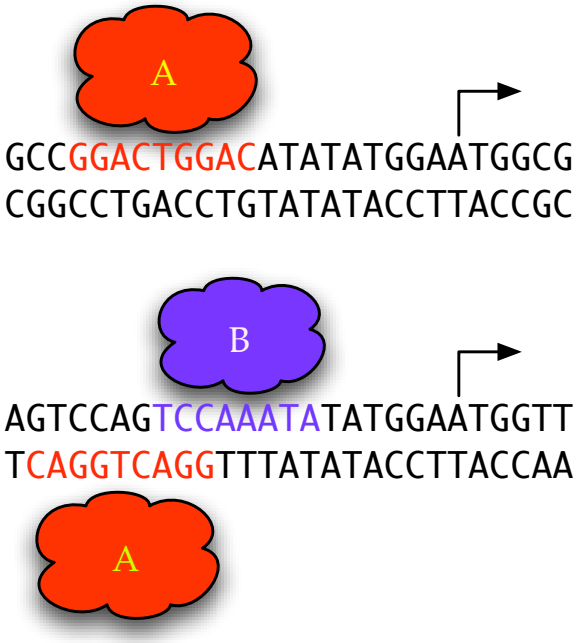
- Regulatory sequences may be far upstream



- This is not as much of a problem with *S. cerevisiae*, since their regulatory region begins around 800bp upstream of the coding sequence.
- In higher eukaryotes this becomes much more difficult, as regulatory regions are often longer than 10kb

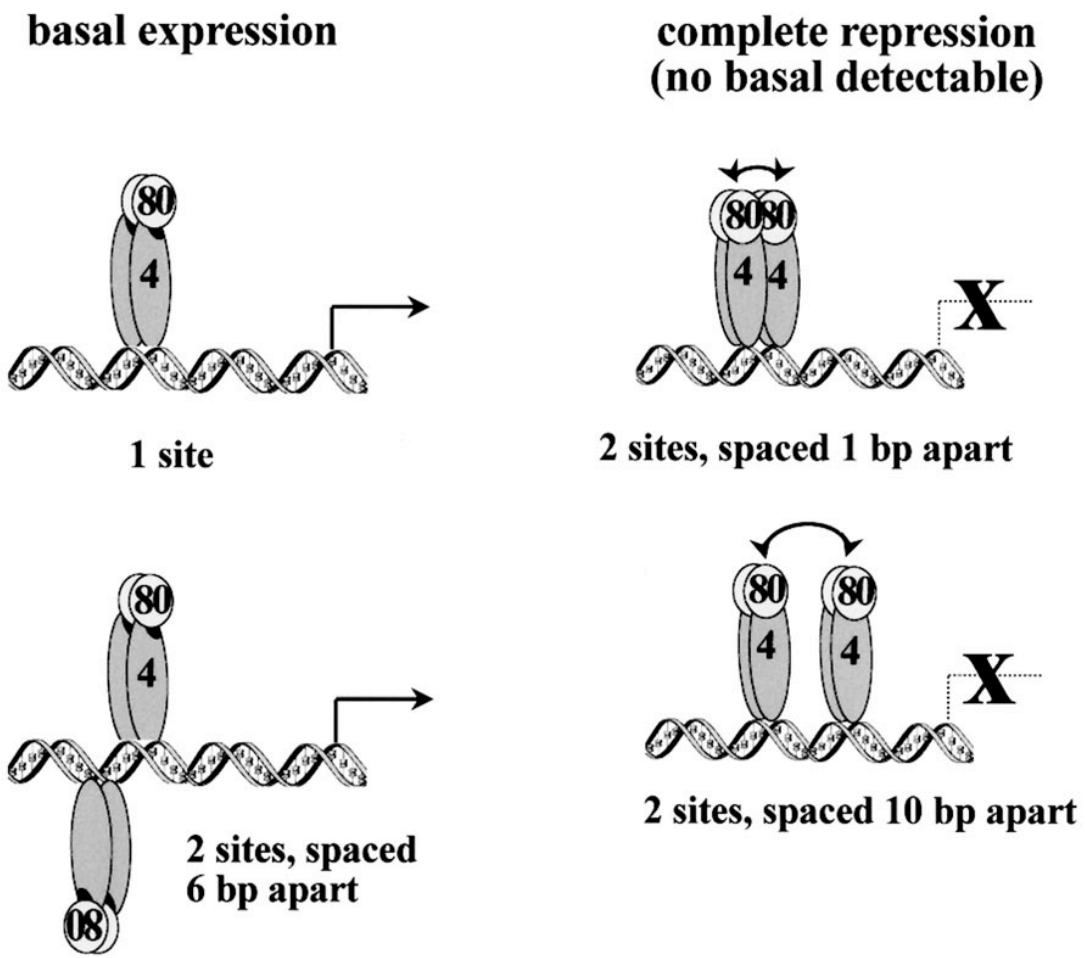
Difficulties – II

- The regulatory sequences are not necessarily in the same orientation as each other nor the coding sequences



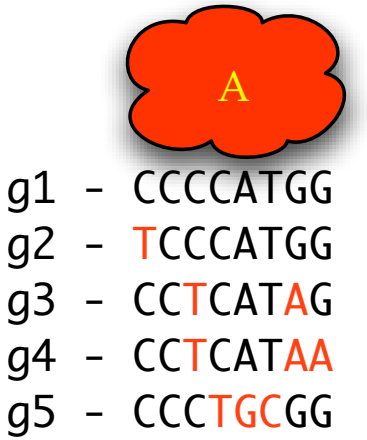
Difficulties – III

- Some transcription factors have several binding sites in one regulatory region (e.g: Gal4p)



Difficulties – IV

- One transcription factor can have high variability amongst its binding sites



- This sort of variability caused many problems, making it extremely difficult and time-consuming to find transcription factor binding sites

Drawbacks of Other Methods

- Only exact matches are allowed

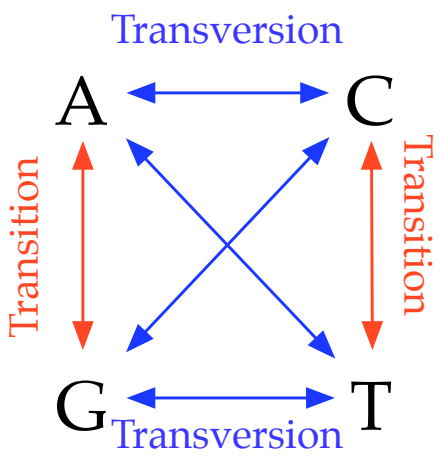
```
AGTCAACGTTA
AGTCAACGTTA
AGTCAACGTTA
AGTCAGCGTTA
```

Not Acceptable!

  though when variability was first incorporated, a maximum of 1 substitution was allowed
- No spacers (i.e: Gal4p consensus: `CGGNNNNNNNNNNNCCG`)
- All occurrences of a motif at distinct positions are assumed to be probabilistically independent, but in reality there are elaborate dependencies
- Rare motifs are under-represented and therefore have less statistical significance, making them much harder to accurately find
- No possibility of multiple sites for one TF with single genes

Variability Amongst Motif Instances

- Can't realistically expect exact matches
- Spacers of 1–11 base pairs are quite common in the middle of the motif due to TFs binding as dimers
- The number of conserved (non-spacer) bases ranges from 6–10 base pairs
- Variation is usually due to transitions rather than transversions (we use the alphabet `A,G,T,C,R,Y,W,S,N`)



- Due to the structure of the DNA binding domain, insertions and deletions are rare

Previous algorithms

- General methods:
  - Weight matrices or alignments
  - EM or Gibbs sampling
- Prior enumerative methods:
  - Exact matches or restricted number of spacers
  - Assumes likelihood of motif s is independent of position i
- General methods may not guarantee optimal results
- Enumerative methods are only practical with a small motif size.

Motif structure

In Yeast Motif Finder 3.0 (YMF), the exact number of nonspacers k and the minimum and maximum number of spacers n_{min},n_{max} are input parameters. Motifs have the following structure:

$$(s_1, \ldots s_{k/2})(\mathtt{N}^i)(s_{k/2+1}, \ldots, s_k), \forall n_{min} \leq i \leq n_{max}$$

Where $s_i \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}, \mathtt{R}, \mathtt{Y}, \mathtt{S}, \mathtt{W}\}$.

$$\mathtt{R} \in \{\mathtt{A}, \mathtt{G}\}$$
$$\mathtt{Y} \in \{\mathtt{C}, \mathtt{T}\}$$
$$\mathtt{S} \in \{\mathtt{C}, \mathtt{G}\}$$
$$\mathtt{W} \in \{\mathtt{A}, \mathtt{T}\}$$

And at most c of the s_i are possibly chosen from $\{\mathtt{R}, \mathtt{Y}, \mathtt{S}, \mathtt{W}\}$.

Statistical approach

- Let U be a set of m upstream sequences having uniform length (typically 800).
- Let X be a set of m random DNA sequences generated by a 3rd-order Markov chain.
- Let N_s be the number of times motif s is found in U.
- Let X_s be the number of times motif s is found in X.
- Then the *z-score* of s is defined as:

$$z_s = \frac{N_s - E(X_s)}{\sigma(X_s)}$$

Accuracy will tend to increase as the size of U increases.

Algorithm inputs

- Set of m upstream sequences
- Number of nonspacer characters ($6 \leq k \leq 10$)
- Transition matrix for order-3 Markov chain constructed from all upstream sequences for the organism
- Other parameters
  - Range of spacer lengths
  - Maximum number of motifs to output
  - Maximum number c of $\{\mathtt{R}, \mathtt{Y}, \mathtt{W}, \mathtt{S}\}$ symbols in motifs
  - Absolute minimum count of required appearances

Algorithm procedure

Enumerate 4^k motifs for $s_i \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$
Set $z_{min} = -1000$
For $i = n_{min}$ to n_{max}:
    For each upstream sequence:
        Calculate index for each $(pre)(N^i)(suf)$
        Increment count[index]
    For all possible motifs $s_i \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}, \mathtt{R}, \mathtt{Y}, \mathtt{W}, \mathtt{S}\}$:
        Calculate closure over $s_i \in \{\mathtt{R}, \mathtt{Y}, \mathtt{W}, \mathtt{S}\}$
        Calculate total count of occurrences
        Prune motif if possible
        Calculate full z-score and save in result
Print result

Details of counting

All nucleotides stored as strings of the form:

$$\texttt{A} \rightarrow 0$$
$$\texttt{C} \rightarrow 1$$
$$\texttt{G} \rightarrow 2$$
$$\texttt{T} \rightarrow 3$$

This allows indices to be calculated using radix 4 math:

$$index = (prefix_4 * 4^{k_{suffix}}) + suffix_4$$

For example:

$$\texttt{CTGNNTAT} \rightarrow (132_4 * 4^3) + 303_4 = 1971_{10}$$

Counting must be performed twice for odd k

Details of closure

- The counting proceeds only over all "true" nucleotides
- The z-scores are calculated over all possible instances of a motif.
- For example, if we consider the motif: `WCTNNGGA`
- The algorithm must consider the number of occurrences of both `TCTNNGGA` and `ACTNNGGA`.

Details of pruning

The algorithm maintains a value z_{min} which is the lowest z-score included in the results so far.

1. Occurrence count N_s must exceed an absolute threshold (typically 2)
2. Given that $\sigma(X_s) \geq \sqrt{E(X_s) - E(X_s)^2}$, prune if

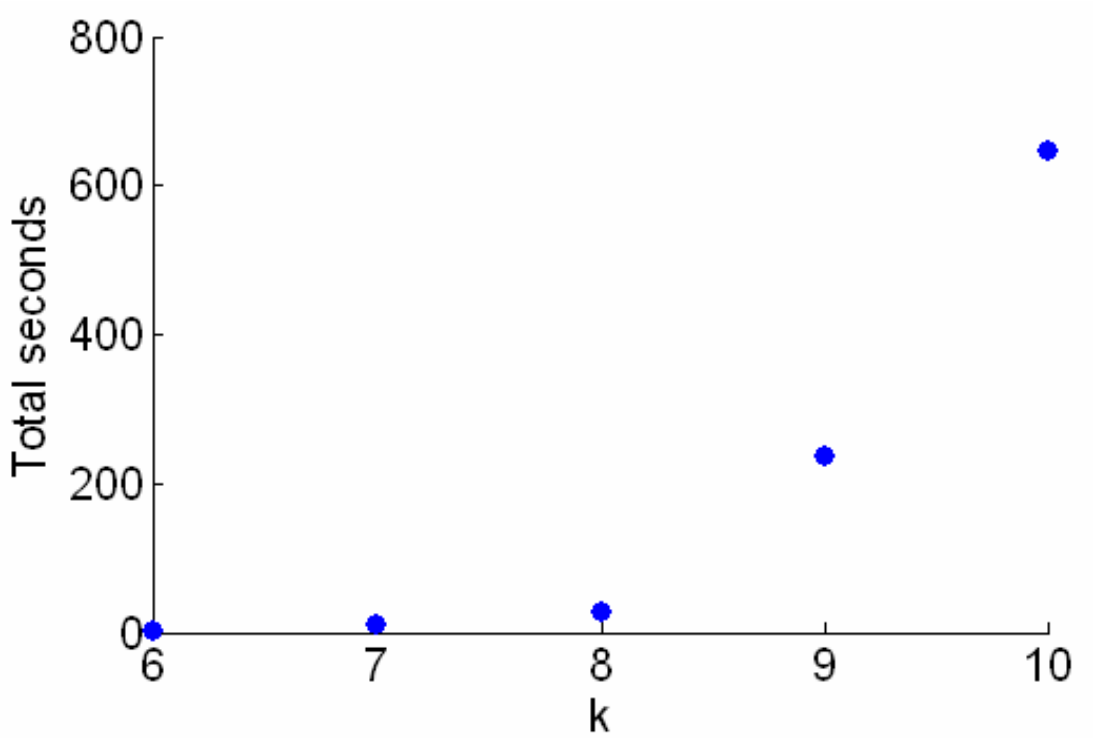
$$\frac{N_s - E(X_s)}{\sqrt{E(X_s) - E(X_s)^2}} < z_{min}$$

.

3. Estimate z-score while ignoring overlaps, prune if $z_{est} < z_{min}$

Complexity

- Linear in size of upstream sequence
- z-score computation is $O(k^2c^2)$ per motif
- Exponential in length of motif: $O(4^k)$



Implementation details

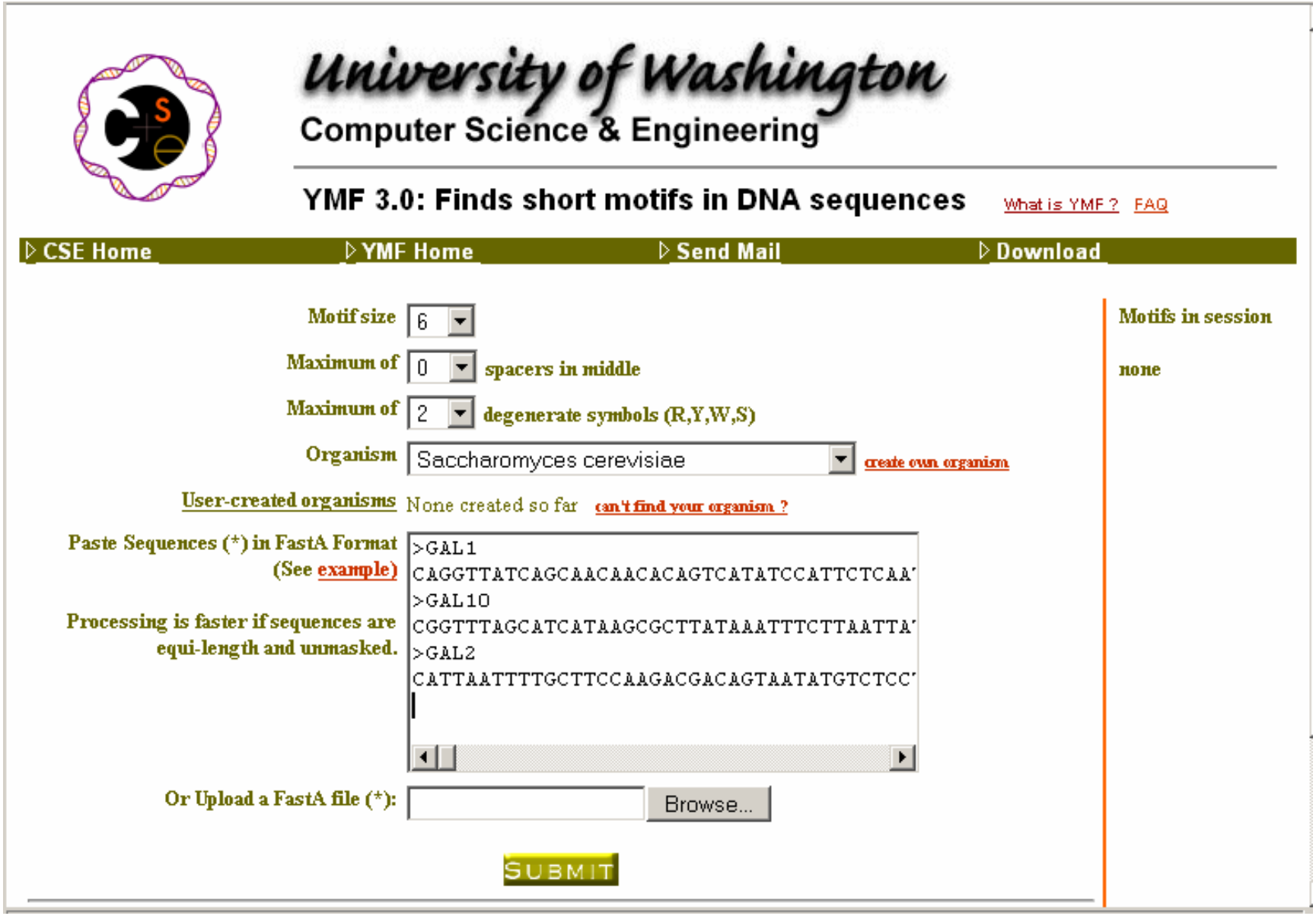
- **stats** - The main program
- **statsvar** - As above, but modified for variable sequence length
- **preproc** - Calculates 3rd-order Markov transition model for novel organisms given a set of upstream regions
- **findDivergent** - Find promoters with substantial overlap (e.g. divergent genes) to avoid duplicates
- **removeDivergent** - Use results from findDivergent to reorganize the set of input sequences

Post-processing

- Problem: YMF will return many artifacts of binding sites
  - Suppose `TCACGCT` is a "true" binding site.
  - YMF my report variations such as `TCACGCW` or `CACGCTT`.
- FindExplanators (Blanchette and Sinha, 2001):
  - Given: U, M, and τ
  - Find: Smallest $E \subset M$ s.t. $\forall m \in M, Z(m|E) < \tau$

Uses a greedy algorithm to add the "least explained" motif to E on each iteration.

Web interface



http://wingless.cs.washington.edu/YMF/YMFWeb/YMFInput.pl

- Let U be a set of m upstream sequences having uniform length (typically 800).
- Let X be a set of m random DNA sequences generated by a 1st-order Markov chain.
- Let N_s be the number of times motif s is found in U.
- Let X_s be the number of times motif s is found in X.
- Then the *z-score* of s is defined as:

$$z_s = \frac{N_s - E(X_s)}{\sigma(X_s)}$$

- Also define the set W as containing all the strings that result from replacing `R, Y, S and W` by all possible combinations of `A,C,G,T` in motif s and its reverse complement.

- Also define the set W as containing all the strings that result from replacing `R`, `Y`, `S` and `W` by all possible combinations of `A`,`C`,`G`,`T` in motif s and its reverse complement.
- Define X_w^a as the number of times $w \in W$ is found in $X^a \in X$ so that

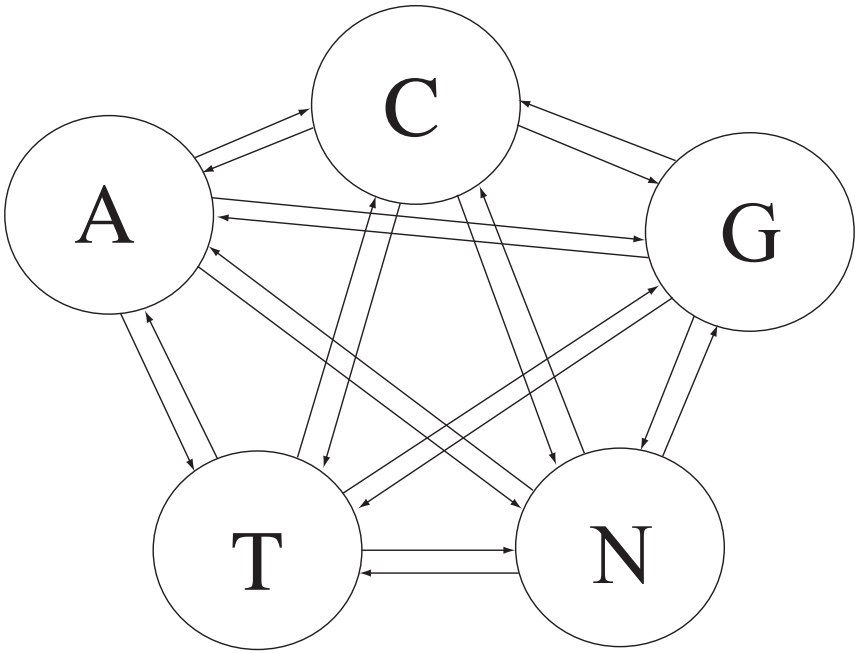
$$X_s = \sum_{X^a \in X} \sum_{w \in W} X_w^a$$

and

$$E(X_s) = \sum_{X^a \in X} \sum_{w \in W} E(X_w^a)$$

First Order Markov Chain

- consider a stochastic process $x_0, x_1, x_2, \cdots, x_\ell$ with values in $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}, \mathtt{N}\}$
- let the stochastic vector $b(t)$ be the distribution of x_t so $b_i(t) = P(x_t = i)$
- and define a probability transition matrix $P_{ij} = P(x_{t+1} = j | x_t = i)$



Generating A Random Sequence

- to generate a sequence of length n:
  - sample a value x_0 from $b(0)$

Generating A Random Sequence

- to generate a sequence of length n:
  - sample a value x_0 from $b(0)$
  - sample another value x_1 from the appropriate row of transition matrix P

Generating A Random Sequence

- to generate a sequence of length n:
  - sample a value x_0 from $b(0)$
  - sample another value x_1 from the appropriate row of transition matrix P
  - continue sampling another $n - 2$ values using the transition matrix P

Generating A Random Sequence

- to generate a sequence of length n:
  - sample a value x_0 from $b(0)$
  - sample another value x_1 from the appropriate row of transition matrix P
  - continue sampling another $n-2$ values using the transition matrix P
- for example if we take $n = 5$ and generate the sequence $S = \texttt{AGTTC}$ then $p(S) = b_A(0)P_{AG}P_{GT}P_{TT}P_{TC}$

Generating A Random Sequence

- to generate a sequence of length n:
  - sample a value x_0 from $b(0)$
  - sample another value x_1 from the appropriate row of transition matrix P
  - continue sampling another $n-2$ values using the transition matrix P
- for example if we take $n = 5$ and generate the sequence $S = \texttt{AGTTC}$ then $p(S) = b_A(0)P_{AG}P_{GT}P_{TT}P_{TC}$
- so the probability $p_j(w)$ that a word $w \in W$ of length ℓ occurs at position $j < n - \ell$ in a sequence $X^a \in X$ of length n is then

$$p_j(w) = b_{w_1}(j)P_{w_1 w_2}P_{w_2 w_3} \cdots P_{w_{\ell-1} w_\ell} \tag{1}$$

The First Moment

- Define an indicator variable I_j:

$$
\begin{aligned}
I_j &= 1 \texttt{ if } w \in W \texttt{ occurs at position j of } X^a \in X \\
&= 0 \texttt{ otherwise}
\end{aligned}
$$

The First Moment

- Define an indicator variable I_j:

$$\begin{aligned} I_j &= 1 \texttt{ if } w \in W \texttt{ occurs at position j of } X^a \in X \\ &= 0 \texttt{ otherwise} \end{aligned}$$

- then the expected number of times the word w occurs in the sequence X^a is

$$E(X_w^a) = \sum_{j=1}^{n-l+1} E(I_j) = \sum_{j=1}^{n-l+1} P(I_j = 1) = \sum_{j=1}^{n-l+1} p_j(w)$$

The First Moment

- Define an indicator variable I_j:

$$\begin{aligned} I_j &= 1 \texttt{ if } w \in W \texttt{ occurs at position j of } X^a \in X \\ &= 0 \texttt{ otherwise} \end{aligned}$$

- then the expected number of times the word w occurs in the sequence X^a is

$$E(X_w^a) = \sum_{j=1}^{n-l+1} E(I_j) = \sum_{j=1}^{n-l+1} P(I_j = 1) = \sum_{j=1}^{n-l+1} p_j(w)$$

$$= \sum_{j=1}^{n-l+1} b_{w_1}(j) P_{w_1 w_2} P_{w_2 w_3} \cdots P_{w_{\ell-1} w_\ell}$$

Perron-Frobenius Theory

- we know that $b(t+1) = P \times b(t) = P^t \times b(0)$

Perron-Frobenius Theory

- we know that $b(t+1) = P \times b(t) = P^t \times b(0)$
- the distribution of a *regular* markov chain always converges to its unique invariant distribution regardless of the initial distribution $b(0)$

$$\lim_{t \to \infty} P^t = \pi$$

Perron-Frobenius Theory

- we know that $b(t+1) = P \times b(t) = P^t \times b(0)$
- the distribution of a *regular* markov chain always converges to its unique invariant distribution regardless of the initial distribution $b(0)$

$$\lim_{t \to \infty} P^t = \pi$$

so

$$\lim_{t \to \infty} b(t) = \pi b(0)$$

Perron-Frobenius Theory

- we know that $b(t+1) = P \times b(t) = P^t \times b(0)$
- the distribution of a *regular* markov chain always converges to its unique invariant distribution regardless of the initial distribution $b(0)$

$$\lim_{t \to \infty} P^t = \pi$$

so

$$\lim_{t \to \infty} b(t) = \pi b(0)$$

- so we use an approximation and substitute the invariant distribution π for $b(t)$ and the first moment simplifies to

$$E(X_w^a) = p_j(w) = \sum_{j=1}^{n-l+1} b_{w_1}(0)\pi_{w_1} P_{w_1 w_2} P_{w_2 w_3} \cdots P_{w_{\ell-1} w_\ell}$$

Perron-Frobenius Theory

- we know that $b(t+1) = P \times b(t) = P^t \times b(0)$
- the distribution of a *regular* markov chain always converges to its unique invariant distribution regardless of the initial distribution $b(0)$

$$\lim_{t \to \infty} P^t = \pi$$

so

$$\lim_{t \to \infty} b(t) = \pi b(0)$$

- so we use an approximation and substitute the invariant distribution π for $b(t)$ and the first moment simplifies to

$$E(X_w^a) = p_j(w) = \sum_{j=1}^{n-l+1} b_{w_1}(0)\pi_{w_1} P_{w_1 w_2} P_{w_2 w_3} \cdots P_{w_{\ell-1} w_\ell}$$

$$= (n-l+1) b_{w_1}(0)\pi_{w_1} P_{w_1 w_2} P_{w_2 w_3} \cdots P_{w_{\ell-1} w_\ell}$$

The Overlapping Phenomenon

- consider the word `ATA`; the string of minimal length that contains at least 3 occurrences of this word is `ATATATA` which has length 7.

The Overlapping Phenomenon

- consider the word `ATA`; the string of minimal length that contains at least 3 occurrences of this word is `ATATATA` which has length 7.
- but for the word `ATC` we need a string of at least length 12 `ATCATCATCATC`

The Overlapping Phenomenon

- consider the word `ATA`; the string of minimal length that contains at least 3 occurrences of this word is `ATATATA` which has length 7.
- but for the word `ATC` we need a string of at least length 12 `ATCATCATCATC`
- so in a randomly generated string the word `ATA` is more likely to occur than `ATC`
- the distribution of X_s is affected by this

Overlaps

- define $w(i)$ as a prefix of length $i(< \ell - 1)$ of w and a **composite word**

$$cw(i) = w(i) + w$$

Overlaps

- define $w(i)$ as a prefix of length $i(< \ell - 1)$ of w and a **composite word**

$$cw(i) = w(i) + w$$

- if a prefix of $cw(i)$ contains w then we call this an **overlap**
- $\{cw\}$ gives us a uniquely defined set of overlaps for w

The Second Moment

- $$\sigma(X_s)^2 = E(X_s^2) - E(X_s)^2$$

The Second Moment

- $$\sigma(X_s)^2 = E(X_s^2) - E(X_s)^2$$

- let us assume that X and W are singleton sets then since $X_s = I_1 + I_2 + \cdots + I_{n-l+1}$ and $X_s^2 = (I_1 + I_2 + \cdots + I_{n-l+1})(I_1 + I_2 + \cdots + I_{n-l+1})$ we have:

The Second Moment

- $$\sigma(X_s)^2 = E(X_s^2) - E(X_s)^2$$

- let us assume that X and W are singleton sets then since $X_s = I_1 + I_2 + \cdots + I_{n-l+1}$ and $X_s^2 = (I_1 + I_2 + \cdots + I_{n-l+1})(I_1 + I_2 + \cdots + I_{n-l+1})$ we have:

$$E(X_s^2) = \sum_{i=1}^{n-l+1} \sum_{j=1}^{n-l+1} E(I_j I_k) = \sum_{i=1}^{n-l+1} E(I_i I_i) + 2 \sum_{j<k}^{n-l+1} E(I_j I_k)$$

The Second Moment

- $$\sigma(X_s)^2 = E(X_s^2) - E(X_s)^2$$

- let us assume that X and W are singleton sets then since $X_s = I_1 + I_2 + \cdots + I_{n-l+1}$ and $X_s^2 = (I_1 + I_2 + \cdots + I_{n-l+1})(I_1 + I_2 + \cdots + I_{n-l+1})$ we have:

$$E(X_s^2) = \sum_{i=1}^{n-l+1} \sum_{j=1}^{n-l+1} E(I_j I_k) = \sum_{i=1}^{n-l+1} E(I_i I_i) + 2 \sum_{j<k}^{n-l+1} E(I_j I_k)$$

- $I_j I_k$ indicates when a word has occured in both positions j and k simultaneously; suppose that $j < k$ and $k - j < l$ then there is an overlap $cw(j) \in \{cw\}$ at the position j
- so the variance of X_s is affected by the expected number of overlaps for each $cw(i) \in \{cw\}$

- we can also define composite words composed of different strings; say $cw_1(i) = w_1(i) + w_2$ and $cw_2(i) = w_2(i) + w_1$
- the co-variances between counts of words is affected by the expected number of overlaps $E(n(cw_1(i)))$ and $E(n(cw_2(i)))$

Results – Known Regulons

- Ran the program on seventeen known *S. cerevisiae* coregulated gene sets (i.e: the TF and the binding site consensus were already known)
- The algorithm was successful in 15 of the 17 gene sets. Of the 15,
  - 9 had the known consensus amongst the top three highest-scoring motifs
  - 6 had a very similar consensus in the top three
- Example of results:

s	N_s	z_s
TCANNNNNNACG	27	9.67
TCRNNNNNNACG	34	9.36
YCANNNNNNACG	34	8.58
TCANNNNNNWCG	37	8.39
YCANNNNNNWCG	52	8.31

  Known consensus: TCANNNNNNACG
- As for the other two sets, both having very few genes, the correct consensus was in the top twenty motifs

Results – Coexpressed Gene Clusters

- Ran the software on eight coexpressed gene clusters
- The top five motifs for four of the eight clusters matched the binding site consensus of the regulating transcription factor
- Example:

s	N_s	z_s
GACGNNNNNNGGAC	27	9.67
CTGCNNNNNGCAG	34	9.36
GCANNNCTGC	34	8.58
CAGANTCTG	37	8.39
CAGANNCTGC	52	8.31

Any Questions?

Bibliography

- Sinha, S. and Tompa, M. (2000) A statistical method for finding transcription factor binding sites. *Intelligent Systems for Molecular Biology* 2000.
- Kleffe, J. and Borodovsky , M. (1992) First and second moment of counts of words in random texts generated by Markov chains. *Computer Applications in the Biosciences* 8(5):433-441.
- Sinha, S. and Tompa, M. (2002) Discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Research* 30(24):5549-5560.
- Blanchette, M. and Sinha, S. (2001) Separating real motifs from their artifacts. *Bioinformatics* 17:S30-S38.
- Lodish et al. (2004) *Molecular Cell Biology* 5th ed.