Dictionary models for regulatory regions in DNA and gene expression arrays

Chiara Sabatti

Departmants of Human Genetics and Statistics, UCLA

MSRI, February 2004

Thanks to MSRI!

and go for the UC system!

Binding sites found experimentally

- Example of how a known binding site looks like (CRP):

```
>aldB -18->4
attcgtgatagctgtcgtaaag
>ansB 103->125
ttttgttacctgcctctaactt
>araB1 109->131
aagtgtgacgccgtgcaaataa
>araB2 147->169
tgccgtgattatagacactttt  [...]
```

(from `http://arep.med.harvard.edu/ecoli_matrices/`)

- Example of a database:

`http://transfac.gbf.de/TRANSFAC/`

contains 8415 entries, 4504 of them referring to sites near 1078 eukaryotic genes, the species of which ranging from yeast to human.

Finding binding sites computationally - Methods

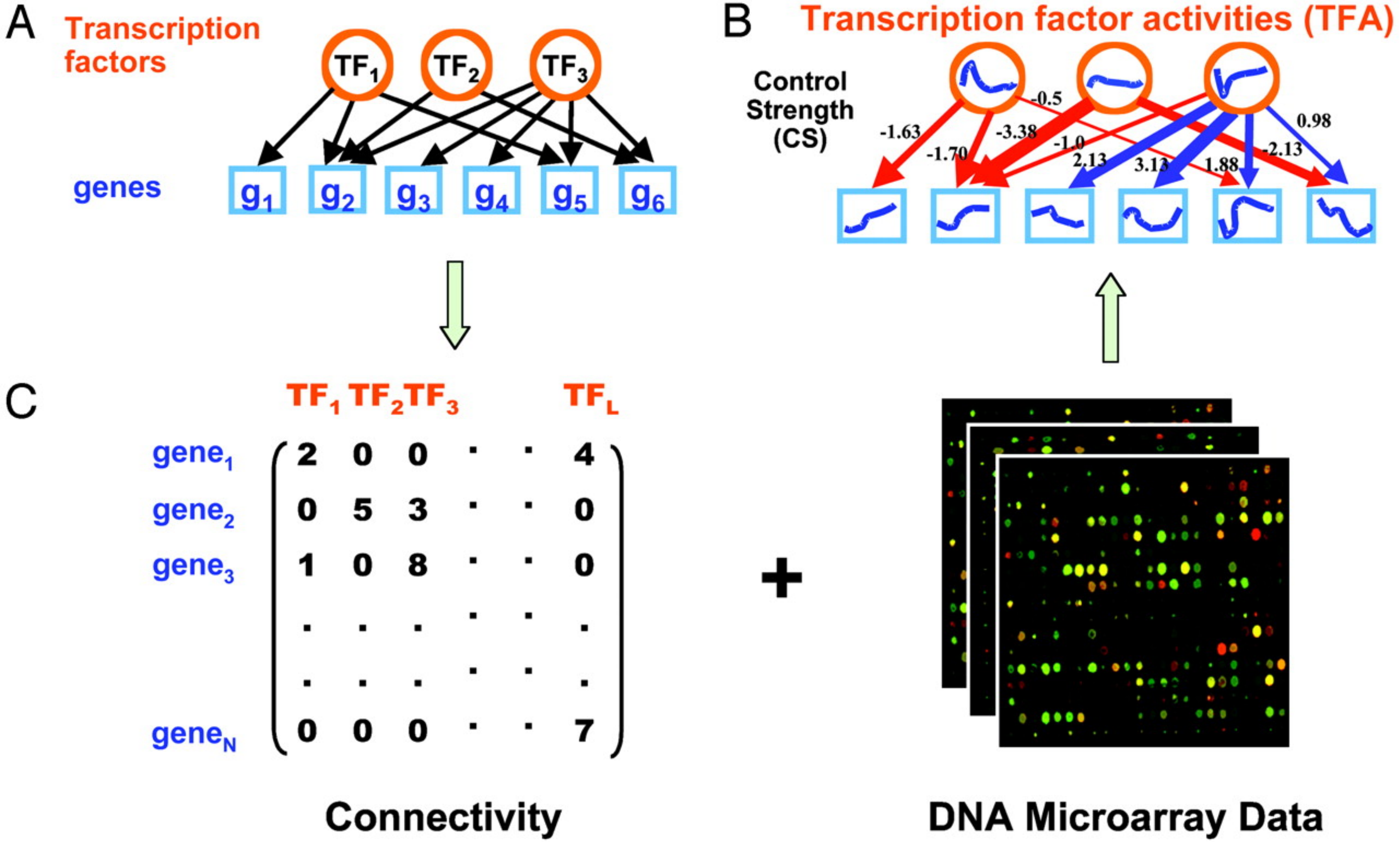
Identifying new motifs: (1) identify genes that are coregulated (maybe using array) and (2) search in the up-stream regions of their sequence for common motif. (3) Use a probabilistic model for motif and EM or Gibbs Sampler strategies. *Lawrence, C.E., Altschul, S.F., Bogouski, M.S., Liu, J.S., Neuwald, A.F., and Wooten, J.C. (1993), ”Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment,” Science, 262, 208-214.*

Finding occurrences of a known motif: Compare a sequence of interest against a collection of known motifs and known identify the ones that score high. *Quandt, K., K. Frech, H. Karas, E. Wingender, T. Werner, (1995) “MatInd and MatInspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data,” Nucleic Acids Res.l 23, 4878–4884.*

Using array data to help interpreting sequences

- Gathered a collection of “short” sequences enriched in few binding sites one can use exact word counts methods or a probabilistic description of binding sites with EM, Gibbs Sampler algorithms (Lawrence and Reilly, 1990, Lawrence et al. 1993, etc..)
- A long list of putative binding sites can be compiled for the up-stream region of each gene for which we have expression information and “significant” motifs are identified through regression of array experiments (Bussemaker et al. 2001, Keles et al. 2002, Colon et al. 2003)

Using binding site to extract information from array



$$\begin{array}{c|cccccc}
 & TF_1 & TF_2 & TF_3 & & & TF_L \\
\hline
gene_1 & 2 & 0 & 0 & \cdot & \cdot & 4 \\
gene_2 & 0 & 5 & 3 & \cdot & \cdot & 0 \\
gene_3 & 1 & 0 & 8 & \cdot & \cdot & 0 \\
 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
gene_N & 0 & 0 & 0 & \cdot & \cdot & 7
\end{array}$$

Liao et al. (2003)

Our problem

Find all the binding sites for a set of known regulatory proteins in a genome.

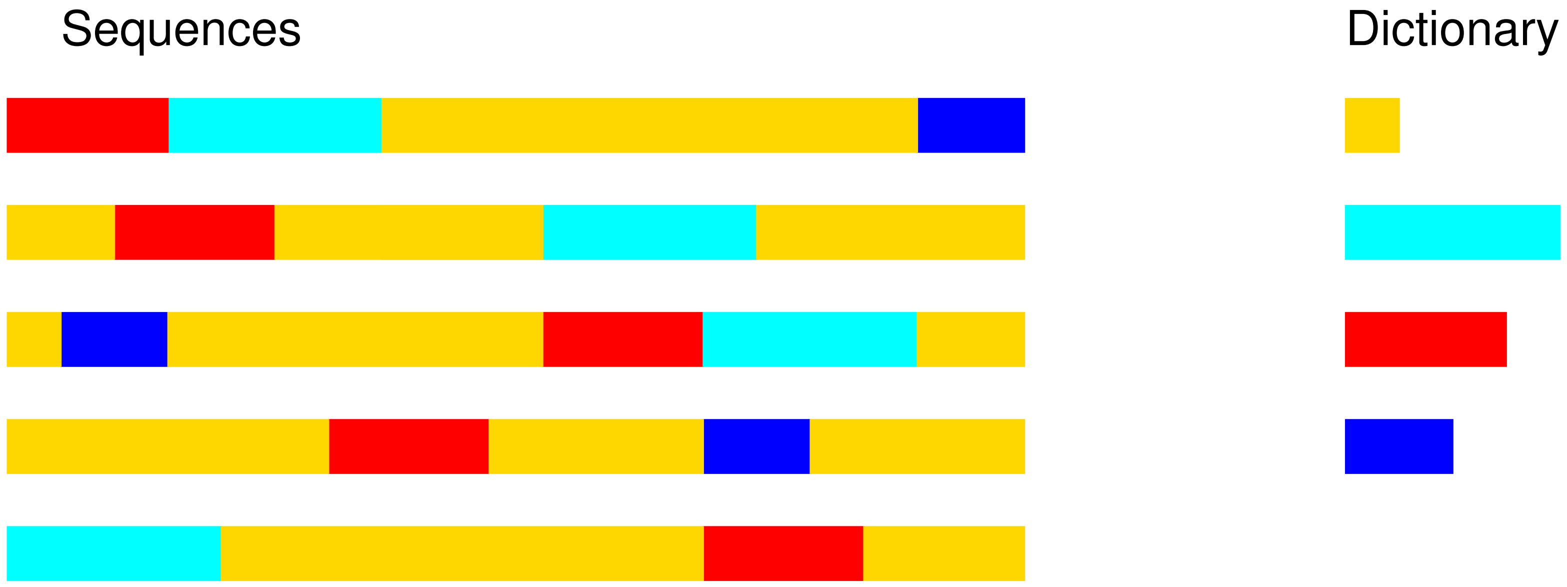
- There is prior information on the binding sites
- One expects many different binding sites occurring in only a fraction of the sequences
- One expects to find a significant new number of binding site for each protein

⟶ need a probabilistic model that

- allows incorporation of prior information
- models the entire sequence with multiple motifs
- allows calculation of probabilities of a motif being in a specific location

Dictionary models for DNA

- Proposed by Bussemaker et al (2000) PNAS 97:11096-10100
- The DNA sequence is the result of the concatenation of a series of words, chosen independently and with different probabilities.
- The DNA dictionary contains all the words and their probabilities.
- Words of length 1 play the role of background.



Sequences

Dictionary

Beyond the original Dictionary

- Words are deterministic: not practical for binding sites as described in the databases.
- We payed more attention to the normalizing constant of the likelihood.
- Reletated extension by Gupta and Liu.

The Vocabulon model - terminology

(joint with Kenneth Lange)

Word $\boldsymbol{w}$: irreducible semantic unit, or in the genetic context, a motif

Ex. *theater*:= a building for dramatic performances

Spelling

Ex. in English, *theater* and *theatre* represent the same word.

In our model a word $\boldsymbol{w}$ is always spelled with the same number of letters $|\boldsymbol{w}|$.

Segments $\boldsymbol{s}$: a block of letters from a sequence corresponding to one word

Ex. The segment *pot* can be the spelling of (1) a cooking utensil or (2) something to smoke.

Parameters of the dictionary

We need to specify
(1) the *words*;
(2) the *words probabilities*
(3) *probabilities of different spellings*

With regard to (1) and (2) it is actually useful to group words on the base of their length k and to impose a maximum word length $k_{\max}$ on our dictionary.

Currently we adopt the following parameterization for the dictionary:

- Probability of a word of length k: q_k, with $\sum_{k=1}^{k_{\max}} q_k = 1$. If there are no words of length k, then $q_k = 0$.
- Probability of word w, among the words of length $|w|$: r_w, with $\sum_{|w|=k} r_w = 1$.
- Independent multinomial distributions of parameters $\ell_{wi} = (\ell_{wiA}, \ell_{wiC}, \ell_{wiG}, \ell_{wiT})$ describing the letter that appears in the ith position of word w.

Length prob.	Word	Word prob.	Spelling prob.
			A C G T
q_1=0.8	Background	$r_w = 1$	$\ell_{w1*} = (.25, .25, .25, .25)$
			A C G T
			$\ell_{w1*} = (.80, .10, .05, .05)$
			$\ell_{w2*} = (.20, .10, .60, .10)$
q_5=0.2	w="AGTCA"	$r_w = .5$	$\ell_{w3*} = (.10, .25, .10, .55)$
			$\ell_{w4*} = (.10, .50, .20, .20)$
			$\ell_{w5*} = (.90, .05, .05, .00)$
			A C G T
			$\ell_{w1*} = (1.0, 0.0, 0.0, 0.0)$
			$\ell_{w2*} = (0.0, 1.0, 0.0, 0.0)$
	w= ACG-T	$r_w = .5$	$\ell_{w3*} = (0.0, 0.0, 1.0, 0.0)$
			$\ell_{w4*} = (.25, .25, .25, .25)$
			$\ell_{w5*} = (0.0, 0.0, 0.0, 1.0)$

Probability of a segment

Consider a generic segment $s = (s_1, \ldots, s_k)$ of given length. Then

$$p(s) \quad = \quad \sum_{|w|=k} r_w \prod_{i=1}^{k} \ell_{wis_i} \qquad (1)$$

is the conditional probability of s given that it is formed by a single word.

We represent missing letters by question marks ? and introduce the additional letter probability

$$\ell_{wi?} = 1$$

Probability of a sequence

- We observe a sequence s of $|s|$ letters. We do not know breakpoints between words and which words are spelled.

- A partition π divides s into $|\pi|$ segments $s[\pi_1], \ldots, s[\pi_{|\pi|}]$.

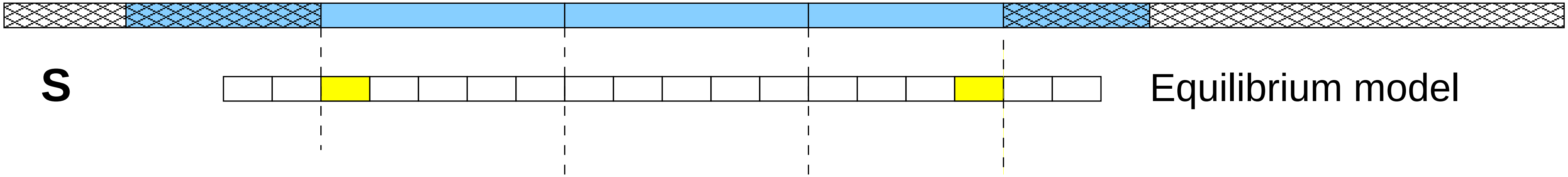
- Given π, the likelihood of the sequence is:

$$\mathcal{L}(S) \approx \sum_{\pi} \mathbf{Pr}(\pi) \prod_{j=1}^{|\pi|} p(s[\pi_i])$$

- $\mathbf{Pr}(\pi)$ depends on the exact model.

Equilibrium model

We observe a random portion of fixed length of an infinitely long text. The first and last words maybe only partially observed.



$$\mathbf{Pr}(\pi|\mathcal{E}) = \frac{\prod_{i=1}^{|\pi|} q_{|\pi_i|}}{\sum_{j=1}^{k_{\max}} j q_j}.$$

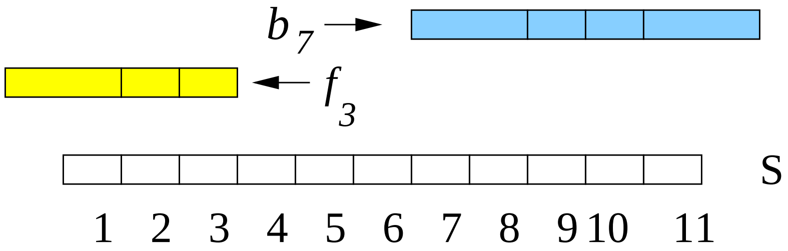
The sequence probability is then:

$$\mathcal{L}_E(s) = \frac{1}{\sum_{k=1}^{k_{\max}} k q_k} \sum_{\pi \in \mathcal{E}} \prod_{i=1}^{|\pi|} q_{|\pi_i|} p(s[\pi_i])$$

Algorithms for evaluation of sequence probability

Let $S[i:k]$ indicate portion of the sequence that goes from position i to position k.

Let B_i indicate the event that a word ends at position i. Let $n = |s|$



Forward (joint) probabilities $f_i := \mathbf{Pr}(S[1,i] = s[1,i], B_i)$

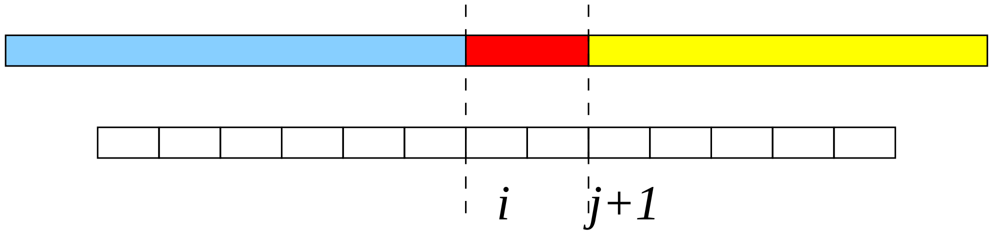
Backward (conditional) probabilities $b_i = \mathbf{Pr}(S[i,n] = s[i,n] \mid B_{i-1})$

Forward Algorithm Initialize as $f_i = 1/(\sum_{k=1}^{k_{\max}} kq_k)$ for $i = 1 - k_{\max}, \ldots, 0$.

Update recursively: $f_i = \displaystyle\sum_{k=\max\{1,i+1-n\}}^{k_{\max}} f_{i-k} q_k p(s[(i-k+1):i])$

$\mathcal{L}_E(s) = f_n + \cdots + f_{n+k_{\max}-1}$.

Probability that a portion of the observed sequence spells a word



The restriction that a particular word w fills this segment has conditional probability

$$\rho_{ij}(w) \quad = \quad \frac{f_{i-1} q_{|w|} r_w \prod_{k=1}^{j-i+1} \ell_{wks_{i+k-1}} b_{j+1}}{\mathcal{L}_E(s)}. \qquad (2)$$

$\Longrightarrow$ These are used to identify locations of binding sites.

Parameter estimation: missing data framework

Missing data:

1. the partition π segmenting the sequence
2. words assigned to the different segments of s generated by π.

For the equilibrium model, the complete data likelihood is

$$\frac{1}{\bar{q}} \prod_{i=1}^{|\pi|} q_{|\pi_i|} r_{w_i} \prod_{j=1}^{|w_i|} \ell_{w_i j s_{\pi_{ij}}},$$

where segment $s[\pi_i]$ is assigned word w_i, π_{ij} denotes the jth index of π_i, and $\bar{q} = \sum_{k=1}^{k_{\max}} k q_k$.

Augmented data log-likelihood and EM algorithm

M_k: number of segments of length k

N_w: number of appearances of word w

L_{wjt}: number of letters of type t occurring at position j of the segments assigned word w

$$\ln \mathcal{L} = \sum_{k=1}^{k_{\max}} M_k \ln q_k + \sum_w N_w \ln r_w + \sum_{w,i,j} L_{wij} \ln \ell_{wij} - \ln \bar{q}.$$

Expectation: Compute the expected values of M_k, N_w, and L_{wij} conditional on the data (easy)

Maximization: Maximize the expected log-likelihood with respect to the parameters.

Maximization Step

The expected log-likelihood separates the parameters q, r and ℓ.

- r and ℓ are easy
- The maximization with respect to q is more complicated. We need to maximize

$$g(q \mid q^m) \quad = \quad \sum_{k=1}^{k_{\max}} \mathrm{E}\left(M_k \mid S = s, q^m, r^m, \ell^m\right) \ln q_k - \ln\left(\sum_{k=1}^{k_{\max}} k q_k\right)$$

Use $\log(x) \leq \frac{x}{y} + \log(y) - 1$ to find a minorization and maximize that

(So it is really an MM algorithm)

Bayesian Framework

- Maximum likelihood $\longrightarrow$ Maximum a posteriori.
- The prior information on the parameters $\boldsymbol{q}, \boldsymbol{r}, \boldsymbol{\ell}$ easy to state with Dirichlet priors (they are even good for convergence purposes)
- The parameters of these priors can be set to summarize the experimental information available in the data-bases.

A dictionary of binding sites for E. Coli.

(in collaboration with Lars Rohlin and James Liao)

Goal: map all the locations of some* known binding sites in E. Coli.

Input:

- number of binding sites and their spellings, as represented in data-bases;
- sequence information on E. Coli: for each gene, we consider 700 bp, 600 prior to the start of the coding region and 100 after. If adjacent genes are known to be in an operon, we use the up-stream region of the first gene only.

Similar problem considered by Robinson (1998), where similarity scores are used. A results from C. Lawrence problem using multiple genomes.

$\ast$ Selected to avoid structural overlap and too little conserved binding sites.

Test 1: the set used to compile prior information

Difficult to test an algorithm, if we do not know the answer. A binding site may be present, but not discovered yet.

One possible test is to use the group of sequences that went into the binding site description.

- 233 sequences, with 430 motifs appearances.
- Generally, we call a motif, if the estimated probability that it appears at the given location is $> \mathbf{0.5}$.

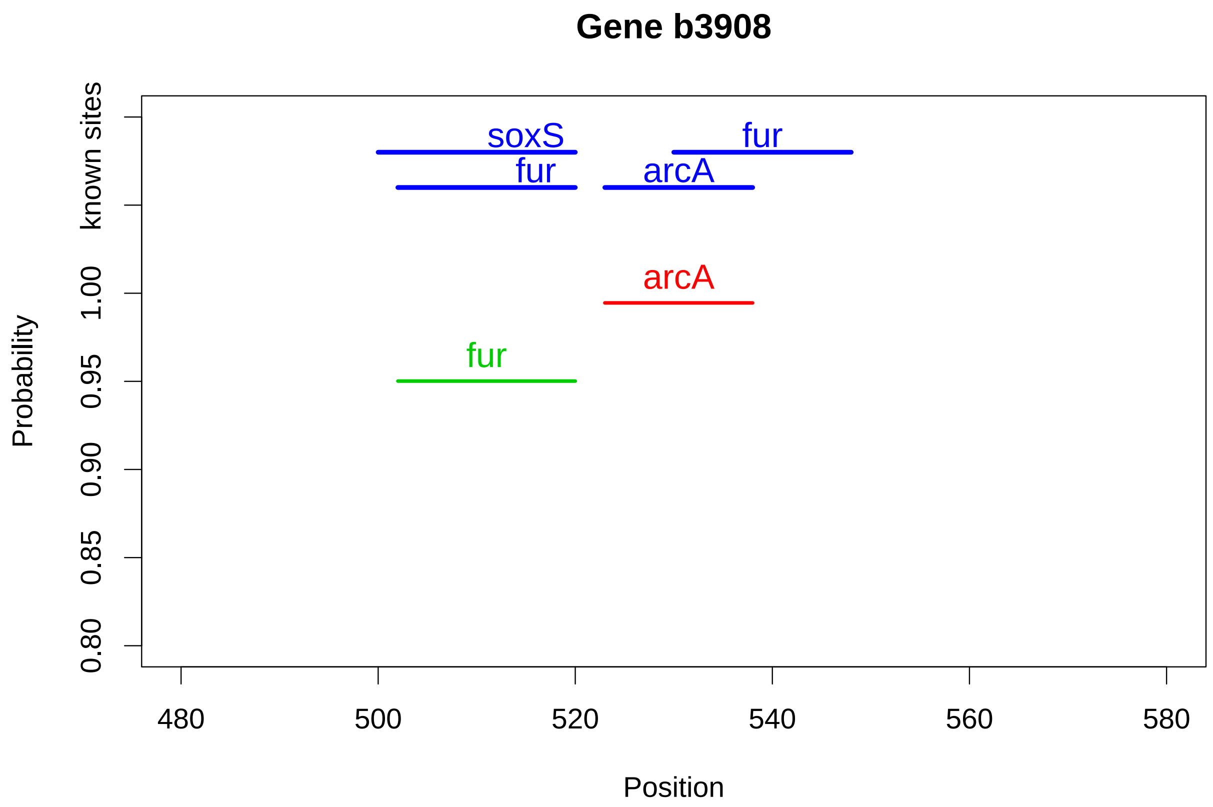
Globally, we recover 340 motifs, miss 90 and impute an additional 322 motifs. If we set the threshold of recovery to prob $> \mathbf{0.2}$, we get 373 recovered motifs. The proportion of recovered motifs goes from .8 to .7 if instead of considering only these 233 sequences we extend it to the up-stream sequences of all the genes in E. Coli that are predicted to be at the beginning of an operon (3200)

Word	recovered sites	missed sites	imputed sites
araC	6	0	6
arcA	8	5	28
argR	15	2	24
cpxR	11	1	29
creB	8	0	9
crp	36	13	131
cspA	4	0	4
cytR	2	3	7
dnaA	7	1	41
fadR	7	0	8
fis	8	7	36
fliA	12	0	14
fnr	12	0	14
fruR	12	0	18
fur	8	1	18
galR	7	0	10
gcvA	4	0	4
glpR	7	6	20
hipB	2	2	2
lexA	19	0	24
malT	4	6	6

Word	recovered sites	missed sites	imputed sites
metJ	6	3	8
metR	5	3	10
nagC	6	0	9
narL	7	3	9
narP	8	0	4
ntrC	4	1	4
ompR	5	4	28
oxyR	4	0	4
phoB	10	2	12
purR	21	1	25
rpoH2	6	1	6
rpoH3	8	0	8
rpoN	6	1	11
rpoS17	5	10	9
rpoS18	4	3	8
soxS	11	6	22
torR	3	1	5
trpR	4	0	4
tus	5	0	5
tyrR	13	4	19
Total	340	90	663

The most serious problem is overlap.

```
             Overlap
              No Yes
  Detected   361  43
  Missed      17   9
```



In blue, known binding sites: there are 4 in this region, in overlapping pairs. In green and red reconstructed ones: for each overlapping pair, we recognize only one.

Test 2: comparing with microarray experiments

If we look at all the E. Coli genes, we do not have available a detailed list of which binding sites are present.

Results from gene expression array experiments should offer a way to prove/disprove results empirically.

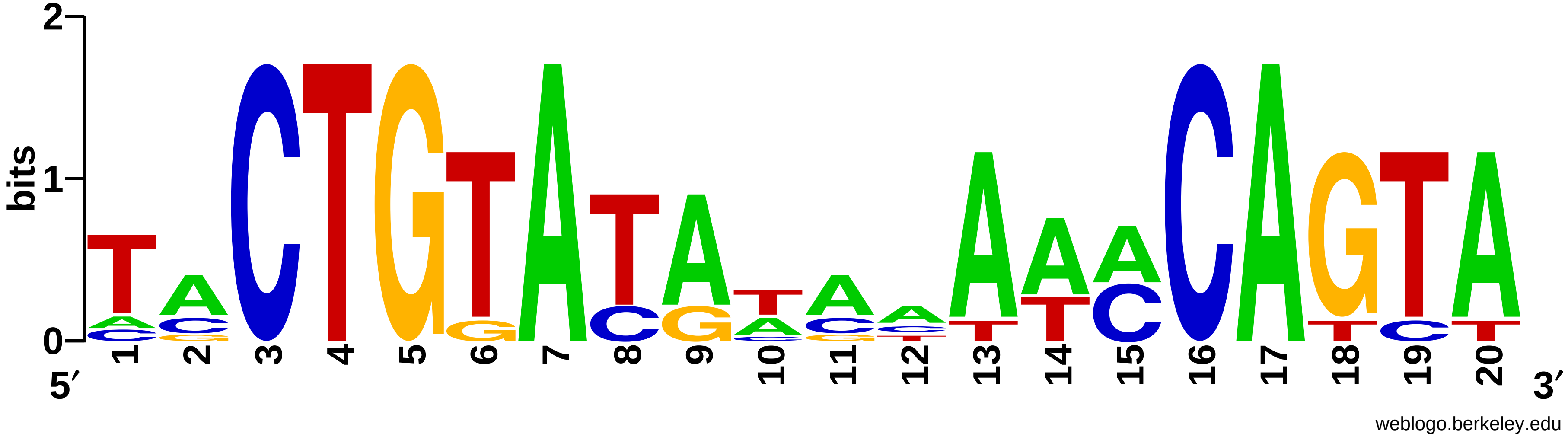
However, even for cases where the binding sites are experimentally verified, people are not very successful in correlating them with array results.

One avenue that appears succesful is the reconstruction of which regulatory proteins are activated on the base of expression array data and binding site information.

An experiment with UV

Courcelle (2002) conducted a series of microarray experiments, exposing E. Coli to UV. This is known to affect the genes regulated by LexA.

Here is a pictorial representation of the LexA binding site as reconstructed in our algorithm.



Notice that it is palindromic. Generally identified on both strands.

Explaining changes in expression with binding sites

We took the change in expression on the log scale for all E. Coli genes in one of such experiments and regressed it against the expected number of binding sites in the upstream region of each gene, for all of the protein in our dictionary.

The explanatory variable that results by far most significant is lexA.

```
              Estimate Std. Error  t value Pr(>|t|)
(Intercept) -2.021e-02  7.223e-03   -2.798  0.00518 **
malT         2.201e-01  3.320e-01    0.663  0.50737
:

cspA         5.521e-02  1.176e-01    0.469  0.63889
lexA         3.841e-01  4.572e-02    8.401  < 2e-16 ***
fnr          3.284e-02  7.832e-02    0.419  0.67501
:
hipB         4.159e-01  2.494e-01    1.668  0.09552 .
fis          1.428e-01  3.312e-02    4.313 1.67e-05 ***
oxyR         4.259e-02  2.878e-01    0.148  0.88236
:
```

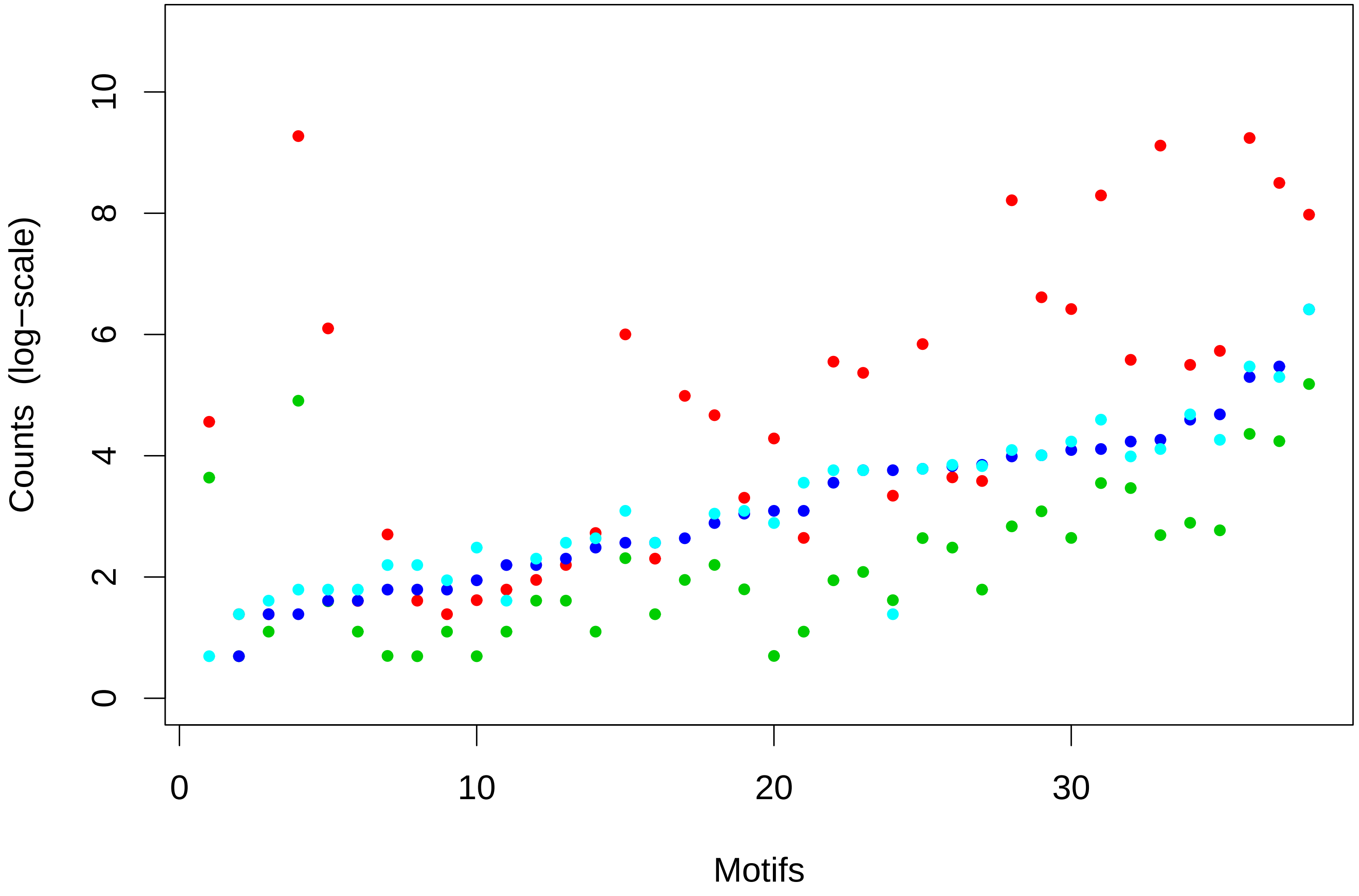
Comparing with other imputations

● Robison’s predictions

● Robison’s over stringent cutoff

● # sites with probability $> \mathbf{0.5}$

● Expected number of sites



Current work

- Analysis of multiple related genomes to confirm predictions
- Constructing a grammar for this language: how to model co-regulation
- Using our predictions in the Network Component Analysis model.

Aknowledgments

- Kennet Lange (Sabatti and Lange 2002, Proceedings IEEE)
- Lars Rohlin and James Liao (UCLA Chemical Engineering)

See you at IPAM for the proteomics and genomics meeting!

Institute for Pure & Applied Mathematics

IPAM

University of California, Los Angeles