ABSTRACT

Title of Thesis:	A COMBINATORIAL DESIGN OF A PROTEIN-BINDING DNA MICROARRAY
Degree candidate:	Aaron M. Qureshi
Degree and year:	Master of Science, 2004
Thesis directed by:	Dr. Brian R. Hunt Department of Mathematics

The biological process of *transcription* creates from a template DNA strand (i.e., the gene) copies of short-lived mRNA. The amount of mRNA produced determines the gene's *expression* in the cell, which affects the activity of the gene at a given time. *Transcription factors* are proteins which bind to the DNA in the neighborhood of the gene in order to regulate the location and rate of transcription. An important biological question is therefore to find binding locations and binding strengths for transcription factors.

This has traditionally been a laborious experimental process, but a new technology called a protein-binding microarray allows us to assay the binding affinities of a given transcription factor for many different DNA sequences in parallel. This thesis addresses a suitable combinatorial design for these microarrays that is both effective and economical.

A COMBINATORIAL DESIGN OF A PROTEIN-BINDING DNA MICROARRAY

by

Aaron M. Qureshi

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2004

Advisory Committee:

Dr. Brian R. Hunt, Chairman/Advisor
Dr. Martha L. Bulyk
Dr. James A. Yorke

ACKNOWLEDGEMENTS

I would like to thank Dr. Martha L. Bulyk of Harvard Medical School for kindly hosting me in her lab in the spring of 2004. Dr. Brian Hunt diligently supervised my thesis and made a great number of helpful suggestions. Finally, my deepest gratitude to Anthony Philippakis and Eirene Kontopoulos, without whose patience and love I would be both thesis–less and hungry.

TABLE OF CONTENTS

List of Tables **iv**

List of Figures **v**

1 Introduction **1**

2 Description of the Problem **4**

3 Verifying the Hamming Ball Property **7**

4 Designing the Array **13**

5 Recoverability **21**

6 Discernibility **26**

7 Conclusion **29**

LIST OF TABLES

3.1 Hamming ball sizes for various choices of k and r. On the horizontal axis is the palindromicity of the center of the ball. 9

3.2 A summary of the Hamming ball data. 12

5.1 The expected number of balls that survive the sieving process which do not contain the motif S. (By design, the correct ball also survives.) Here we are assuming $|\Gamma_S| = 4 \cdot |S|$. 24

6.1 Average percentage of discernible elements for balls $B(\tilde{v}, r)$. . . . 27

LIST OF FIGURES

1.1 DNA transcription. The RNA polymerase, moving to the right, is transcribing a segment of DNA (a gene) and outputting an RNA strand. Here, the transcription factors have bound to the DNA upstream of the start of transcription. 2

4.1 The De Bruijn graph with $A = \{\mathtt{A}, \mathtt{B}\}$ and $k = 2$. 14

4.2 A shift register of degree 4 over $\mathbf{F}_2$. 15

Chapter 1

Introduction

Many fundamental biological processes are governed by the action of *transcription*, a series of events that occur at the level of DNA. To transcribe a strand of DNA, an RNA polymerase attaches itself to the DNA, reads the sequence of nucleotides for a certain length, and then produces a corresponding strand of mRNA for use elsewhere (see Figure 1.1). The rate and location at which this transcription is performed is regulated by one or more proteins known as *transcription factors*, which bind themselves to the DNA in the neighborhood of the section to be transcribed.

The exact correspondence between these transcription factors and the rate of RNA synthesis is currently poorly understood. We have little information on the binding affinity of these various proteins to the DNA. We do not know in general where proteins bind on the DNA to affect the transcription process. We do not know which proteins can potentially affect regulation in general, nor which affect regulation in a particular case. Experiments which answer these questions can be performed, but it is laborious to test the many possibilities.

Bulyk et al. [1] have proposed a technology with which we can answer the first of these questions. The technique allows us to assay the binding affinities (i.e., the binding strengths) of a given transcription factor for many different DNA strands in parallel. A wafer called a *protein-binding microarray* (PBM) can be spotted with many different DNA strands, on sites spaced out across the wafer.

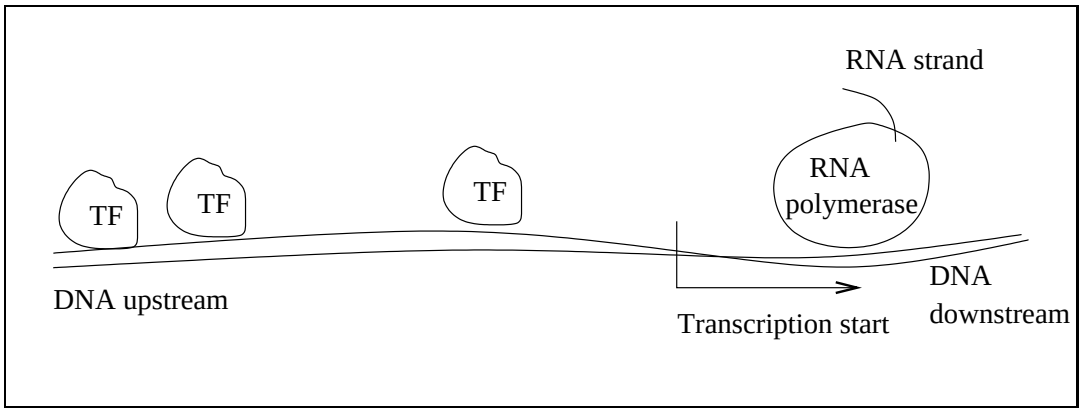

Figure 1.1: DNA transcription. The RNA polymerase, moving to the right, is transcribing a segment of DNA (a gene) and outputting an RNA strand. Here, the transcription factors have bound to the DNA upstream of the start of transcription.

The microarray is then exposed to our (known) transcription factor of interest. The protein binds to DNA sequences for which it has an affinity. The wafer is then probed with a laser, causing fluorescently-labelled antibodies attached to the protein to fluoresce. The binding affinity of the transcription factor for each particular DNA sequence can then be assessed by the intensity of the fluorescence at that position in the microarray.

This paper concerns the combinatorial design of PBMs. How can we order the DNA strands on the PBM so as to to test for as many potential binding sites as possible? Ideally, for a given length k of a transcription factor's DNA binding site sequence, the spots on the PBM would contain all of the 4^k possible such sites, one per spot. After performing the experiment, we could simply read off the binding affinities from each spot. Unfortunately, it is not economically

feasible to spot a PBM with all such sequences when $k \geq 8$. Instead, we must meld many different DNA sequences onto each spot without losing too much of the experiment's informative power.

A complete description of the problem is given in Chapter 2. In Chapter 3 we justify some of the assumptions from Chapter 2 and demonstrate that the chosen parameters are reasonable. The design method is explained in Chapter 4. In Chapters 5 and 6, we address concerns about the power of this design.

Chapter 2

Description of the Problem

A PBM is a glass slide on which can be printed a number of distinct *spots*, each of which contains a copy (actually, many copies) of one particular DNA sequence. Each spot must have the same length l, and for our problem, we were given $l = 44$. Our rectangular wafer can contain $103 \times 213 = 21939$ such spots. We must leave a small number of these (approximately 1000) as control spots for experimental purposes.

One unknown parameter of the problem is the length k of the sites to which the transcription factor will bind. Though k is not known a priori, experimental evidence on a variety of transcription factors suggests that $5 \leq k \leq 15$. For our design, we assumed $k \leq 9$; in Chapter 3 we will show that this is a reasonable choice for many transcription factors.

Although different transcription factors bind with different affinities, we do not know if the PBM experiment will be sensitive enough for us to distinguish between different affinity magnitudes. We therefore adopt the simpler convention that a transcription factor either binds or does not bind to a spot, with some appropriate threshold between the two determined by the laboratory technician.

Let S be the set of binding sites, known as a *motif*. It is generally believed that each transcription factor has its own unique motif. Each element of a particular motif is of the same length k. We do not know a priori how many elements the motif contains. Again, we will give evidence in Chapter 3 that for many

transcription factors, $10 \leq |S| \leq 40$. One important aspect of S, however, is that its elements are "close" to each other, in the following sense.

Consider the set of letters {A,C,G,T}, the nucleotides from which DNA is constructed. Map this set in a natural way to $\mathbf{F}_4 \cong \mathbf{F}_2 \times \mathbf{F}_2$, the finite field of four elements, identifying {A,C,G,T} with $\{00, 01, 10, 11\}$, respectively. This mapping is somewhat arbitrary, although it is slightly more computationally convenient to identify A and T with numbers which are binary complements of each other, and similarly for C and G. We will not discuss the addition and multiplication operations on $\mathbf{F}_4$, as they do not have biological relevance for our purposes.

Our sequence space of k–long DNA strands (or *k-mers*) can then be identified with $\mathbf{F}_4^k$. A natural metric for this space is the Hamming metric, which counts the number of mismatches between two k-mers:

$$d : \mathbf{F}_4^k \times \mathbf{F}_4^k \longrightarrow \mathbf{N}$$

$$d(w, v) = |\{1 \leq i \leq k : w_i \neq v_i\}|,$$

where $0 \leq d \leq k$.

However, because of the double-strandedness of DNA, a protein which binds to a DNA strand $v \in \mathbf{F}_4^k$ also binds to its reverse complement, $\overline{v}$.

Definition Let $v = (v_1, \ldots, v_k)$ be a k-mer. Then the *reverse complement* of v is $\overline{v} = (\overline{v_k}, \ldots, \overline{v_1})$, where $\overline{\texttt{A}} = \texttt{T}$, $\overline{\texttt{C}} = \texttt{G}$, $\overline{\texttt{G}} = \texttt{C}$, and $\overline{\texttt{T}} = \texttt{A}$.

For instance, if $k = 6$, then a protein that binds to $v = \texttt{AAGTCA}$ also binds to its reverse complement, $\overline{v} = \texttt{TGACTT}$. Thus, in our space, we wish to identify v and its reverse complement. Define a relation $\sim$ on the set $\mathbf{F}_4^k$ such that for

$w, v \in \mathbf{F}_4^k$, $w \sim v$ iff $w \in \{v, \overline{v}\}$. It is easy to check that $\sim$ is an equivalence relation. Write the set of equivalence classes as $\widetilde{\mathbf{F}}_4^k$, and for each class choose its lexicographically lesser element as its representative element. We define a modified Hamming metric for this set:

$$\tilde{d} : \widetilde{\mathbf{F}}_4^k \times \widetilde{\mathbf{F}}_4^k \longrightarrow \mathbf{N}$$

$$\tilde{d}(\tilde{w}, \tilde{v}) = \min(d(w, v), d(w, \overline{v})).$$

Proposition 2.1 *The function $\tilde{d}$ defines a metric on $\widetilde{\mathbf{F}}_4^k$.*

Proof It is immediate that $\tilde{d}$ is symmetric and positive definite, since these properties follow from d. We must show that the triangle inequality holds. For $\tilde{u}, \tilde{v}, \tilde{w} \in \widetilde{\mathbf{F}}_4^k$,

$$\begin{aligned}
\tilde{d}(\tilde{w}, \tilde{v}) &= \min(d(w, v), d(w, \overline{v})) \\
&= \min(d(w, v), d(w, v), d(w, \overline{v}), d(w, \overline{v})) \\
&\leq \min(d(w, u) + d(u, v), d(w, \overline{u}) + d(\overline{u}, v), d(w, u) + d(u, \overline{v}), d(w, \overline{u}) + d(\overline{u}, \overline{v})) \\
&= \min(d(w, u) + d(u, v), d(w, \overline{u}) + d(u, \overline{v}), d(w, u) + d(u, \overline{v}), d(w, \overline{u}) + d(u, v)) \\
&= \min(d(w, u), d(w, \overline{u})) + \min(d(u, v), d(u, \overline{v})) \\
&= \tilde{d}(\tilde{w}, \tilde{u}) + \tilde{d}(\tilde{u}, \tilde{v}). \quad \blacksquare
\end{aligned}$$

Experimental evidence suggests that our set S of binding sites are all clustered in relatively near proximity in sequence space. We start with the assumption that S is contained in some ball in $\widetilde{\mathbf{F}}_4^k$ of small radius (relative to k). Since this assumption is of importance in our design, some justification is presented in the next chapter.

Chapter 3

Verifying the Hamming Ball Property

A general consensus among computational biologists is that elements of a motif are "close" in sequence space, since a motif's elements tend to look similar. However, we are not familiar with any study which has explicitly confirmed this assumption. We want first to define this notion of closeness and analyze existing data sets for verification of this property.

Definition The *Hamming ball* with center $\tilde{v} \in \widetilde{\mathbf{F}}_4^k$ and radius $r \in \mathbf{N}$ is defined to be $B(\tilde{v}, r) = \{\tilde{w} \in \widetilde{\mathbf{F}}_4^k : \tilde{d}(\tilde{w}, \tilde{v}) \leq r\}$.

Note that in our definition we are using the term "Hamming" somewhat loosely: these are balls under the modified Hamming metric $\tilde{d}$ rather than the normal Hamming metric d. We expect our motif to be contained in some ball of small radius under $\tilde{d}$. The first question to address is how many elements a Hamming ball in our space $\widetilde{\mathbf{F}}_4^k$ contains, for not all such balls are of the same size. Recall that each element $\tilde{w} \in \widetilde{\mathbf{F}}_4^k$ is an equivalence class $\{w, \overline{w}\}$ of elements from $\mathbf{F}_4^k$. The number of elements of $\widetilde{\mathbf{F}}_4^k$ in $B(\tilde{v}, r)$ is thus at most the number of elements of $\mathbf{F}_4^k$ in $B(v, r)$. Since there are $3^i\binom{k}{i}$ elements with distance i from v, the largest a ball can be is $\sum_{i=0}^{r} 3^i\binom{k}{i}$. However, if our center is a palindrome, this sum double-counts many elements and our ball is of considerably smaller size.

Definition Let $v \in \mathbf{F}_4^k$. Then v is a *palindrome* if $v = \overline{v}$. More generally, v is

p–palindromic if

$$p = \sum_{i=1}^{\lfloor k/2 \rfloor} \delta(v_i = \overline{v}_{k+1-i}),$$

where $\delta(\cdot)$ is the Kronecker δ-function. If $p = \frac{k}{2}$, then we say p is *(completely) palindromic*.

In fact, for a given k and r, the sizes of various Hamming balls vary according to the palindromicity of their centers. The intuition here is that we are counting the number of elements of a ball $B \subset \mathbf{F}_4^k$ and of its complement $\overline{B}$, then halving the total due to our equivalence class relation corresponding to $\widetilde{\mathbf{F}}_4^k$. But as the palindromicity of the center of B increases, the set $B \cap \overline{B}$ increases in size, and we are wrongly double-counting these elements in $B \cap \overline{B}$ which are not themselves palindromes (i.e., whose equivalence classes have a size of 1). More formally,

$$|B(\tilde{v}, r)| = \sum_{i=0}^{r} 3^i \binom{k}{i} - \frac{|\{v \in B(v,r) : \overline{v} \in B(v,r)\}| - |\{v \in B(v,r) : v = \overline{v}\}|}{2}.$$

Example We seek the order of $B(\tilde{v}, 2) \subset \widetilde{\mathbf{F}}_4^k$, for $v = \texttt{AAAATT}$. The palindromicity of v is 2. We must find all the non-palindromic $w \in B(v, 2)$ such that $\overline{w} \in B(v, 2)$. We can change the center two (non-palindromic) $\texttt{A}$s to any letter, since $\overline{\texttt{AAMNTT}} = \texttt{AA}\overline{\texttt{N}}\overline{\texttt{M}}\texttt{TT} \in B(v, 2)$ for any $\texttt{M}, \texttt{N} \in \{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}\}$. This gives 4^2 elements, but 4 of these yield complete palindromes. Also, we can change one of the middle two letters to a $\texttt{T}$, and one of the other letters to any different letter. This gives us an additional $\binom{2}{1}\binom{4}{1} \cdot 3$ elements, of which none are palindromes. The formula above yields

$$|B(\tilde{v}, 2)| = \sum_{i=0}^{2} 3^i \binom{6}{i} - \frac{1}{2}\left(4^2 - 4 + \binom{2}{1}\binom{4}{1} \cdot 3\right) = 136. \quad \blacksquare$$

(k, r)	p = 0	p = 1	p = 2	p = 3	p = 4
(4, 1)	13	13	7		
(4, 2)	66	55	37		
(5, 1)	16	16	14		
(5, 2)	106	97	80		
(6, 1)	19	19	19	10	
(6, 2)	154	153	136	82	
(7, 1)	22	22	22	20	
(7, 2)	211	211	202	173	
(8, 1)	25	25	25	25	13
(8, 2)	277	277	276	253	145
(8, 3)	1789	1783	1704	1513	901
(9, 1)	28	28	28	28	26
(9, 2)	352	352	352	343	302
(9, 3)	2620	2620	2580	2426	2066

Table 3.1: Hamming ball sizes for various choices of k and r. On the horizontal axis is the palindromicity of the center of the ball.

Similar calculations as in this example yield Table 3.1.

Calculating the number of balls of each of these sizes is equivalent to calculating the number of elements of $\mathbf{F}_4^k$ of a given palindromicity. Fortunately this does not prove to be difficult. If k is even, for a given palindromicity p, we are free to choose $k-p$ letters but have restrictions on the remaining p letters. If k is even, for $0 \leq p \leq \frac{k}{2}$,

$$|\{v \in \mathbf{F}_4^k : v \text{ has palindromicity } p\}| = 4^{\frac{k}{2}} \binom{\frac{k}{2}}{p} 3^{\frac{k}{2}-p}.$$

If k is odd, the formula is nearly the same:

$$|\{v \in \mathbf{F}_4^k : v \text{ has palindromicity } p\}| = 4^{\frac{k+1}{2}} \binom{\frac{k-1}{2}}{p} 3^{\frac{k-1}{2}-p}.$$

Since we know the size of each type of Hamming ball, and the number of balls of that type, given r, we can calculate an expected ball size $E[|B(\tilde{v}, r)|]$ for a randomly chosen center $\tilde{v} \in \widetilde{\mathbf{F}}_4^k$. We will need this in Chapter 5.

Our assumption is that the set of binding sites $S \subseteq B(\tilde{v}, r)$ for some v and some $r \leq 3$. We also assume $10 \leq |S| \leq 40$, so for $k \geq 5$ the proportion of elements of the Hamming ball that are binding sites may be quite small.

To justify these assumptions, we use the well known TRANSFAC [3] and JASPAR [6] databases, which contain many transcription factors' DNA binding site motifs. TRANSFAC contains 111 data sets suitable for our analysis, and JASPAR contains 76. These databases record not just the sequence of a binding site, but also a (variable) number of surrounding nucleotides. Therefore, to prepare our data, we first need to align the sequences and remove the extraneous nucleotides.

We use the motif-finding program AlignACE [5] to align the sequences into sets of possible motifs, varying the required number of conserved (i.e., constant) columns from 5 to 14. Then, for each motif, for each column j we calculate the monographic distribution $P_j = (p_{j\mathtt{A}}, p_{j\mathtt{C}}, p_{j\mathtt{G}}, p_{j\mathtt{T}})$ on the four letters. Reasoning that in the case of a spurious motif generated by AlignACE, this distribution will not differ significantly from the generic distribution $Q = (0.28, 0.22, 0.22, 0.28)$ found across all genome data, we calculate the relative entropy between P^j and Q. However, in order not to bias this information in favor of motifs with more columns, for each j we need to subtract off the mean information μ of a random distribution P'. Thus our score is

$$I = \sum_j \sum_{i \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} (p_{ji} \log(p_{ji}/q_i) - \mu) \,.$$

We take as our putative S that motif with the highest I.

Out of the 187 data sets, 42 have motifs whose best scoring length is $k \leq 9$. For each of these, we then proceed to calculate the smallest Hamming ball that contains all the elements of the motif. A summary of this experiment is in Table 3.2. From the table, we see of the 42 motifs with $k \leq 9$, all but 5 had $r \leq 3$. Thus, this restriction on r seems reasonable. The assumption that $10 \leq |S|$ is only somewhat supported by the data. However, the data sets are not necessarily comprehensive, so at least for larger k and r, this lower bound seems roughly correct.

TF Identifier	k	r	$\lvert S\rvert$
JASPAR			
Broad_4	8	2	4
c-myb_all_sites	8	3	22
c-myb_single_sites	8	3	16
eN-1	8	3	5
gfi1	8	2	4
Snail	8	2	9
arh-arnt	9	3	21
arnt_homo	9	3	13
caat-box	9	3	34
dorsal_nogaps	9	2	6
e4bp4	9	1	3
Hox15	9	4	6
Nkx	9	2	5
TRANSFAC			
I-UBX-01	5	0	1
V-AREB6-01	5	0	1
V-HOXA3-01	6	1	4
V-NCX-01	6	2	5
V-SPZ1-01	6	3	13
I-SN-01	7	1	6
P-Alfin1-Q2	7	1	2
V-AHRARNT-01	7	2	7
V-CIZ-01	7	2	6
V-MRF2-01	7	2	7
V-TBP-01	7	2	9
F-ADR1-01	7	3	14
P-GAMYB-01	7	3	14
V-ZIC2-01	7	3	11
V-ZIC3-01	7	3	20
V-RREB1-01	7	4	13
V-ATF6-01	8	1	2
P-DOF3-01	8	2	5
V-ERR1-Q2	8	2	3
V-NKX25-01	8	2	7
V-NKX3A-01	8	2	7
V-SRY-01	8	3	9
V-ZF5-B	8	4	15
V-HFH1-01	9	2	9
V-ARNT-01	9	3	13
V-E2F1DP2-01	9	3	12
V-FOXJ2-01	9	3	18
V-MSX1-01	9	4	11
V-ZIC1-01	9	4	18

Table 3.2: A summary of the Hamming ball data.

Chapter 4

Designing the Array

We turn now to the design of the array, i.e., the makeup of the 20,000 spots on the array. If we had no restriction on the number of the spots, we would simply put each of the 4^k k–mers on a spot by itself. After running the experiment, the set of binding sites S would be the set of the "lit" spots. But for $k \geq 8$, this proves to be impossible given the size of the array.

We instead must put many k–mers on each spot. For the moment, consider $k = 9$, and note that a spot of 44 consecutive nucleotides contains 36 overlapping 9–mers. If we could pack 36 distinct 9–mers onto each spot, never repeating a 9–mer, we would need only about $\frac{4^9}{36}$ spots rather than 4^9. Of course, recovering S will not be so trivial: a lit spot could indicate transcription factor affinity for any of the 36 9–mers on the spot, or even for more than one. We will address this issue later.

For now, we would like to place every k-mer onto the array an equal number of times in these "packed" spots. Fortunately, the mathematical object known as a *De Bruijn sequence* can help us here.

Definition Let A be a finite alphabet, and fix a $k \in \mathbf{N}$. A *De Bruijn graph* is a graph whose vertices are k–long words over A, with directed edge from word $a = (a_1, a_2, \ldots, a_k)$ to word $b = (b_1, b_2, \ldots, b_k)$ whenever $a_2 = b_1, a_3 = b_2, \ldots, a_k = b_{k-1}$. A *De Bruijn sequence* is a circuit along the edges of a De Bruijn graph which traverses each vertex exactly once.

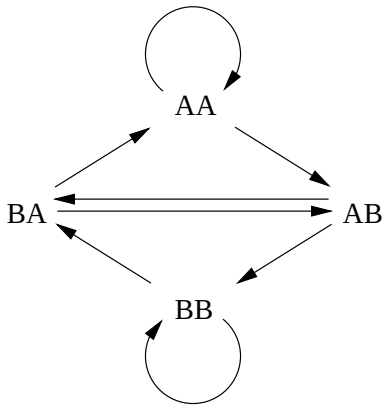

Figure 4.1: The De Bruijn graph with $A = \{\mathtt{A}, \mathtt{B}\}$ and $k = 2$.

Figure 4.1 gives a simple example of a De Bruijn graph. A De Bruijn sequence here would be `AA`, `AB`, `BB`, `BA`. More compactly, we can write the sequence as `AABB`, understanding that we must "wrap around" at the end.

Our approach is first to find a De Bruijn sequence on the De Bruijn graph whose nodes are in $\mathbf{F}_4^k$. If the length of a spot is l, we can subdivide this sequence into l–long subsequences. Note that we must repeat the last $k-1$ letters on a spot when we begin a new spot.

Example Let $A = \{\mathtt{0}, \mathtt{1}, \mathtt{2}, \mathtt{3}\}$, and let $k = 2$, so that we are interested in 2–long words over A. A De Bruijn sequence over the graph is `0021223301031132`. All 2–long words are represented exactly once in this sequence. If $l = 5$, we can define our spots as:

$$\{\mathtt{00212},\ \mathtt{22330},\ \mathtt{01031},\ \mathtt{11320}\}$$

Note we start a spot by repeating the last character from the previous spot. ∎

It is natural to ask how many De Bruijn sequences exist for a given k (if indeed there are any at all). This question is answered in [4]:

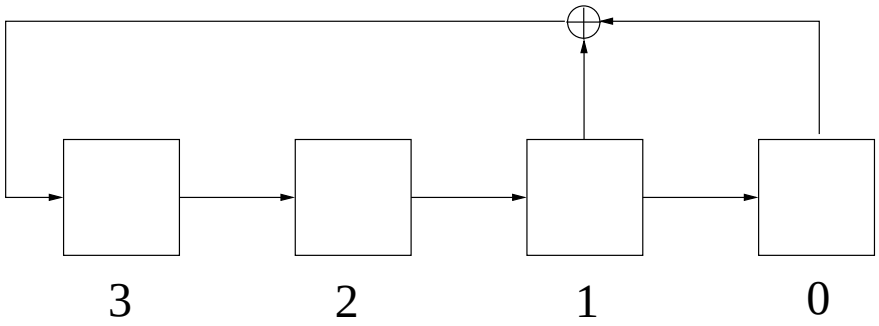

Figure 4.2: A shift register of degree 4 over $\mathbf{F}_2$.

Theorem 4.1 *The number of De Bruijn sequences on words of length k over an alphabet of size a is $(a!)^{a^{k-1}}/a^k$.*

Thus we are guaranteed for $5 \leq k \leq 9$ that in fact a great number of De Bruijn sequences exist. There are a number of algorithms to generate De Bruijn sequences. We chose one algorithm which has been well studied and is known to exhibit desirable randomness properties. It is based on a construct from algebraic coding theory known as a linear shift register. An extensive theory has been developed behind shift registers, and a theoretical treatment is given in [2]. A diagram of a shift register is given in Figure 4.2.

For now, let us consider De Bruijn sequences over $\mathbf{F}_2$. A shift register is associated with a polynomial $f(x) \in \mathbf{F}_2[x]$ by constructing a series of delay boxes of length $\deg(f)$ and letting the tap (i.e., output line) on the jth box represent the coefficient of the term x^j for $0 \leq j < \deg(f)$. For instance, in Figure 4.2, the associated polynomial is $f(x) = x^4 + x + 1$. The coefficients of f are 1 for $j = 0, 1$, and 0 for $j = 2, 3$. Thus, there are taps on boxes 0 and 1, and no taps on

boxes 2 and 3. To generate a sequence, the boxes are filled with elements in $\mathbf{F}_2$, then the register is stepped. To step the register, the sum (in $\mathbf{F}_2$) of the tapped elements is computed, then all the other elements are shifted to the right by one. The rightmost element "falls off" the register and is discarded. The computed sum is entered in the leftmost box and is also recorded as the next element in the sequence. This can be repeated as many times as necessary to generate a sequence of the desired length.

It is clear that if the register contains the same elements at time t' that it did at step t, the output sequence will start to repeat. We might wonder, for different arrangements of taps, how long the output sequence will go before it repeats. Clearly, no non-repeating sequence can be longer than the length of the De Bruijn sequence, 2^k. One consequence of the theory behind shift registers is found in [2] :

Theorem 4.2 *A shift register of degree k generates a non-repeating sequence of length $2^k - 1$ if and only if its associated polynomial $f(x) \in \mathbf{F}_2[x]$ is primitive of degree k.*

Recall that primitive polynomials are polynomials whose every root generates the entire field $\mathbf{F}[x]/(p(x)) \cong \mathbf{F}_2^{\deg p(x)}$. For each value of k in our range of interest, there are numerous known primitive polynomials of degree $2k$, a fact that we shall use shortly.

Thus we can associate any degree k primitive polynomial in $\mathbf{F}_2[x]$ with a sequence of length $2^k - 1$. We might call this a "nearly De Bruijn" sequence. The missing element is the k–long $\mathbf{0}$ vector. To make our sequence truly a De Bruijn

sequence, we need only to insert a single 0 into the sequence immediately before the $(k-1)$–long run of zeroes.

Example We use the register described in Figure 4.2 to generate the table below. Note that the polynomial $x^4 + x + 1$ is a primitive polynomial over $\mathbf{F}_2$.

Time	Contents	Recorded	Sequence
t_0	1111	—	—
t_1	0111	0	0
t_2	0011	0	00
t_3	0001	0	000
t_4	1000	1	0001
t_5	0100	0	00010
t_6	0010	0	000100
t_7	1001	1	0001001
t_8	1100	1	00010011
t_9	0110	0	000100110
t_{10}	1011	1	0001001101
t_{11}	0101	0	00010011010
t_{12}	1010	1	000100110101
t_{13}	1101	1	0001001101011
t_{14}	1110	1	00010011010111
t_{15}	1111	1	000100110101111

To the 15–long output sequence 000100110101111 we prepend a 0 before the 3–long run of 0s to get 0000100110101111. We can see that this sequence contains all 4–long elements exactly once, wrapping around at the end. Note that the

initial content of the register was chosen arbitrarily; any fill except the all zeroes fill would generate such a sequence. ∎

For our purposes, however, we need not a 2^k–long sequence over $\mathbf{F}_2$, but rather a 4^k–long sequence over $\mathbf{F}_4$. One way to generate such a sequence is to generalize the idea of the shift register from $\mathbf{F}_2$ to an arbitrary finite field. However, since 4 is a power of 2, it is easier to use a trick to generate a De Bruijn sequence over $\mathbf{F}_4$ using a polynomial in $\mathbf{F}_2[x]$.

Proposition 4.3 *Let $p(x) \in \mathbf{F}_2[x]$ be a primitive polynomial of degree $2k$. Double–step a shift register driven by $p(x)$, and map the 2–long recorded output under any bijective mapping $M : \mathbf{F}_2 \times \mathbf{F}_2 \longrightarrow \mathbf{F}_4$. Then double-stepping the register $2^{2k} - 1$ times generates a $(4^k - 1)$–long sequence in $\mathbf{F}_4$ under M, with no k–long repeats in $\mathbf{F}_4$.*

Proof Since we have double–stepped the register $2^{2k} - 1$ times, and M gives us one element of $\mathbf{F}_4$ at each double–step, clearly the length of the sequence is $4^k - 1$. We must show that the sequence has no k–long $\mathbf{F}_4$ repeats.

Our $\mathbf{F}_2$ sequence has length $2 \cdot (2^k - 1)$; it is simply two complete concatenated cycles of the nearly De Bruijn sequence for $2k$–long words over $\mathbf{F}_2$, using Theorem 4.2 and the fact that p is primitive. Assume that a k–long repeat over $\mathbf{F}_4$ occurs. This means that a $2k$–long repeat over $\mathbf{F}_2$ occurred, since M is one–to–one. Since the $\mathbf{F}_2$ cycles have period $2^{2k} - 1$, the distance between the two occurrences of this repeat must be $2^{2k} - 1$ elements of $\mathbf{F}_2$ apart. Therefore, the first occurrence is contained wholly within the first $2^{2k} - 1$ elements, and the second occurrence is wholly within the second $2^{2k} - 1$ elements. Assume the first

occurrence starts at index i; then the second begins at index $2^{2k}-1+i$. Since the k–long $\mathbf{F}_4$ sequences start at these elements, we must have that i and $2^{2k}-1+i$ have the same parity. This is a contradiction; therefore no such $\mathbf{F}_4$ repeat can occur. ∎

Note again that the element $\mathtt{0} \in \mathbf{F}_4$ is again missing from our sequence, so we need to insert it at an appropriate place. Using this algorithm we can easily generate a great number of De Bruijn sequences over $\mathbf{F}_4^k$ by picking any primitive polynomial in $\mathbf{F}_2[x]$ of degree $2k$, stepping its shift register $2 \cdot (2^{2k}-1)$ times and applying M to the output.

Example To continue the example from above, we concatenate two copies of the nearly De Bruijn sequence and read off the numbers in pairs. We use the bijective map from $\mathbf{F}_2 \times \mathbf{F}_2$ to $\mathbf{F}_4$ mentioned in Chapter 2.

$$\underbrace{00}_{0}\underbrace{01}_{1}\underbrace{00}_{0}\underbrace{11}_{3}\underbrace{01}_{1}\underbrace{01}_{1}\underbrace{11}_{3}\underbrace{10}_{2}\underbrace{00}_{0}\underbrace{10}_{2}\underbrace{01}_{1}\underbrace{10}_{2}\underbrace{10}_{2}\underbrace{11}_{3}\underbrace{11}_{3}$$

Prepending a `0` yields the sequence `0010311320212233`. Note that this sequence contains every 2–long sequence over $\mathbf{F}_4$ exactly once. ∎

We thus have a method for generating De Bruijn sequences for any given k. Due to the size restrictions on our microarray, the largest value we could choose was $k = 9$, and we were given a spot length of 44. The first $k-1 = 8$ letters of any spot are repeated from the last 8 letters of the previous spot. The number of spots needed is $\lceil 4^9/(44-8) \rceil = 7282$ for this register. (We simply continue generating elements with the shift register to fill out the last spot.)

Note that we have not yet accounted for the reverse complement bindings; we have been working in $\mathbf{F}_4^k$ rather than $\widetilde{\mathbf{F}}_4^k$. The transcription factor will bind at two places on our array, where $v \in S$ occurs and where the reverse complement $\overline{v}$ occurs. So we have built some redundancy into our array: usually two spots will light up for each binding site. (Rarely, when a single spot contains both v and $\overline{v}$ or when $v = \overline{v}$, only a single spot will be lit.) This is desirable, given the vagaries of the experimental data.

Since we have about 20,000 spots to work with, we will include two De Bruijn sequences over 9-long strings, using two different generating primitive polynomials. We also include two sequences over 8-long strings, each of which take $\lceil 4^8/(44-8) \rceil = 1821$ spots. (The remaining spots will be left as experimental controls and to run side experiments with.) So in fact, a 9-long binding site will generally light up the array in 4 to 8 distinct spots.[1]

[1]If our binding site is shorter, it will light up even more spots. For instance, if the binding site is 8 characters, it could light up as many as 20 spots. If our binding site is longer, there is still a chance that our array contains at least one copy of the string. For instance, if the binding site is 10 characters, there is approximately an 85.9% chance that the array has at least one spot that contains the binding site.

Chapter 5

Recoverability

Our strategy for recovering the motif S is first to recover the Hamming ball $B(\tilde{v}, 3) \supset S$, and then to determine the elements of $B(\tilde{v}, 3)$ that comprise the elements of S. But will our proposed array allow us to do this?

For now, let us ignore the effect of reverse complements, as they tend to make computation difficult. So we will work in $\mathbf{F}_4^k$ rather than in $\widetilde{\mathbf{F}}_4^k$. In the first step, we would like to use the pattern of lit spots to determine which Hamming ball contains S. But clearly different Hamming balls have regions of overlap; moreover, since each spot has multiple k-mers, elements from disjoint Hamming balls can be contained on the same spot in the array. We would like to know what the chance is that the pattern of lit spots does not allow us to distinguish between two Hamming balls, B and B'.

Let us first establish some notation. Let $\lambda_i \subset \mathbf{F}_4^k$ be the set of k–mers on the ith spot. Represent the entire array by $\Lambda = \{\lambda_i\}_{i=1}^m$, where m is the number of spots. For a set $A \subset \mathbf{F}_4^k$, let $\Gamma_A = \{\lambda_i \in \Lambda : \lambda_i \ni a \text{ for some } a \in A\}$ be the set of spots which are lit by matches to A, and let $|A| = n$ and $|\Gamma_A| = g$. We continue to use $S \subset \mathbf{F}_4^k$ to denote the motif for our transcription factor. Thus Γ_S is the set of spots lit by our motif elements.

Given Γ_A for an arbitrary set of k–mers A, we can use a straightfoward algorithm to attempt to find a ball B which contains A (if any exist at all). For $\lambda_1 \in \Gamma_A$, mark any ball containing at least one k–mer in λ_1. Then rule out any

ball not marked in this way. Continue this sieving process on the set of all (k, r) balls for $i = 1, \ldots, |\Gamma_A|$. Denote the set of remaining balls which are never ruled out by this algorithm $\mathcal{B}(\Gamma_A)$.

Definition A ball B is *recoverable* with respect to a set $A \subset \mathbf{F}_4^k$ if $B \in \mathcal{B}(\Gamma_A)$.

If our motif S is drawn from B, note that B will always be recoverable with respect to S. It is possible that another ball B' also survives the sieving process. If $S \subset B \cap B'$, then this is unavoidable. But it is not problematic to our ultimate goal: we can claim either $S \subset B$ or $S \subset B'$ and move on to identifying S, since our choice of the two balls is inconsequential.

But what if $S \not\subset B'$? This indicates that we have lost some information due to our particular packing of the k–mers on the spots. Though we have no information allowing us to discriminate between B and B', it matters very much which one we choose. We would like for this situation to occur very rarely, if at all. More generally, we would like $S \subset \bigcap_i B_i$ for $B_i \in \mathcal{B}(\Gamma_S)$.

Let us approximate the probability that a spurious ball, i.e., one not containing S, lies in $\mathcal{B}(\Gamma_S)$. Consider the expected number of balls in $\mathcal{B}(\Gamma_A)$ for an arbitrary set A. When this expected number of survivors is small compared to 1, the probability of a spurious ball surviving the sieve should also be small.

We first approximate how many spurious balls we expect to survive the sieve for a given $\lambda \in \Gamma_A$ (i.e., for one stage of the sieve). For simplicity, for now let us consider an array which contains every k–mer exactly once.

We use Table 3.1 to determine an expected Hamming ball size $E[|B|]$ for $B \subset \mathbf{F}_4^k$. For $v \in \lambda$, we can therefore estimate $\Pr(B \ni v) \approx \frac{E[|B|]}{4^k}$. If there are l

elements in λ, assuming independence among them, we obtain

$$\Pr(B \ni v_i \text{ for some } v_i \in \lambda) \approx 1 - \left(1 - \frac{E[|B|]}{4^k}\right)^l.$$

Again, making the assumption that the probability of a ball surviving one step of the sieve is independent of it surviving the other steps, we find that the expected number of spurious balls which survive the g steps of the sieving process is

$$E[|\mathcal{B}(\Gamma_S)|] \approx 4^k(1 - (1 - \frac{E[|B|]}{4^k})^l)^g,$$

where $g = |\Gamma_A|$. We would now like to obtain an estimate for g.

Recall that $m = |\Lambda|$ is the number of spots, and $n = |A|$ is the size of our arbitrary set. For k sufficiently large, $m, n \ll 4^k$. Then we will show

$$\Pr(|\Gamma_A| = x) \approx \frac{\binom{m}{x} \sum_{n_1,\ldots,n_x} \frac{n!}{n_1! \cdots n_x!}}{m^n},$$

where $\sum_{i=1}^{x} n_i = n$ and $n_i > 0$ for all i.

To see this, let the random variable Y_i be the spot index for the k–mer $v_i \in A$, with $1 \leq Y_i \leq m$. Since the spots are all the same size, $\Pr(Y_i = j) = \frac{1}{m}$ for $1 \leq i \leq n$ and $1 \leq j \leq m$. Since $m \ll 4^k$, our spots are large, and since $n \ll 4^k$, there are not too many elements of A on a given spot. So we can say $\Pr(Y_i = j \mid Y_{i'} = j) \approx \Pr(Y_i = j)$ for $i' \neq i$, or the Y_i are independent.

Let $Z_j = \sum_{i=1}^{n} \delta(j = Y_i)$ be the number of elements of A on spot j. Since the Y_i are independent trials with the uniform distribution, the Z_j have the joint multinomial distribution:

$$\Pr(Z_1 = n_1, \ldots, Z_m = n_m) = \frac{n!}{n_1! \cdots n_m!} \left(\frac{1}{m}\right)^n,$$

(k, r)	$\lvert S\rvert = 5$	$\lvert S\rvert = 10$
(7, 2)	8.96×10^{-5}	4.89×10^{-13}
(8, 3)	7.01	7.51×10^{-4}
(9, 3)	1.01×10^{-5}	3.91×10^{-16}

Table 5.1: The expected number of balls that survive the sieving process which do not contain the motif S. (By design, the correct ball also survives.) Here we are assuming $|\Gamma_S| = 4 \cdot |S|$.

where $\sum_{j=1}^{m} n_j = n$. Finally, our random variable $|\Gamma_A| = \sum_{j=1}^{m} \delta(Z_j > 0)$. If $|\Gamma_A| = x$, we have $\binom{m}{x}$ ways of choosing our lit spots. For each such choice, we can have any partition $(n_1, \ldots, n_x)$ of n into x non-zero elements. The condition on the sum above expresses exactly this.

With these results in hand, we can now estimate how many random balls we expect to survive the sieve. In our case, we have an array which contains two De Bruijn sequences over 9-mers (as well as two over 8-mers). Including reverse complements, we are guaranteed that a given transcription factor will bind to at least four locations on the array. We see from our previous result that, for $|\Lambda| = 18,206$ (the number of spots on our array) and a spot length of 44 (the length of our spots), the number of lit spots $|\Gamma_S|$ will be equal or nearly equal to $4 \cdot |S|$. For instance, when $|S| = 5$, $\Pr(|\Gamma_S| = 20) = 0.990$. When $|S| = 10$, $\Pr(|\Gamma_S| = 40) = 0.958$. If we proceed under the assumption that $|\Gamma_S| = 4 \cdot |S|$, we obtain Table 5.1. (The numbers are similar when $|\Gamma_S|$ is not much smaller than $4 \cdot |S|$.)

Having developed this theory for general subsets of $\mathbf{F}_4^k$, we must turn now to

our actual problem. There are two complicating factors. One is that $A = B(v, r)$ is now not a random subset of $\mathbf{F}_4^k$, but rather a subset of a Hamming ball. This is relevant, for instance, when we consider $B(\texttt{AAAAAAAA}, 1)$, which contains the element `AAAAAAAC`. These two elements are more likely to be contained on the same spot than two random elements. Still, this effect seems intuitively to be negligible; we will soon verify this computationally. The second, of course, is that we need to be working in $\widetilde{\mathbf{F}_4^k}$, and so we must account for the effect of reverse complements. The problem is sufficiently complex that a Monte Carlo approach seems justified.

For a given k and r, we created $B(\tilde{v}, r)$ for a random $\tilde{v} \in \widetilde{\mathbf{F}_4^k}$. We took as our motif S a random subset $S \subset B$, with $\frac{|S|}{|B|} = \frac{1}{4}$. We then checked against our array of spots Λ to find $\mathcal{B}(\Gamma_S)$. In our tests, running 1000 iterations each time, we found no instance when $|\mathcal{B}(\Gamma_S)| > 1$, for $k = 6, 7, 8, 9$ and $r = 1, 2, 3$. This seems to agree well with our theoretical results.

We can therefore be reasonably sure that for k and r in this range, our ball B will be the only element of $\mathcal{B}(\Gamma_S)$. We now must identify S itself, the set of binding sites within B.

Chapter 6

Discernibility

After recovering the Hamming ball, our second step is to identify which elements of the ball comprise S, the set of binding sites. Having identified the ball B, we look at the set Γ_S of lit spots on the array. For each $\lambda \in \Gamma_S$, the set $\lambda \cap B$ should be a non-empty set containing at least one element of S. However, if $|\lambda \cap B| > 1$, any or all of the elements of the intersection could be elements of S. This leads to the following definition.

Definition Let B be a Hamming ball. A binding site $v \in S$ is *discernible* with respect to B if there exists a spot λ such that $\lambda \cap B = \{v\}$.

Thus, if S consists entirely of discernible elements, we should be able to recover all the elements. If some elements are not discernible, we can only suggest that those elements may be in S if every spot on which they occur is lit. We can definitively exclude elements from S if spots they are on are *not* lit.

For a given Hamming ball in $B(v, r) \subset \mathbf{F}_4^k$ with radius r, what proportion of its elements are discernible? We can certainly suppose that as r increases, fewer and fewer elements are discernible, since a given spot $\lambda \in \Lambda$ is more likely to contain multiple elements of $B(v, r)$. For $5 \leq k \leq 9$, as k increases, discernibility should diminish, since fewer spots contain a given k-mer (and hence there are fewer chances for that k-mer to be discernible). Matters become more complicated when we move into $\widetilde{\mathbf{F}_4^k}$ to account for the effect of reverse complements.

(k, r)	Discernible
(6, 1)	100.00%
(6, 2)	99.59%
(7, 1)	100.00%
(7, 2)	99.75%
(8, 1)	100.00%
(8, 2)	99.88%
(8, 3)	96.93%
(9, 1)	99.95%
(9, 2)	99.53%
(9, 3)	91.06%

Table 6.1: Average percentage of discernible elements for balls $B(\tilde{v}, r)$.

For instance, if $v = \overline{v}$, then $B(v, r)$ is about half of its "normal" size, and so discernibility should increase.

Because of the complexity of these various parameters, we decided to estimate the discernibility rate via a Monte Carlo simulation. For a random center $\tilde{v} \in \widetilde{\mathbf{F}_4^k}$, we generated the ball of radius r around $\tilde{v}$. We then checked our array of spots, containing two 9-mer and two 8-mer De Bruijn sequences, to see what percentage of elements of $B(\tilde{v}, r)$ were discernible. Performing 1000 iterations of this routine gave us the results in Table 6.1.

In light of Theorem 4.1, we might wonder whether there would be a significant diffence between various De Bruijn sequences we might generate. To answer this question, we picked 500 primitive polynomials of degree 18 in $\mathbf{F}_2[x]$. For each polynomial, we generated corresponding the array of spots Λ, consisting only of this single De Bruijn sequence. For all $B(8, 3)$, we found how many times each

$v \in B$ was discernible. We took as our score for the polynomial the sum

$$\text{Score}(p(x)) = \frac{\sum_B \frac{\sum_{v \in B} \frac{d_{v,B}}{1+d_{v,B}}}{|B|}}{\text{number of balls}},$$

where $d_{v,B}$ is the number of spots where v was discernible with respect to B. The number of balls is just $|\widetilde{\mathbf{F}}_4^k|$. This score reflects the fact that we would like for v to be discernible at least once, but additional discernible spots are diminishingly helpful. We compute this average for each ball, then average over the averages. The result is a score between 0 and 1.

As a frame of reference, we give an maximum for the score above. Since $v \in \mathbf{F}_4^8$ and $p(x)$ is of degree 9, each v appears on at most 4 spots. Thus $\frac{d_{v,B}}{1+d_{v,B}} \leq \frac{4}{5}$, and likewise, $\text{Score}(p(x)) \leq \frac{4}{5}$.

Over our 500 tests, we obtained a mean of 0.4022 and a standard deviation of 0.0014. Our scores ranged from 0.3910 to 0.4043. This seemed like a small enough range for us to assert that the choice of polynomial was not significant.

Since we are to use not one but two De Bruijn sequences on our PBM, we also wanted to check that discernibility using a second polynomial is not affected by our choice for the first. Excluding the case where the two polynomials were the same (in which case we obtained a discernibility of 0.4899), the scores' mean was 0.5775 with standard deviation 0.0006 and range from 0.5727 to 0.5784. Again, our choice of the second polynomial given the first did not seem to matter.

Chapter 7

Conclusion

We thus have an overall strategy for recovering the motif from the pattern of lit spots on our array. Having constructed the array with a combination of various De Bruijn shift register sequences, we first run the experiment on the transcription factor of interest, whose binding motif lies in some Hamming ball of small radius. We showed in Chapter 5 that we have a high probability of recovering this Hamming ball, or one that is equally as good. By the discernibility calculations shown in Chapter 6, we again have an excellent chance of identifying the elements of the Hamming ball which make up the motif.

BIBLIOGRAPHY

[1] Bulyk ML, Huang XH, Choo Y, Church GM. Exploring the DNA-binding specificities of zinc fingers with DNA microarrays. Proc. Natl. Acad. Sci. U.S.A. 2001 Jun 12; 98(13): 7158-7163.

[2] Golomb, Solomon W. Shift Register Sequences, 2nd edition. Aegean Park Press. 1982.

[3] Heinemeyer T, Wingender E, Reuter I, Hermjakob H, Kel AE, Kel OV, Ignatieva, EV, Ananko, EA, Podkolodnaya, OA, Kolpakov, FA, Podkolodny NL, Kolchanov, NA. Databases on Transcriptional Regulation: TRANSFAC, TRRD, and COMPEL. Nucleic Acids Res. 1998; 26: 364-370.

[4] Hurlbert, Glenn H. On Spanning Trees of Certain Graphs. Unpublished [Colloquium, University of California, Santa Barbara, CA]. 1993. `http://math.la.asu.edu/~hurlbert/papers/STCG.ps`.

[5] Roth FP, Hughes JD, Estep PW, Church GM. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. Nat Biotechnol. 1998 Oct; 16(10): 939-45.

[6] Sandelin A, Wynand A, Engstrom P, Wasserman W, Lenhard B. JASPAR: an open access database for eukaryotic transcription factor binding profiles. Nucleic Acids Res. 2004 Jan; 32(1) Database Issue.