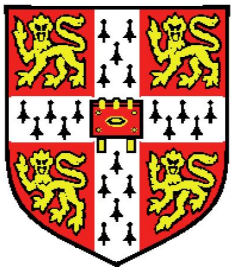
Computational Methods for the Measurement of Protein-DNA Interactions



Daniel James

Wellcome Trust Sanger Institute

University of Cambridge

A thesis submitted for the degree of

Doctor of Philosophy

February 2017

Abstract

Daniel James

Computational Methods For The Measurement Of Protein-DNA Interactions

It is of interest to know where in the genome DNA binding proteins act in order to effect their gene regulatory function.

For many sequence specific DNA binding proteins we plan to predict the location of their action by having a model of their affinity to short DNA sequences. Existing and new models of protein sequence specificty are investigated and their ability to predict genomic locations is evaluated.

Public data from a micro-fluidic experiment is used to fit a matrix model of binding specificity for a single transcription factor. Physical association and disassociation constants from the experiment enable a biophysical interpretation of the data to be made in this case. The matrix model is shown to provide a better fit to the experimental data than a model initially published with the data.

Public data from 172 protein binding micro-array experiments is used to fit a new type of model to 82 unique proteins. Each experiment provides measurements of the binding specificity of an individual protein to approximately 40000 DNA probes. Statistical, 'DNA word', models are assessed for their ability to predict held back data and perform very well in many cases.

Where available, ChIP-seq data from the ENCODE project is used to assess the ability of a selection of the DNA word models to predict ChIP-seq peaks and how they compare to matrix models in doing so. This *in vitro* data is the closest proxy to the true sites of the proteins' regulatory action that we have.

Dedicated to my grandmother Myrtle. Sorry it took so long.

Acknowledgements

I would like to thank the Wellcome Trust and the Sanger Institute for their generous funding and support throughout.

Contents

List of Figures

Nomenclature

$\mathbb{1}(x)_{\{x\in \mathrm{a}\}}$ Indicator function: 1 on set A, else 0

M $\mathrm{mol\,dm^{-3}}$

L_p Metric space with p-norm

p-norm $||\mathbf{x}||_p = (\sum_i^n |x_i|^p)^{1/p}$

A,C,G,T Nucleobases: adenine, cytosine, guanine, thymine

B-spline Polynomial approximation functions with local support

bHLH Basic helix-loop-helix TF family

ChIP Chromatin immunoprecipitation

Chip Alternative name for micro-array

HK Designation of particular PBM design

HU Bacterial histone analogue

Lasso Sparse regression method

LFSR Linear feedback shift register

ME Designation of particular PBM design

Motif Model of specific nucleotide sequence

PBM Protein binding micro-array

PDB Protein data bank

PPM Position frequency matrix

PPM Position probability matrix

PWM Position weight matrix

RT Product of molar gas constant and temperature

TF Transcription factor

Chapter 1

Introduction

In the author's opinion the study of protein-DNA interactions is an interesting problem because we would like to mimic some of the engineering feats of Nature.[1] Whilst RNA may be able to replicate in the absence of proteins under certain conditions, anything more complicated, e.g. the simplest viruses, require some protein DNA interactions in order to function.

In this introduction there is a brief characterisation of the central objects in this thesis, namely DNA and DNA binding proteins. A brief overview of some historical papers that are of significance to the study of DNA protein interactions is then given. From this historical summary an idea of progress can be had. On the one hand it is remarkable to think that 50 years ago the molecular structures and basic functions of DNA and binding proteins were only just becoming available. On the other it is interesting to observe that certain questions, asked early on, appear to be have been left unanswered whilst others have been re-visited again and again.

[1] Richard Feynman famously said, 'What I cannot create, I do not understand.' Therefore, in order to understand the most basic cell or virus we must be able to create it. As another point of trivia, Craig Venter et al recently encoded the words, 'What I cannot build, I cannot understand.' in a synthetic genome.

1.1 DNA

DNA is a composition of the four nucleobases adenine, cytosine, guanine and thymine in two linear sequences, each linear sequence twisted around the other to form a double helical, macro-molecular structure. The composition and local structure is well described but the three dimensional dynamic structure of the macro-molecule is not as simple.

Cellular DNA is often thought to be found in the B-DNA structure, as published by James Watson and Francis Crick in 1953 but sometimes, if not frequently, DNA may occur in single stranded 'bubbles' [1], sometimes in families of large, three dimensional, structures known as g-quadruplexes [2] sometimes in alternative A-DNA, C-DNA or Z-DNA helical forms.

More recently an abundance of geometrical configurations of DNA have been engineered *in vitro*, some of these are beginning to have applications *in vivo* [3, 4]. Together with structural proteins, DNA *in vivo* is further folded and packed resulting in the objects we call chromosomes.

An example of the tertiary structure of DNA in prokaryotes is super-coiling induced by bends, created by the bacterial DNA binding protein HU [5].

In eukaryotes there are several levels of DNA organisation within the nucleus [6]. In recent years it has become possible to speak of the topology of the eukaryotic genome [7, 8]. The organisation of chromosomes within the nucleus can vary during the cell life cycle, we are beginning to understand the significance of chromosome topology.

All this being said, for the rest of this thesis, what we will understand DNA to be is a B-DNA molecule; uniform and still, in all its glorious symmetry. Just like the model in the atrium of your local genomics institute.

1.2 DNA binding proteins

There are many proteins that interact with DNA for many different purposes. For packing, bending, winding, cutting, joining, copying and annotating DNA there are histones, topisomerases, nucleases, ligases, polymerases and methyl transferases, all of which interact proximally with the DNA molecule and have func-

tions concerned with the processing of information stored in DNA [6, 9–11].

In order to describe what a DNA binding protein is we will need to agree on what we mean by bind. The word 'bind' implies some physical interaction but are all interactions significant? For instance, the sugar-phosphate backbone of DNA is negatively charged and hydrophilic and so we might call a DNA binding protein 'any protein with a positively charged residue' or 'any protein able to make polar bonds'.

In section 1.2.1 some historical context for the expression 'sequence specific DNA binding protein' is given. Following this (Section 1.2.2) are descriptions of some well known transcription factors and methods used to characterise them. An effort has been made to include several of the transcription factors that appear later in this thesis.

1.2.1 Early bio-physical models of DNA protein interactions

In von Hippel and McGhee's 1972 review, 'DNA-protein interactions', the authors firstly categorise DNA *interacting* proteins between those that promote the flow of information and those that repress the flow of information [12]. The authors proceed to sub-categorise DNA-interacting proteins into those that interact with specific nucleotide sequences and those that interact with DNA 'non-sequence specifically'. In their review several potential complexities of DNA protein interactions are discussed, some of these are listed next,

- The interaction of a number of functional groups of both the protein and DNA molecule must be involved in order to allow sufficient free energy change for the tightness of observed interactions.
- Groups must be positioned, at least temporarily, in specific conformational ways that permit thermodynamically favourable interactions.
- Recognition of specific binding sites can be a compound process where both DNA melting and specific nucleotide occurrences are required or denied.

- The ready inter-conversion of B form to A and C form double helix suggests that the predominant B form might easily be deformed into structural perturbants in solution.
- Cross-groove phosphate-phosphate spacings vary, providing local conformation distortions that can be recognised by a protein.
- Proteins may recognise base tilt or twist relative to the helical axis.
- Proteins may recognise pitch and stacking differences.
- Chemical modifications such as methylation, glucosylation and single strand 'nicks' may be recognised by a DNA binding protein.
- The pathways used by a protein in searching for an interaction site are likely more complicated than a simple 3D diffusion.
- Protein protein interactions for proteins of the same type and different types can affect binding affinity. Those of the same type may aggregate if the binding affinity is increased by their cooperation.

Von Hippel and McGhee's review was written at a time when the sequence of the Lac operon had yet to be determined and the only other gene regulatory protein identified was the λ-repressor. Already though, the authors have described a formidable set of physical parameters that should be quantified in a realistic physical model of transcription factor activity. Apart from in a small number of cases, many of the parameters described in the review are still hard to measure accurately. The parameter that is often given in a physical measurement of protein-DNA interaction is the association constant. A short review of measurements of constants for the lac repressor is given in the introduction of the next chapter of this thesis (Chapter 2.1).

The archetypal 'sequence specific' DNA binding protein must originate from the early 1960s and the work on gene regulatory mechanisms by Jacob and Monod

on the lac repressor[1] and that of Mark Ptashne and others on the λ-phage repressor [13, 14]. It was established that the lac repressor was a protein and that it bound to the DNA of the lac operator [14, 15]. A quantitative description of the interaction between the lac repressor and the lac operator was given by means of an assay that records the amount of repressor-bound operator DNA stuck to a nitrocellulose filter. Thermodynamic and kinetic parameters were obtained. Thermodynamic means 'association constant' and kinetic means 'on and off rates'; values for these parameters will be discussed further in chapter 2. At this early stage, in 1968, it was also discovered that a mutated operator sequence bound the repressor protein less well, i.e. had a smaller association constant [16].

A nucleotide sequence for the lac operator was not published until 1973 [17]. Several years later again, in 1980, it was found that the repressor actually bound to other nucleotide sequences in the *E. coli* genome [18]. By the mid 1980s large collections of nucleotide sequences that were candidate protein interaction targets were becoming available. An early example described in 1983 was for sequence differences at the -10 positions for 168 *E. coli* promoters [19]. The necessity of computers for the processing of such data-sets was becoming apparent [20].

In the mid 1980s Stormo provided commentary on ways to present sequence statistics, ways to parametrise models of protein-DNA interaction and links between the sequence statistics and physical parameters [21, 22]. Links between sequence statistics and physically measurable quantities were being discussed by others [23] during the same period.

Throughout the 1980s and 1990s several new families of DNA binding proteins were discovered. Some of these are described in section 1.2.2. With growing libraries of nucleotide sequences, the early 2000s saw a large number of algorithms described for the inference of DNA binding loci in sequence data. A selection of these algorithms are described in section 1.5.

[1] In their early publication, studying the Lac repressor of *E. coli* it was not understood that the 'repressor' responsible for regulating the β-galactosidase gene was necessarily a protein, the authors speculated that this 'repressor' might in fact be an RNA molecule. This was speculated to be the repressor transcript.

Polymerase	1099
Helix turn helix	537
Zinc finger	406
Homeodomain	213
Helix loop helix	161
Histone	159
High mobility group	114
Ribosome	81
Topoisomerase	73
Leucine Zipper	51
Forkhead	14

Table 1.1: PDB DNA binding structures by type or domain keyword.

Homo sapiens	754
Mus musculus	171
Escherichia	374
Saccharomyces	151
Drosophila	49

Table 1.2: PDB DNA binding structures by organism keyword.

1.2.2 Sequence specific DNA binding proteins

In the early 1980s crystal structures of DNA protein complexes were established. The λ-phage proteins, Cro and cI, and the CAP protein of *E. coli*, were the first to be described. For the first time researchers had a validated visual model for how sequence specific, gene regulatory, proteins might perform their function. It took longer to obtain a crystal structure for the lac repressor. The lac repressor is a tetrameric protein that apparently yielded less easily to crystalisation [24]. The first structures were all of the helix turn helix variety.

At the time of writing, a search in the PDB for entries that include protein and DNA reveals 2903 structures. Refining the search with keywords provides the results in tables 1.1 and 1.2. The key-word ribosome turns up because of many results for structures containing DNA and protein.

Perhaps the largest net that can be cast to catch 'transcription factors' is via sequence homology. Any part of the genome that contains a candidate protein coding sequence can be assessed for containing a DNA binding domain. We

can then classify all such candidate proteins as transcription factors. Pfam offers a catalogue of protein domains including DNA binding domains [25]. The number of 'transcription factors' is estimated to be 300 for *E. coli* and 3000 for humans, based upon homology of DNA binding domains taken from amino-acid sequences [26].

DNA binding proteins can be grouped into protein families [25, 27]. These families can be constructed using structural similarity, amino acid sequence similarity and nucleic acid sequence similarity. Proteins can be structurally, functionally or evolutionarily related. A selection of sequence specific DNA binding proteins will be described next with some reference to their families and when they were first studied. Since the classification of these proteins into families is in part a subjective exercise, historical context is useful to understand how they have come about. Representative crystal structures from these families are drawn, all of which have been taken from the protein data bank [28]. Physical structures guide our physical intuition about the dynamics of the molecular interactions involved in the DNA protein interaction and remind us that these interactions are likely to be complicated. Looking at the complexity and diversity of the shapes of these proteins convinces us that a 'simple recognition sequence' view of DNA binding is unlikely to well describe all the interesting behaviour.

1.2.2.1 Helix turn helix

The lac repressor and cro repressor both have DNA binding domains of this type. The cro repressor was the first DNA binding protein complex to be crystalised and so was also the first protein DNA binding interaction to 'have its picture taken' [24]. The helix turn helix family contains the Homeo domain, the Myb DNA binding domain and the POU DNA binding domains [25] (Figure 1.1).

1.2.2.2 Basic helix loop helix

This family of DNA binding domains includes the transcription factors that are under discussion in the first chapter. This domain is characterised by two α-helices connected by a loop. The protein forms homo and hetero-dimeric complexes when binding to DNA. The DNA binding domain is associated with an

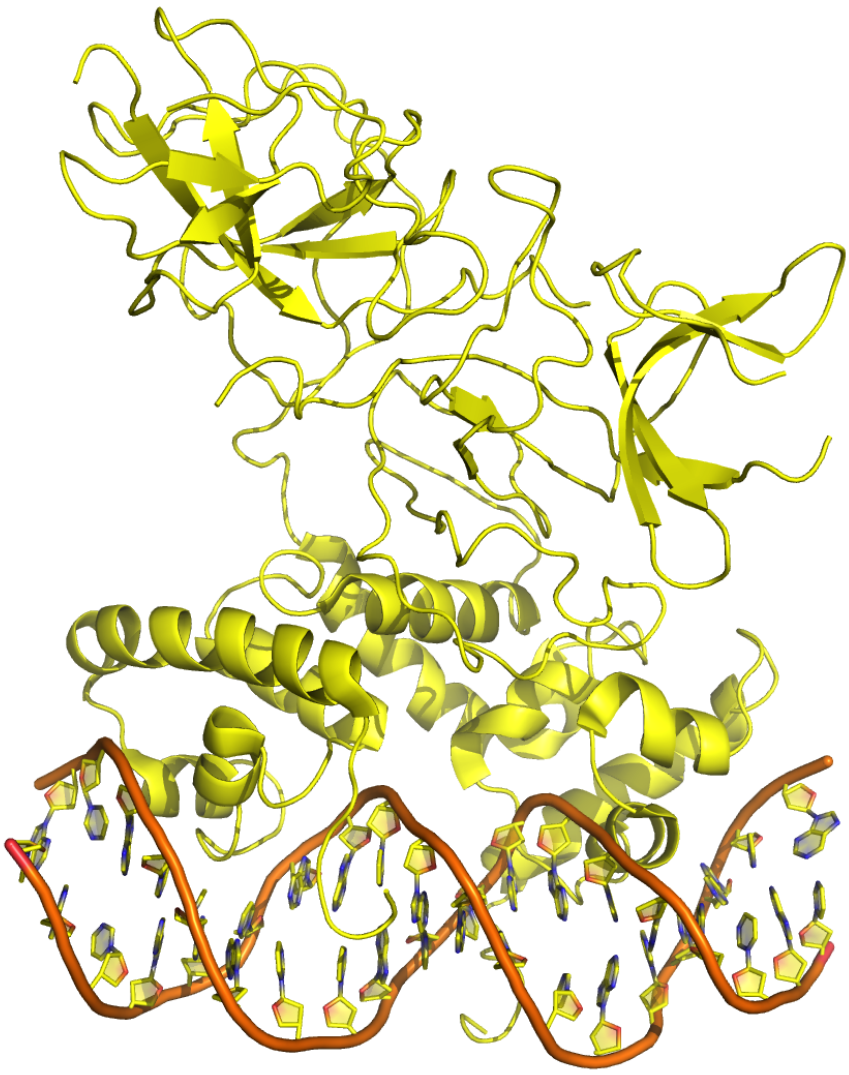


Figure 1.1: DNA protein structure for the lambda repressor. This helix-turn-helix domain is shown as part of PDB structure 3bdn.

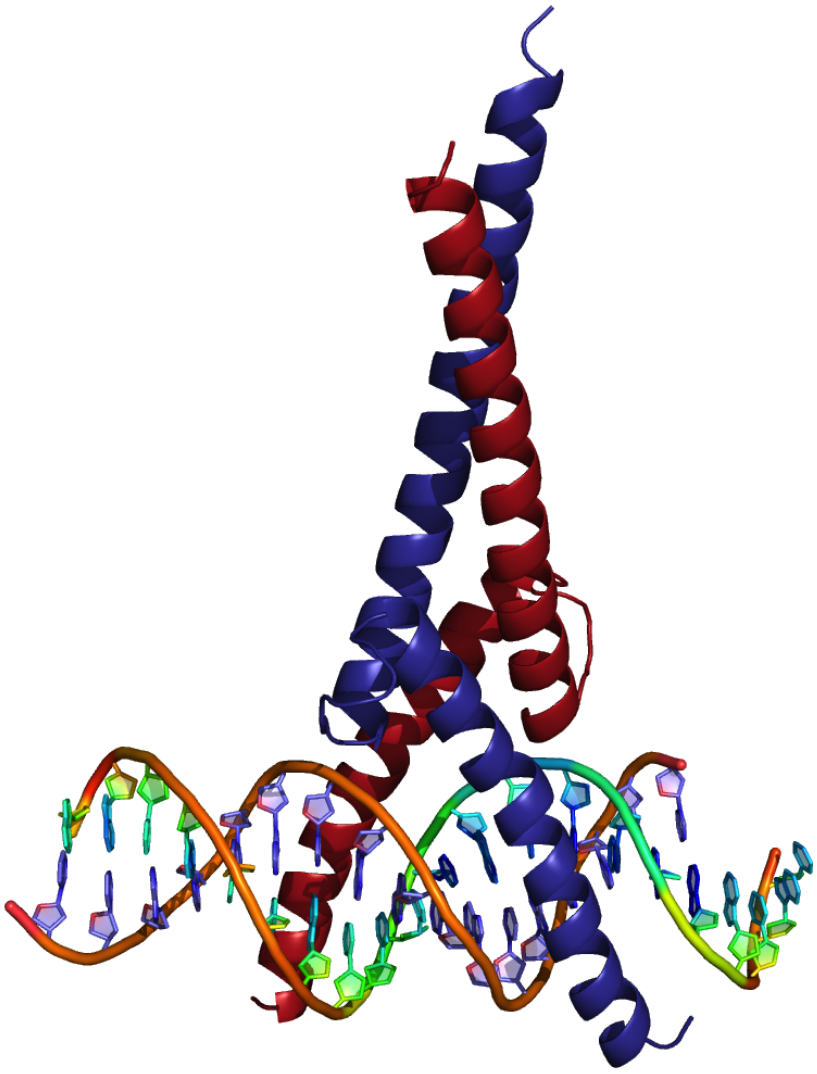


Figure 1.2: DNA protein structure for SREBP-1A. This helix-loop-helix domain is PDB entry 1am9.

'E-box' consensus sequence of CACGTG. This domain is sometimes found in tandem with another, such as the leucine zipper domain, that aids dimerisation [29]. The first examples of this type of binding domain were discovered in the late 1980s as mouse transcription factors [30]. The SREBP, Myc, Max and Clock proteins have domains of this type (Figure 1.2).

1.2.2.3 Basic leucine zipper

This protein often forms dimers that recognise palindromic DNA sequences. This family includes the Fos, Jun, Cebpb, Atf and Nrf2 proteins [25] (Figure 1.3). The 'leucine zipper' describes the periodic placement of leucine residues that 'zip' the protein into its dimeric form. This appears to have been first proposed in 1988 [31].

1.2.2.4 Forkhead domain

The first forkhead protein was identified from mutations of the forkhead gene of *Drosophila* in 1989 [32] (Figure 1.4). These are described as pioneer proteins because they are capable of directly decondensing chromatin [33].

1.2.2.5 High mobility group domain

These are the DNA binding domain of the chromatin associated high-mobility proteins. These proteins bind to DNA in forms other than the B-DNA form. Proteins containing this domain include the SRY, SOX and TCF factors [34] (Figure 1.5). The HMG proteins were originally discovered in the mid 1970s but their DNA binding characteristics were not established until the early 1990s [35].

1.2.2.6 Zinc finger

The first eukaryotic transcription factor to be described was a member of a new family of DNA interacting proteins, now known as zinc finger proteins [36] (Figure 1.6). Unlike the helix turn helix motif of the lac repressor and cro repressor the zinc finger proteins have repeating 'fingers', each containing a zinc ion. The structure and functionality of one of these proteins was first described in the mid 1980s [37].

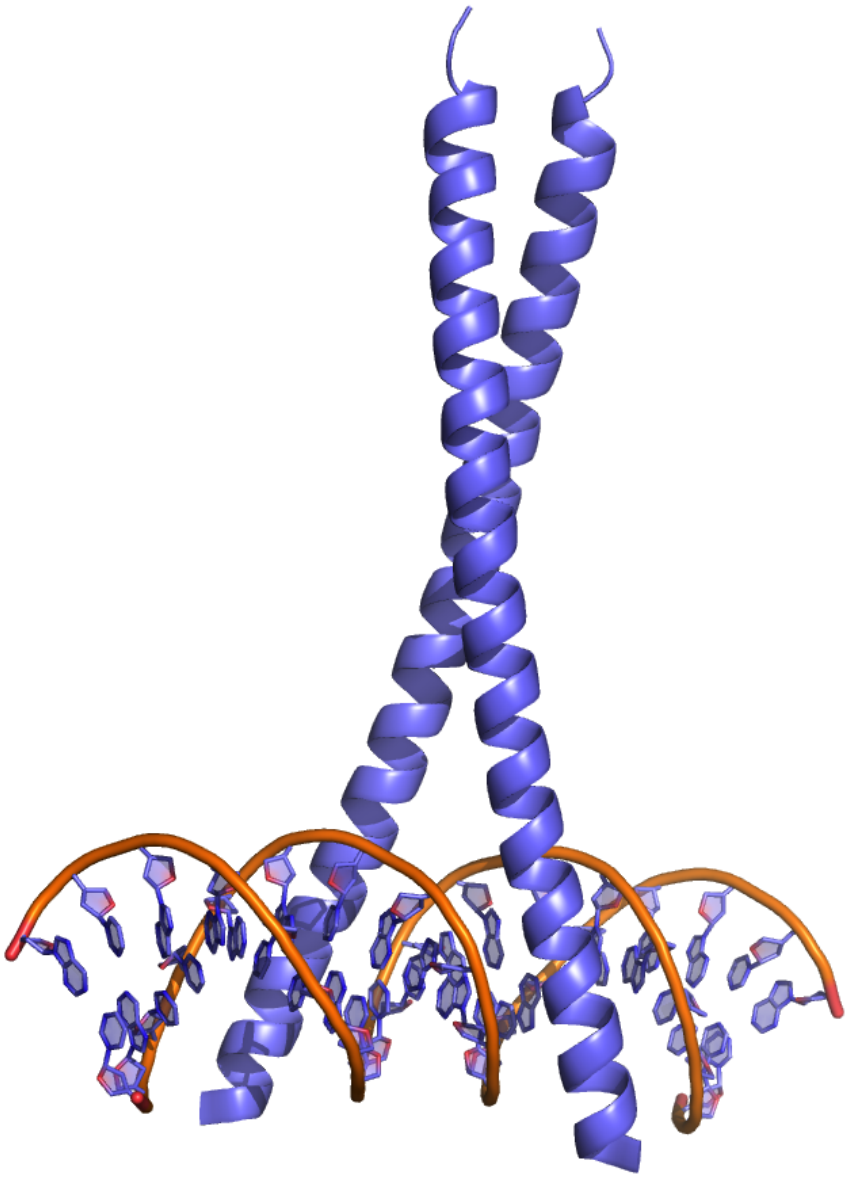


Figure 1.3: DNA- protein structure for the JUN BZIP homo-dimer. This basic leucine zipper protein is PDB entry 2h7h.

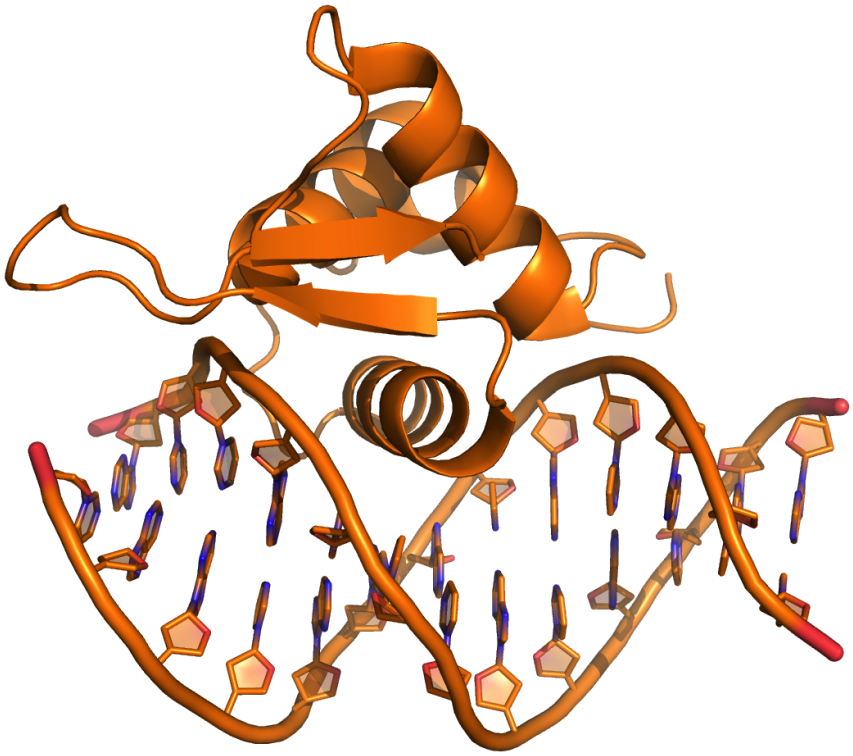


Figure 1.4: DNA- protein structure for Foxo4 DNA binding domain. This basic forkhead box domain is PDB entry 3l2c.

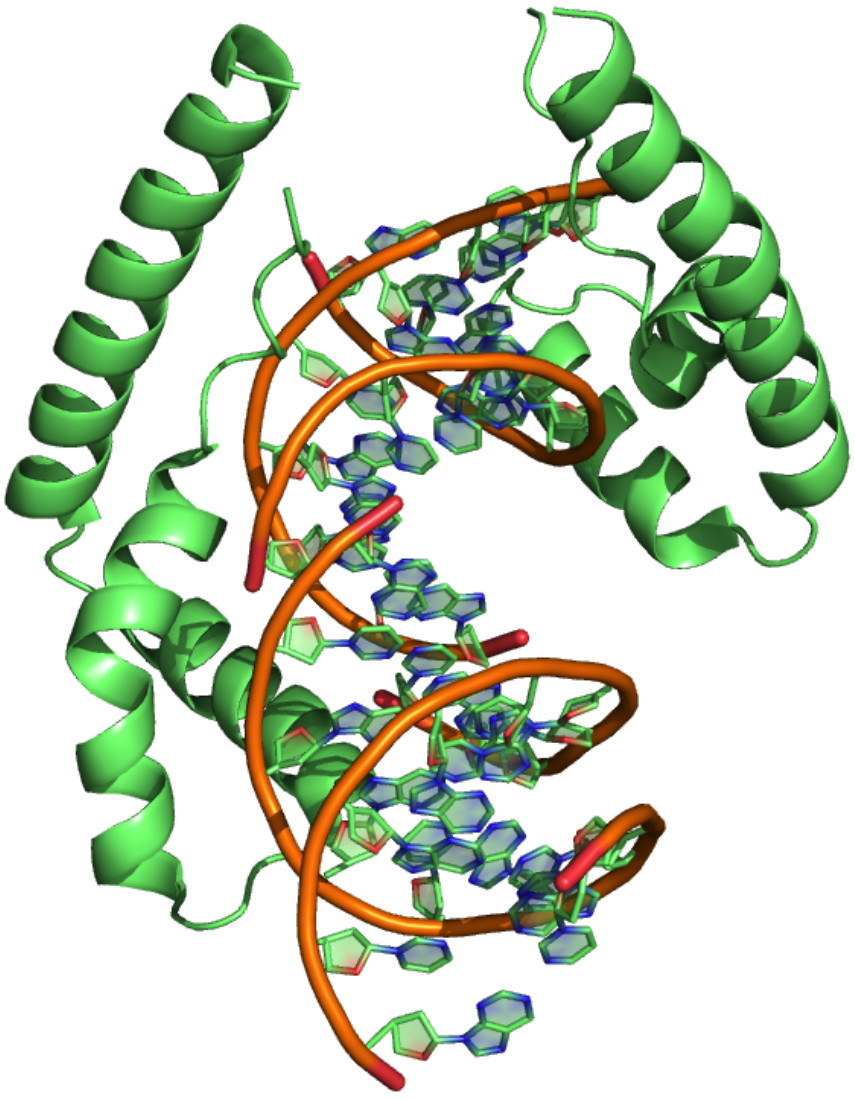


Figure 1.5: DNA-protein structure for the high mobility group D protein bound to DNA. This is PDB entry 2nm9.

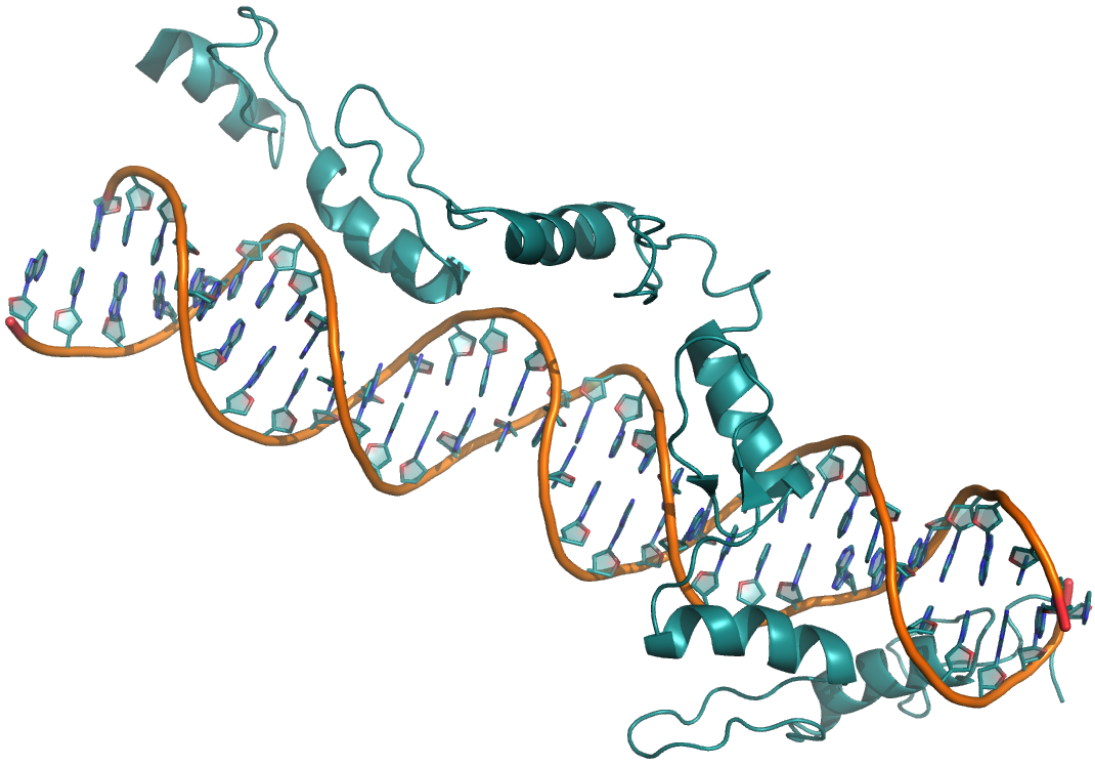


Figure 1.6: DNA-protein structure for a zinc finger protein bound to DNA. This is PDB entry 1tf6.

By the mid 1990s this class of sequence specific DNA binding proteins, engineered to carry nuclease machinery, had become an essential tool in genome manipulation.

It is now possible to design zinc finger proteins to match unique genomic sequences and so target attached machinery to a specific target [38].

Genome editing is currently a hot topic, particularly the CRISPR-Cas9 system [39]. Specificity is a major hurdle to therapeutic applications, libraries of zinc-finger proteins could offer what is needed [40].

1.3 Experiments that identify sequence specific DNA interactions

The discovery of sequence specific DNA binding proteins developed along with the techniques to determine their sequence specificity experimentally. These techniques will be the subject of this section. The following are chosen to demonstrate the major techniques and also some of the breadth of technologies that have been used.

ChIP-chip

The 'ChIP' here stands for Chromatin ImmunoPrecipitation and the 'chip' means chip as in microarray.

In ChIP-chip covalent cross-links are made between proteins and DNA *in vivo*. An antibody for the protein of interest is used to immunoprecipitate the protein-DNA fragments.

The DNA fragments are then labelled in an amplification reaction and then hybridized to a DNA micro-array in order to identify the fragments [41].

The availability of an antibody for the immunoprecipitation step is a constraint and also the specificity of the antibody is not guaranteed due to their variability.

The distribution of DNA fragments can be biased by the immunoprecipitation, amplification or labelling steps. For example, protein-protein interac-

tions could cause the immunoprecipitation of DNA fragments that are not directly interacting with the DNA.

The region of genomic DNA that is mapped to the fragment will be of greater length than a transcription factor binding site typically. Further statistical inference is therefore needed in order to determine protein-DNA sequence specificity.

ChIP-seq

As with ChIP-chip, DNA fragments are immunoprecipitated after cross-linking. In contrast to ChIP-chip, after amplification the DNA fragments are directly sequenced, rather than being hybridized to an array [42]. This offers the detection of any sequence rather than being limited to a subset of probes chosen for an array. Motif finding algorithms are applied to the sequenced fragments to localise binding sites [43] since the sequenced fragments will typically be orders of magnitude longer than a transcription factor binding domain.

DamID

In this experimental approach a fusion protein is made between the protein of interest and Dam. The Dam protein methylates GTAC sequences in the vicinity of the protein of interest. The fusion protein is introduced *in vivo* via a plasmid. No antibody is required in this case, which is an advantage over ChIP-seq. Genomic sequence from a control experiment is compared to that of the methylated sequences to determine binding loci [41].

DNase I footprinting

This method of binding site identification was first described in1978 [44]. In brief, regions of DNA that have bound proteins are protected from a DNase enzyme allowing the inference of bound sequences.

More recently this technique has been made genome wide and used to yield hundreds of thousands of base pair resolution footprints [45].

SELEX

The acronym is Systematic Evolution of Ligands by EXponential enrichment. A target protein is exposed to a library of DNA oligonucleotides. A sequence of rounds of gel shifting with bound protein and PCR amplification of bound DNA segments results in a collection of DNA segments that are biased towards those bound by the protein. SELEX has long been used to test a protein's affinity to synthesised sequences [46]. A high throughput version of SELEX is now also available [47].

Electromobility shift assay (EMSA)

The essence of the experiment is to separate unbound DNA fragments from complexes by electrophoresis in polyacrylamide gels [48]. Initially this method allowed the estimation of association constants for the CAP protein from the lac system in *E. coli* that had not been successfully obtained via the nitrocellulose filter assay. In this early experiment the half life of the CAP protein, bound to its lac operator site, was estimated to be more than an hour. For the experiment to be successful the disassociation rate of the protein from the DNA must be longer than the time taken to separate the unbound DNA from the bound.

ELISA

The acronym is 'enzyme-linked immuno-absorbent assay'. This assay has many applications including the measurement of DNA-protein interactions. A recent publication describes the ability to measure the interaction of a protein with 341 dsDNA probes [49]. The dsDNA probes are secured on a micro-titer plate before adding the protein under study. Bound protein is detected via the binding to an antibody that can be fluorescently detected. The exact position of binding has to be inferred afterwards in much the same way as for the protein binding micro-arrays discussed below.

Surface Plasmon Resonance

Protein is secured to a metal surface and real-time measurement of plasmons, detected via shifts in the reflection of a light source, gives information on binding kinetics and binding constants [50]. The measurement

of on-rates and off-rates makes this technique particularly interesting for quantitative research.

Microfluidics

Chapter 2 gives a more detailed description of a particular microfluidic device [51]. This type of experiment represents an attempt to obtain equilibrium constants with high sensitivity and minimal bias.

DNA arrays

Different types of micro-array have been used for the direct measurement of DNA-protein interaction, these are reviewed by Bulyk [41]. One approach [52] used to measure DNA-protein affinity is an experiment where large sections of genomic DNA are 'tiled' onto arrays. These arrays are sometimes described as protein binding micro-arrays but are different from those that shall be referred to as protein binding micro-arrays hereafter.

Protein binding micro-array

A protein binding micro-array, (PBM), assay is described by Berger & Bulyk [53]. This assay is the model for the production of the data used in the final chapters of this thesis. This type of array seems to have been first described in 1999 [54]. In brief, every possible sequence of a certain length, (ten base pairs in our case), are represented as DNA probes on an array, a transcription factor protein or binding domain is then added to the array and differential binding between the probes is measured.

1.4 Models of sequence specificity and their uses

The different types of experiment described in the previous section produce a variety of types of data. However all describe interactions between DNA and protein. This section describes the various approaches that have been developed to create standardised computational models.

For any model of sequence specific protein DNA interaction we should have a measure, for any nucleotide sequence of a suitable length, of sequence affinity.

We can thereby obtain a number for every locus in a genome and this gives us a 'genome annotation'.

If we are able to predict the affinity of a binding protein for any given nucleotide sequence then we would be a step closer to engineering molecular systems with desired behaviour.

Some limitations to be noted are,

- At present most of the models described can, at best, give a value for an association constant of binding. This will be an equilibrium value that does not tell us about on rates and off rates for instance.
- It is also not clear that expressions used to derive binding probabilities, using concentrations, are applicable in non-equilibrium states and without taking into account chemical activities.
- Protein-protein interactions and protein-ligand interactions are not represented by a model of protein-DNA affinity. These sorts of interactions will likely be important in constructing even the simplest molecular systems.

Next follows descriptions of some of the notations used when describing a model of sequence specificity. The word 'matrix' appears a lot in these descriptions and it might be worth noting that these are not matrices in the sense that is often meant, i.e. they are not linear maps between vector spaces over a field. What they are is a way to define a function from DNA sequences of fixed length to a single number, integer or real.

1.4.1 Consensus sequence

Fifty years ago, when it was difficult to determine even a single sequence that was bound by a single protein, there was only one way to represent a protein's sequence specificity, namely, to write it down.

Given a handful of sequences a particularly simple approach is to start from a multiple alignment, record the most frequent nucleotide at each of the loci, and to call this the 'consensus sequence'.

If we have sequences $s_1, s_2, \ldots, s_N$, each of length m, then denote the jth nucleotide of the ith sequence as s_{ij}. The counting function f_{ij} gives the number of nucleotides of each type at position j (Equation 1.1).

$$\begin{aligned} f_{1j} &= \sum_{i}^{N} \mathbb{1}_{\{s_{ij}=A\}} \\ f_{2j} &= \sum_{i}^{N} \mathbb{1}_{\{s_{ij}=C\}} \\ f_{3j} &= \sum_{i}^{N} \mathbb{1}_{\{s_{ij}=G\}} \\ f_{4j} &= \sum_{i}^{N} \mathbb{1}_{\{s_{ij}=T\}} \end{aligned} \tag{1.1}$$

In each case, here and elsewhere, we identify the nucleobases $\{A, C, G, T\}$ with the indices $\{1, 2, 3, 4\}$ respectively.

The consensus sequence at position j is then the maximum of $f_{1j}, f_{2j}, f_{3j}, f_{4j}$. Extensions to this scheme that use extra letters of the alphabet to stand for one of two different bases are also sometimes used.

1.4.2 Position Frequency Matrix

A refinement on the consensus sequence is to record the frequencies of each nucleotide, at each position, in a matrix. We can call this a position frequency matrix, (PFM). Using the notation of equation 1.1, the PFM can be written as,

$$(f_{ij}) \qquad i = 1, 2, 3, 4 \qquad j = 1, 2, \ldots, n \tag{1.2}$$

Information on the joint distribution of nucleotides has been lost in this case but it is often thought that there is a reasonable degree of independence between positions. Under this assumption the position frequency matrix retains the information available in the original alignment.

1.4.3 Position Probability Matrix

If we normalise the columns of the PFM, to give probability distributions, we have,

$$(p_{ij}) \qquad i = 1, 2, 3, 4 \qquad j = 1, 2, \ldots, n \tag{1.3}$$

This representation actually contains less information than the PFM since we do not know how many observations were made in the creation of the matrix. We will call the normalised version the position probability matrix, (PPM).

1.4.4 Position weight matrix

In 1986 stormo described using weight matrices to model the specificity of a collection of *E. coli* promoter sequences [22]. The matrix is essentially a function from nucleotide sequences to real numbers. Using a PPM or a PWM we can score a sequence by reading a value from each column. With a PPM we multiply probabilities but with a PWM the values are added, having moved to the logarithmic scale.

The biases in the genome for particular nucleotides can be considered when doing the prediction and for this the following strategy is often employed. The occurrence of a given n-tuple nucleotide sequence, $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, is modelled as the realisation of a product of n, independent, multinomial random variables, each with parameters taken from a column of the PPM.

$$\begin{aligned} m_i &\sim \text{Multinomial}(1;\ p_{1i}, p_{2i}, p_{3i}, p_{4i}), \qquad i = 1, 2, \ldots, n \\ m &\sim \prod_{i}^{n} m_i \\ M &= \{\mathbf{a} \text{ is a sample from m}\} \\ \mathbb{P}(M) &= \prod_{i=1}^{n} p_{a_i i} \end{aligned} \tag{1.4}$$

Where a_i is taken to be an integer index in the natural way.

Similarly, given a genomic frequency distribution (q_1, q_2, q_3, q_4) of $\{A, C, G, T\}$, we can write,

$$\begin{aligned} b &\sim \text{Multinomial}(1;\ q_1, q_2, q_3, q_4) \\ B &= \{\mathbf{a} \text{ is a sample from b}\} \\ \mathbb{P}(B) &= \prod_{i=1}^{n} q_{a_i} \end{aligned} \tag{1.5}$$

The 'score' of an n-mer sequence can then be calculated as,

$$S(\mathbf{a}) := \log \prod_{i=1}^{n} \frac{p_{a_i i}}{q_{a_i}} = \sum_{i=1}^{n} \log \frac{p_{a_i i}}{q_{a_i}} \tag{1.6}$$

In statistics this would be called the likelihood ratio of the sequence, $\mathbf{a}$, being a sample from the matrix distribution, m, over the null hypothesis, which is that $\mathbf{a}$ is a sample from the background distribution, b.

Since the availability of massive amounts of genomic sequence data, matrix models have been derived from promotor sequences or enhancer regions of an organism's genome [55–57]. These probablistic models have as their output a PPM or PWM.

1.4.5 Energy matrix model

The following chapter of this thesis will give more details of this approach and so just a summary will be given here. It is worth noting that there is a correspondence between position weight matrices and energy matrices that follows from statistical thermodynamics, i.e. by taking logarithms of probabilities we 'move to an energy scale'.

A protein DNA interaction can be described by a change in Gibbs free energy that results from a change in molecular conformation. This quantity is commonly given the symbol ΔG. If we fix the value of ΔG for a particular sequence then we can then speak of 'changes in the change' in Gibbs free energy, or $\Delta\Delta G$, by changing nucleotides in the sequence. Any change in nucleotide sequence will have its own corresponding $\Delta\Delta G$, the original sequence will have a $\Delta\Delta G$ of zero.

In this case, assuming that individual nucleotides contribute additively to the total binding energy, we will use the matrix in equation 1.7 to represent the function from nucleotide sequences to the values of $\Delta\Delta G$ analogously to the position weight matrix,

$$\begin{aligned} (\epsilon_{ij}) \qquad & i = 1, 2, 3, 4 \qquad j = 1, 2, \ldots, n \\ \Delta\Delta G(\mathbf{a}) = & \sum_{i=1}^{n} \epsilon_{a_i i} \end{aligned} \tag{1.7}$$

1.4.6 Extended matrix models

Energy matrix models can be extended to allow for dependencies between neighbouring pairs of nucleotides. In this case, rather than having 4 rows, an energy matrix could have 16 rows, i.e. one row for each pair of nucleotides. An idea behind this extension is that it retains some relevant information on the physical effects of neighbouring nucleotide interactions. The same extension can be made for a position weight matrix. In the latter case a distribution can be calculated for di-nucleotide frequencies [58].

1.4.7 Mutual information

A further extension to a matrix model is to use mutual information. This is a way to measure the degree of dependence between different distributions. The joint probability distributions of pairs of positions are used, the independence of columns would be described as their having zero mutual information. However this approach requires enough data to reliably estimate the joint probability distributions from which the mutual information is calculated [59].

1.4.8 Weighted words model

Rather than adding or multiplying values for each position, via a matrix representation, we might choose to assign a value to each of a large set of longer 'DNA words'. This approach has been used previously together with special data structures and clustering methods [60, 61]. The number of possible words grows

very quickly with length, i.e. 4^l where l is the length of the longest possible word. In order to create a model of binding in this some sort of sparsity in the set of possible parameters is wanted. Also, plenty of data is required in order to estimate the parameters. Such a method, which is the subject of the later chapters of this thesis, is described in section 4.2.2.1.

1.5 Algorithms used to identify sequence specific DNA interactions

There have been many algorithms published that obtain matrix models of DNA protein binding. The differences between these algorithms often appear small on close inspection [57, 62]. The below is a collection of popular such algorithms that the author has investigated in some detail.

Multiple sequence alignment algorithms

A multi-species multiple alignment can give evidence for sequence conservation. Sequence conservation might imply functional importance; a conserved, non-coding, region of the genome could be inferred to have some function, possibly interacting with proteins. The latter point suggests that multiple sequence alignments can be used to guide motif finding or to weight predictions of protein binding.

On the other hand a lack of sequence conservation might not imply a lack of functional importance. For example species such as human and mouse have a degree of phenotypic variation that is not well explained by variation in protein coding genes. We therefore expect to see greater variation in regulatory regions of the genome and hence transcription factor binding sites may not be well conserved within a multiple sequence alignment [63].

MEME

MEME, or Multiple EM for Motif Elicitation is a popular algorithm for the inference of statistically significant subsequences in a collection of larger sequences [55]. The input sequences do not need to be aligned when using this approach.

Expectation maximisation, EM, is a method for maximising a likelihood function by integrating over data that enables the calculation of conditional expectation.

MatrixREDUCE

This algorithm describes being able to determine bio-physical constants from either ChIP-seq of protein binding micro-array data. An energy matrix is obtained by optimising an objective function over the elements of the matrix [64]. The objective function is very similar to that of NestedMICA.

NestedMICA

This algorithm describes its ability to use a background model of genomic sequences to improve its ability to find sequence motifs [65]. The motifs are obtained by the maximisation of a likelihood function similar to that of MatrixREDUCE.

1.6 Objectives and achievements of this thesis

The objective of this thesis is to see if protein binding micro-array data, together with an alternative to the PWM model, can improve the performance of genomic binding site prediction. Our best proxy information on genomic binding sites at present is ChIP-seq data, therefore we will use ChIP-seq data from the ENCODE project as our gold-standard.

A binding site prediction algorithm would ideally tell us every position in the genome where a transcription factor binds and indicate none where it does not.[1] *In vivo*, other mechanisms that might affect binding site occupancy include,

- Epigenetic modifications can exclude DNA sequences. Chromatin structure can obstruct binding, and whilst the mechanism is unclear, the methylation of DNA also appears to play an important role in gene regulation [11].

[1] We might actually like to know the bound locations that have a functional effect, which might be a subset of those that are actually bound.

- Cooperative binding between several factors might be necessary to provide sufficient affinity [66].

These mechanisms could function independently of local DNA sequence composition. Although, local DNA sequence composition could still carry information on local chromatin structure and methylation.

In this thesis we restrict ourselves to trying to predict binding on a 'sequence specific basis'. We approach this task using protein binding micro-array data. This data provides unbiased sampling of all sequences up to a length of 10bp.

Next is a brief overview of the chapters that form the rest of this thesis. For each chapter a description of the motivation, objectives and achievments are given.

1.6.1 Chapter 2

The work described in chapter 2 is part of that done during the first year of a PhD program starting in 2009. This chapter uses direct measurements of binding affinities for the yeast transcription factor cbf1 made available in Maerkl & Quake [51].

1.6.1.1 Objectives

Modelling of transcription factor binding using biophysical principles could help improve our ability to predict binding locations and action.

The objective of this work is to show an improved fit to the experimental data using an improved model justified by biophysical principals. In particular a parameter is added to account for non-specific binding.

1.6.1.2 Achievments

The improved model improves the fit to the data. It was intended to publish this result but this was done elsewhere [67] before the investigation was finished.

The modelling of non-specific binding, and perhaps *diverse modes* of binding, is a subject that will feature later in this thesis (Section 5.5.1). Whilst only a single parameter is added in this chapter, adding further parameters in later

chapters enables us to capture more complicated non-specific binding and helps in ChIP-seq peak prediction.

1.6.2 Chapter 3

Chapter 3 introduces the protein binding micro-arrays that are the subject of much of the rest of this thesis.

1.6.2.1 Objectives

The protein binding micro-array data is noisy, and whilst methods have already been described to remove this noise, it was decided that a method to model the noise more carefully, including being able to visualise the noise, would help obtain the best possible signal from the arrays overall and from individual probes of interest.

The objectives of the work in this chapter are to create improved software and statistics for modelling the noise component of the data and a tool for the visualisation of this noise component and its removal. A completely automated approach to data-normalisation is desirable, but building an intuition for the data is also vital.

1.6.2.2 Achievments

A system that allows detailed visualisation and efficient manual curation of the data was constructed.

A novel application of B-splines to protein binding micro-array data is described and implemented. The efficiency and applicability of this approach to data of this scale and type is established. Performance of this approach is shown to be least as good at noise removal as several other methods.

An improved approach to quantile normalisation of saturated protein binding micro-array probes is described, implemented and shown to offer significant improvements over methods described elsewhere.

1.6.3 Chapter 4

Chapter 4 turns to the investigation of the protein binding micro-array probes and the construction of the 'model matrix' used in subsequent parts of this thesis. The protein binding micro-arrays have probes from either the HK design or the ME design. These designs are introduced in detail in chapter 4.

1.6.3.1 Objectives

A detailed understanding of the composition of the probe sequences aids the development of computational methods that allow efficient exploration of the data. More efficient computational methods allow a larger parameter space to be explored. Better understanding of the statistics of the sequences allows better insight into the behaviour of the model.

A detailed and mathematically complete description of the DNA sequences that make the probes on the protein binding micro-arrays is sought and also a comparison between the HK and ME designs. An efficient way of computing over a model of the probes is a further objective.

1.6.3.2 Achievments

Differences are shown between the probes of the ME array and the method published in Mintseris & Eisen [68]. The impossibility of the construction described in the publication is demonstrated for certain sequences. This result was the subject of a publication shortly after the completion of this thesis [69].

An algorithm for reverse engineering and hence the production of the HK design is described. The sequences on the HK array are verified to be from a generator polynomial in this design's case, the polynomial is derived.

A method to efficiently locate subsequences within probes, a novel application of a approach already known in the mathematical literature to this research area is suggested, though not implemented.

An efficient model matrix construction is described and implemented that allows the handling of parameter sets larger that suggested possible in previous publications [70].

Using the ability to over-parametrise, extended feature sets are suggested and constructed, including reverse complement sequences and probe position features.

1.6.4 Chapter 5

The learning of a predictor and prediction of readings from counterpart arrays presented in chapter 5 was first attempted as part of the DREAM5 challenge [71], see section 5.2 for a full background.

1.6.4.1 Objectives

The protein binding micro-array data from Weirauch *et al.* [71] allow an independent and non-biased set of sequences on which to train a predictor of transcription factor binding.

The objective of the work in this chapter is to build a new type of sparse predictor for each of the transcription factors available in the data set.

1.6.4.2 Achievments

A simple multiple sequence alignment approach shows a bias in the position of binding sequences within probes.

Predictions of counterpart protein binding micro-array intensities, using the sparse models, are shown to perform at least as well as those published elsewhere.

The sparsest models provide interesting observations, including evidence of multiple binding to individual probes, this has not been shown elsewhere.

Models restricted to short length DNA words are presented including surprising results on their performance as predictors. The substantial differences in the predictability of transcription factors using these reduced models offers new insight.

It is observed that reverse complement features appear to have little or no significant affect when added to the model matrix whilst probe position information does contribute to predictor performance. These properties of extended word models have not been described elsewhere.

1.6.5 Chapter 6

In chapter 6 the predictors trained in chapter 5 are used to predict the binding locations in genomic DNA.

1.6.5.1 Objectives

The ability to accurately predict the location of transcription factor binding would provide insight into gene regulation and disease aetiology. Relative performance of different models can suggest the validity of modelling assumptions.

In this chapter predictions of genomic binding locations, of the available transcription factors, is sought. A comparison between the performance of PWMs and the sparse word models developed in the previous chapter is sought. Models for three transcription factors have predictions assessed via ChIP-seq data from the ENCODE project. Mouse DNA is scanned and compared to the ChIP-seq gold standard.

1.6.5.2 Achievments

A naive and then refined ROC procedure is used to interpret the scan results effectively. The sparse models trained on micro-array data perform comparably to PWM models trained on ChIP-seq data. The results of this chapter are a useful proof of concept of a hard objective. Though a lack of high quality data prevents a thorough assessment or any definitive conclusions.

1.6.6 Chapter 7

A review of the previous chapters is given and some suggestions for further investigation are made.

1.6.7 Appendix A

Some pictures and descriptions of functions of a GUI data viewer are presented.

Chapter 2

Biophysical Binding Model

This chapter deals with direct measurements of physical quantities. We discuss the application of a simple biophysical model fitted to some experimental data for the yeast transcription factor cbf1. This investigation addresses the question of whether the modelling of a transcription factor with a position weight matrix can be justified given biophysical measurements.

In the background (Section 2.1) there is a discussion of early modelling of physical binding, the constants used and their units, and an introduction to some more recent measurements made using a micro-fluidic platform.

In the methods section the model and algorithm used to obtain an energy matrix for a yeast transcription factor are described.

In the results section the fitting of an energy matrix using micro-fluidic data is described.

In the conclusion section there is a discussion of the validity and usefulness of the biophysical models described here.

2.1 Background

2.1.1 Constants for the lac repressor

The lac repressor is probably the best studied DNA binding protein and is perhaps the archetypal, sequence specific, transcription factor. It continues to be the subject of quantitative research on protein DNA interactions [72, 73].

In Riggs *et al.* [16], estimates of the affinity of the lac repressor were made and given in terms of a Michaelis constant, K_m. Michaelis Menton kinetics, from which the constant gets its name, is a model used to describe the kinetics of enzyme substrate reactions. The Michaelis constant K_m is the concentration of a substrate at which the rate of production of the product will be at half maximum rate. Therefore the units of this constant are M. Riggs *et al.*'s early estimates of the affinity of the lac repressor to the lac operator was a K_m value of 2 to 4×10^{-10} M, and 2 to 4×10^{-9} M for non-operator DNA [16].

Association and dissociation constants are the ratios of on and off-rates of DNA-protein binding, one being the reciprocal of the other. For the lac repressor on and off rate constants are given as $1.0 \times 10^{10}\,\mathrm{M}^{-1}\,\mathrm{s}^{-1}$ and $1.0 \times 10^{-3}\,\mathrm{s}^{-1}$ respectively. These numbers imply that the half life for a bound lac repressor is about 20 minutes. The ratio of these constants agrees with the association constant given before. It is of note that the on rate is much faster than might be expected from simple diffusion [73]. The authors of this publication describe a motion of a DNA binding protein along a molecule that they describe as 'linear search'. A physical description of this type of interaction presumably requires more than a site-specific model of binding.

An important aspect of a protein's interaction with DNA is that of 'non-specific binding'. For the lac repressor this was addressed in Revzin & Von Hippel [74]. The lac repressor binds to non-operator DNA preferentially to single stranded DNA and the dissociation constant is estimated as $1.0 \times 10^{9}\,\mathrm{M}^{-1}$, its on rate as, $1.0 \times 10^{7}\,\mathrm{M}^{-1}\,\mathrm{s}^{-1}$ and off rate as $1.0 \times 10^{-2}\,\mathrm{s}^{-1}$, all at concentrations of 0.01 M Na^+. Binding constants to non operator containing DNA vary considerably depending upon base composition too; more AT rich sequences compete more effectively.

2.1.2 Micro-fluidic binding affinity measurements

When dissociation rates are high many assays struggle to measure the association constant. The micro-fluidic apparatus mechanically traps protein bound DNA to get around this problem [51]. The method's authors note the difficulty that

protein binding micro-arrays[1] have in this respect due to the washing that any bound protein has to endure [51]. Off rates of bHLH family proteins are above $10\,s^{-1}$. The authors describe how a series of concentrations are used to estimate dissociation constants for the protein's interaction with 2400 DNA templates. Dissociation constants, K_d are related to Gibbs free energy via the expression,

$$\Delta\Delta G = RT \ln \frac{K_d}{K_{ref}} \tag{2.1}$$

where K_{ref} is set to the dissociation constant of the highest affinity binding sequence.

2.1.3 An energy matrix for a physical binding model

It was described in the introduction how we can use a rectangular array of numbers to describe changes in Gibbs free energy corresponding to changes in nucleotide composition of a DNA-protein interaction site (Section 1.4.5).

An analogy between position weight matrices and energy matrices is often made and an example of this is the figure 2.1 taken from Maerkl & Quake [51]. Maerkl & Quake [51] acknowledge that non-specific binding is partly responsible for the apparent failure of their PWM to predict the experimental results for large values of $\Delta\Delta G$. Our method to account for the non-specific binding, which is the main subject of this chapter, is given in section 2.2. The idea behind this method is to classify binding as one of the following,

Specific binding

In this case the protein has assumed an energetically favourable conformation closely interacting with a specific sequence on the DNA double helix through hydrogen bonds.

Non-specific binding

[1] This, along with variation in non-specific binding, is of relevance to later chapters in this thesis though it is not discussed much further due to limitations in what was achievable with the protein binding micro-array data that was available. These limitations are discussed further in later chapters.

In this case the protein is attracted to the DNA in a non-specific manner and can be thought to move along the double helix 'searching' for a sequence with which it may interact in a more energetically favourable way. The type of bonding is a non-sequence specific electrostatic interaction.

The probability of the protein being in either of these states, given that it is in one of them, is then,

$$\mathbb{P}(\text{Specifically bound}) = \frac{e^{-E_{sp}(\mathbf{a})}}{e^{-E_{sp}(\mathbf{a})} + e^{-E_{ns}}} \tag{2.2}$$

$$\mathbb{P}(\text{Non-specifically bound}) = \frac{e^{-E_{ns}}}{e^{-E_{sp}(\mathbf{a})} + e^{-E_{ns}}} \tag{2.3}$$

where E_{ns} is the non-specific binding energy and $E_{sp}(\mathbf{a})$ is the specific binding energy and is a function of the sequence $\mathbf{a}$. $E_{sp}(\mathbf{a})$ could be any function of $\mathbf{a}$ but we will restrict ourselves to an additive model,

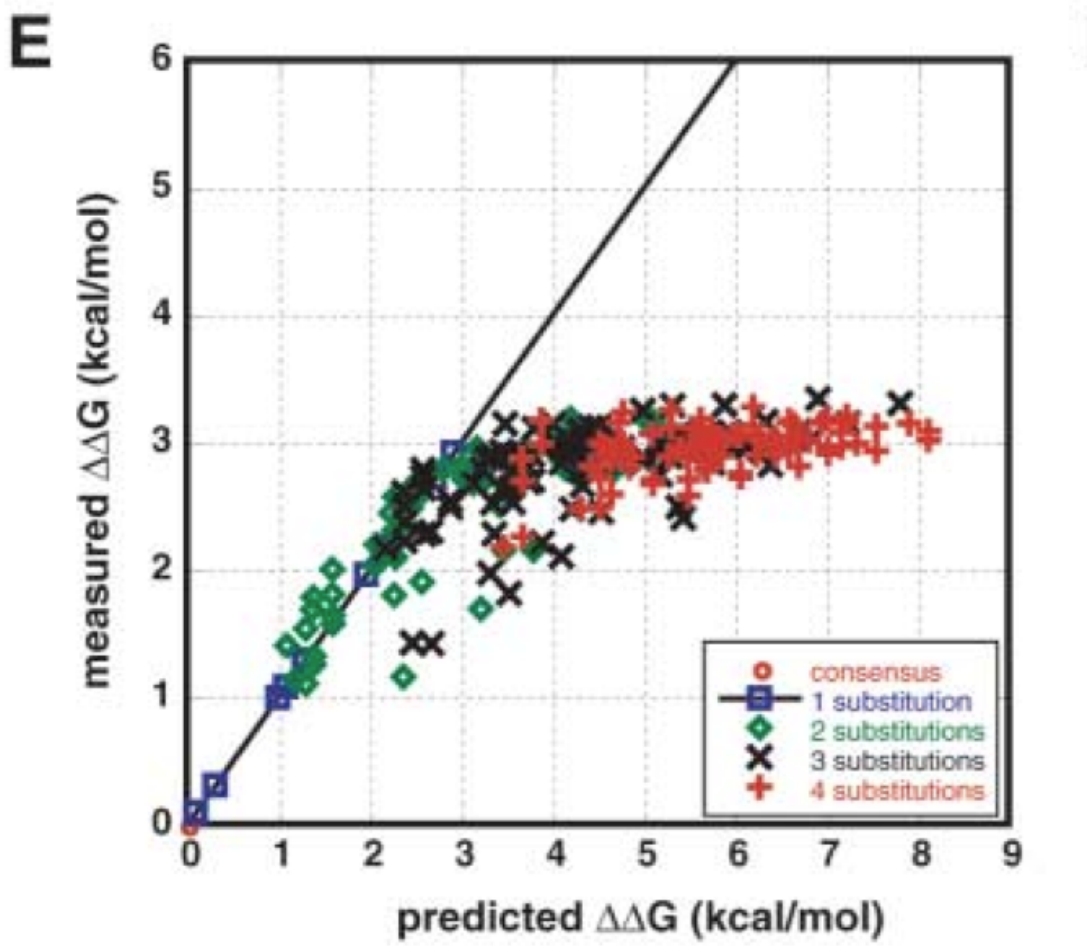
$$E_{sp}(\mathbf{a}) = \sum_{i=1}^{n} \epsilon_i(a_i) \tag{2.4}$$

The binding affinity of any n-mer sequence can then be calculated using the $4 \times n$ parameters $\{\epsilon_{Ni}\}$where $N \in \{A, C, G, T\}$ $i = 1, 2, \ldots, n$ and the parameter E_{ns}.

2.1.4 The Cbf1p DNA binding protein

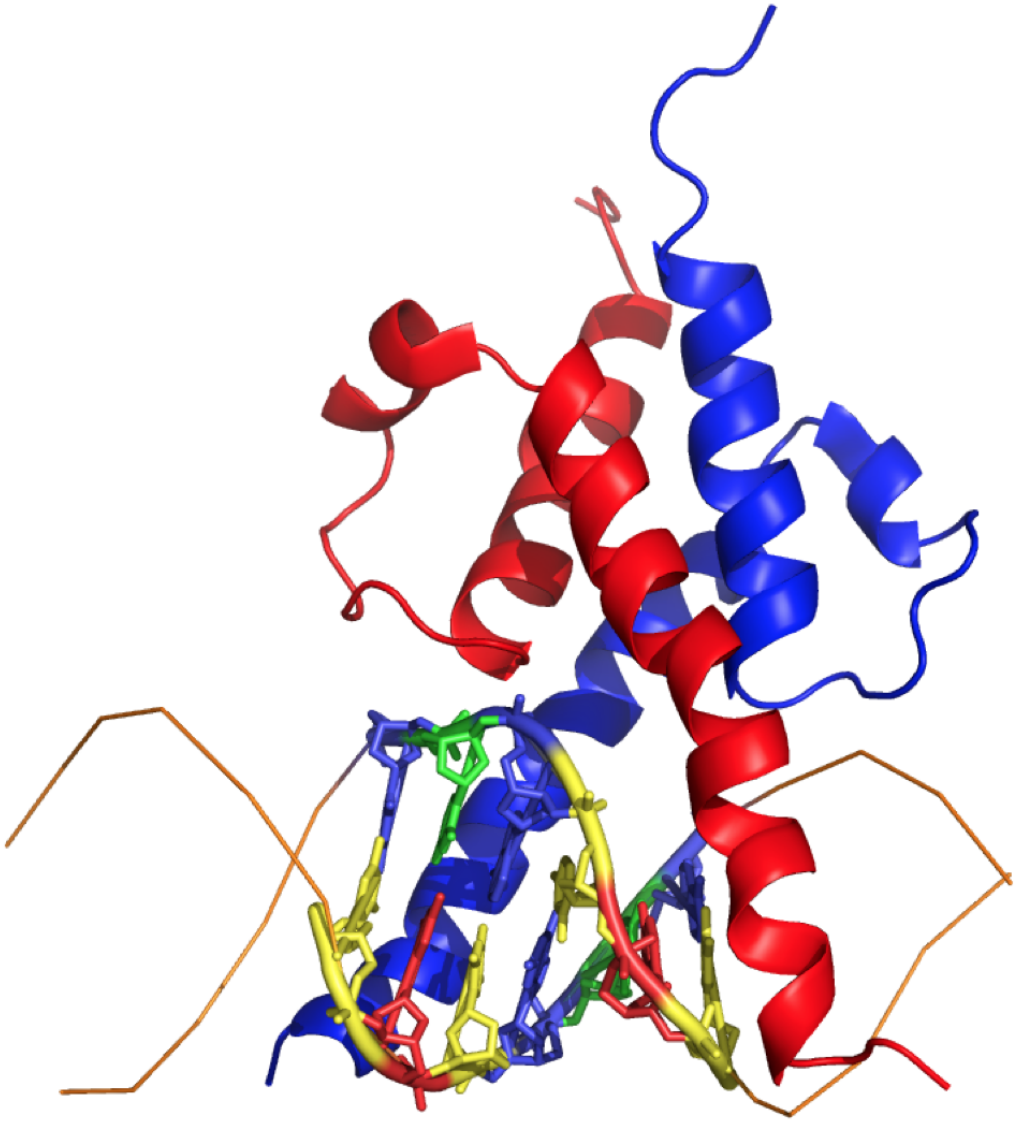
Cbf1p is a member of the basic helix-loop-helix, (bHLH), family of transcription factors, (TFs). It is known to bind to the centromeric DNA of *S. cerevisae* and plays an important role in chromosome segregation. It also has a role in the regulation of biosynthesis of L-methionine and is part of other transcription activation complexes [75]. In common with other bHLH TFs, basic residues on the α-helices make contact with the major groove of the DNA double helix (Figure 2.2). The 6-tuple consensus sequence is CACGTG and is known as the E-box. The TF Pho4p is a similar bHLH protein but regulates distinct pathways; binding is thought to be prevented by a thymidine residue immediately flanking the

Figure 2.1: This image is taken from Maerkl & Quake [51] and shows an application of a matrix model to binding energy. It is observed that for large $\Delta\Delta G$ the modelled values exceed the measured ones. It should be noted that this plot is for the MAX transcription factor, the equivalent plot for Cbf1p was not given in the publication. The purpose of this illustration is to show the disparity between the data and the type of model used in the paper.



E-box in this case. It is of interest to obtain an accurate and sensitive model of the binding preferences of Cbf1p.

Figure 2.2: Structure 1A0A from the PDB is the close relative Pho4 of Cbf1. This homodimeric protein interacts with its target DNA making contact with the core E-box recognition sequence, (the consensus CACGTG).



2.2 Methods

2.2.1 Fitting the energy matrix model

Binding affinity measurements are available from a micro-fluidic experiment that tested Cbf1p's interaction with all permutations of certain oligonucleotide templates [51].

Raw data is available from the url[1] and the templates were,

TTGT CACNNN ACTT (1)

TTTT CACGTG NNNT (2)

The positions marked with Ns are independently permuted over all possible nucleotides giving a total of $4^3 + 4^3 = 128$ possible patterns, the spaces demark the central E-box 6-mer. The binding affinity of the protein to CACGTG is set to have a baseline energy of zero. For this reason, and working under the assumption of independent contributions from each nucleotide, we may add the energy contributions from each pattern to obtain energies for all possible 4^6 patterns of the form CACNNN NNN, (starting from the left side of the E box).

The reverse complement of this pattern gives us all patterns of the form, NNN NNNGTG. In this way we have an energy reading for each of the 512 possible 12-mer patterns centered on the E-box, (again by adding the contributions from each of the four contributing patterns). The best binding energy will then be zero and any deviations from this 'consensus' will have higher energy.

For instance, to obtain an energy for the pattern AAC CGG TTA ACC we would find the reverse complement of the first two triples, AAC and CGG, from patterns 2 and 1 above, in that order:

TTTT CACGTG GTTT (2)
TTGT CACCCG ACTT (1)

[1] http://lbncm1.epfl.ch/twiki/bin/view/MaerklLab/Projects/BindingEnergyLandscapes

The energies of these two give us the energy contribution of the first half of our pattern.

For the third and fourth triples of our pattern, TTA and ACC, we can add the energies from patterns 1 and 2 above without making any transformations. i.e. we would use,

TTGT CACTTA ACTT (1)
TTTT CACGTG ACCT (2)

to find the energy contribution of the second half.

For any 12-mer, $\mathbf{a}$, we now have the experimentally derived energy, $E_{data}(\mathbf{a})$. This energy is a measurement of both specific and non-specific binding and since we are primarily interested in a model of sequence specific binding we must separate the contributions from each. Each of the energy states, specific and non-specific, are modelled as a canonical ensemble of states at equilibrium. Considering the partition function[1] leads us to the following decomposition,

$$e^{-E_{data}(\mathbf{a})} = e^{-E_{sp}(\mathbf{a})} + e^{-E_{ns}} \tag{2.5}$$

which is the denominator in equations 2.2 and 2.3.

To learn the parameters ϵ_{Ni} and the single parameter E_{ns} (Equation 2.5) a C/C++ language implementation of the Levenberg-Marquardt algorithm was used [76]. This non-linear least squares minimisation routine returned optimal values that were consistent across 1000 random starting points. The objective function is the mean squared error between the measured energies and the model (Equation 2.5) parametrised over $\mathbf{a}$,

$$\sum_{\mathbf{a}\in\{A,C,G,T\}^6} (E_{data}(\mathbf{a}) - \log(e^{-E_{sp}(\mathbf{a})} + e^{-E_{ns}}))^2 \tag{2.6}$$

Exploiting the symmetry in the problem we in fact only need to learn parameters

[1] the partition function is motivated as the maximum entropy distribution of states having a given mean.

for a 4×6 matrix that we can afterwards transform and append to create the final 4×12 matrix. It is worth noting that learning two 4×3 matrices independently, one for each of the patterns, and afterwards appending one to the other, would not be satisfactory since the non-specific parameter must be the same in both cases.

2.3 Results

The learned energy matrix is as follows,

$$\begin{pmatrix} 1.44 & 1.22 & 3.14 & 0 & 0.38 & 0.37 \\ 2.12 & 1.07 & 2.20 & 0.37 & 0.04 & 0 \\ 0 & 1.35 & 0 & 0.71 & 0.25 & 0.16 \\ 3.06 & 0 & 1.60 & 1.00 & 0 & 0.09 \end{pmatrix}$$

The non-specific parameter was simultaneously learned as stated in the methods section (Section 2.2.1). Since it is a constant value on an arbitrary scale it is not included here.

The most energetically favourable nucleotide is read from the row containing 0 in each of the columns. The most energetically favourable sequence is GTG ATC. We use the reverse complement to obtain the most energetically favourable 12-mer sequence, GAT CAC GTG ATC. Figure 2.3 shows the fit when using a position weight matrix with an assumption of independent columns, this is analagous to figure 2.1. Figure 2.4 shows the close agreement between the energies as measured and those using the additive model with non-specific binding parameter. This tells us the additive model is a good fit to the data. A more complicated, non-additive, model involving di-nucleotide correlations might not be justified in this case.

In contrast to figure 2.1 we see good agreement between values from our energy matrix model and the experimental data.

Figure 2.3: Correlation between actual and modelled values for energy. Energies from each of the templates are colour coded and a line of best fit is drawn. The correlation coefficient is 0.890. This graph is analagous to figure 2.1

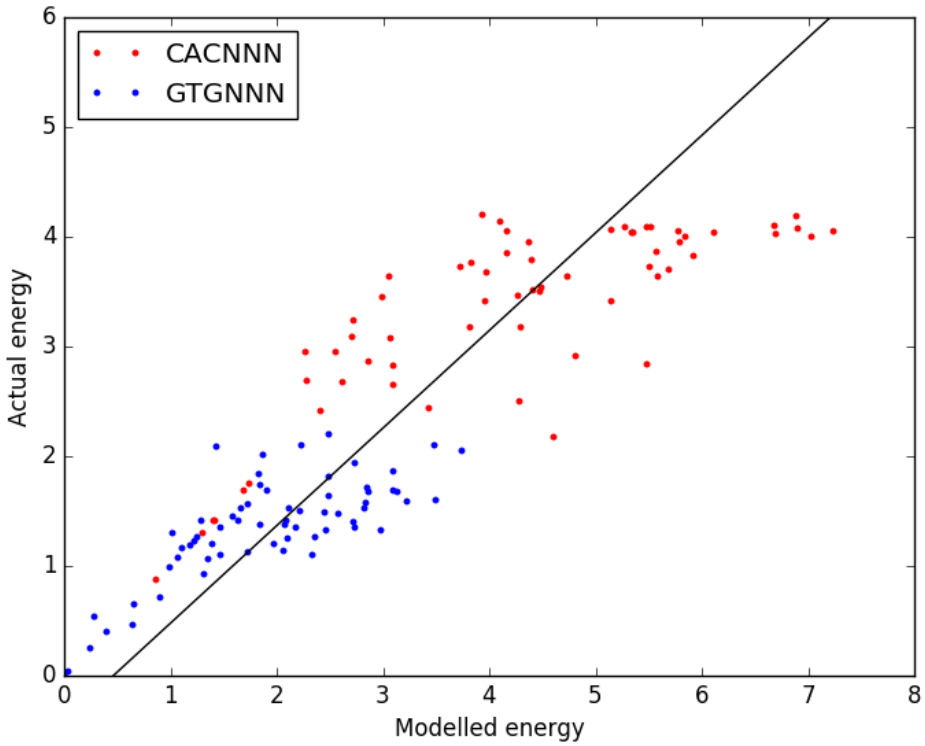
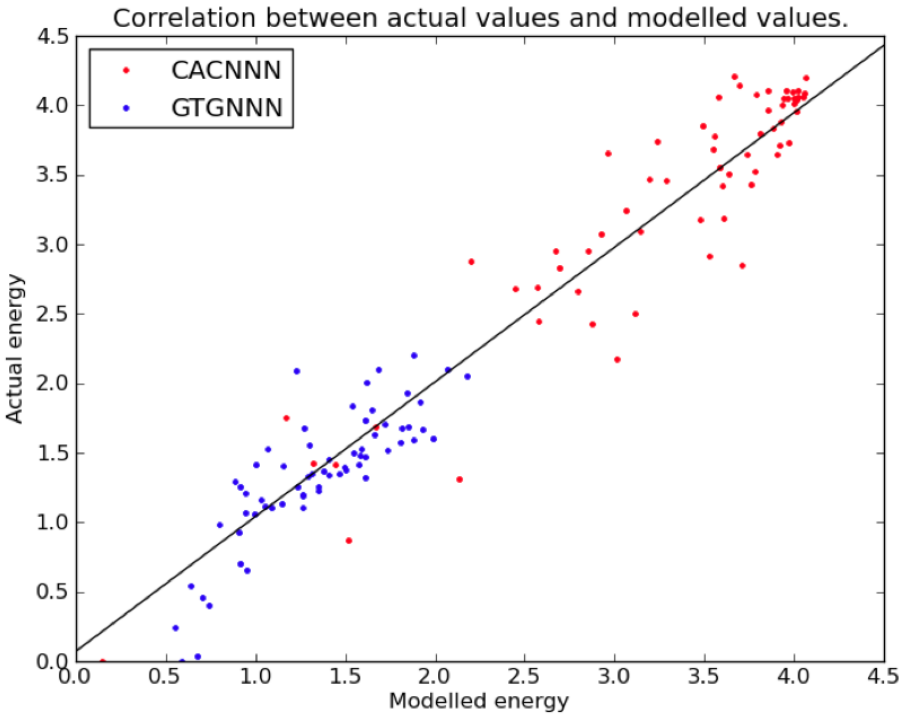


Figure 2.4: Correlation between actual and modelled values for energy. Energies from each of the templates are colour coded and a line of best fit is drawn. The correlation coefficient is 0.969. This fit, using the extra non-specific binding parameter, is clearly improved for the higher energy values.



2.4 Conclusions

In this chapter a method to model physically measured association constants using an additive model has been demonstrated and was shown to be a better fit for the data than a simpler approach made in the paper accompanying the original data.

The model presented uses dissociation constants obtained from experiments that use a series of concentrations in order to determine them. Once we have these constants the model presented in equations 2.2, 2.3 and 2.5 appears to work well in fitting the data.

Given the success of this approach we would like to apply this model to the protein binding micro-arrays that are described later in this thesis but it is not clear how to obtain the dissociation constants from the protein binding micro-array data. Repeated experiments at different concentrations were not available. The scale and locations of intensity distributions appear to change significantly under replicate experiments at apparently the same experimental conditions and so it seems unlikely that robust parameters could be obtained.

Chapter 3

Array Data Normalisation

As stated in section 1.6, the objective of this thesis is to examine whether the use of data that better captures DNA protein interactions can be used to construct better predictive models. Starting with this chapter, a set of PBM datasets are described along with methods to normalise their data. The objective in this chapter is to evaluate different approaches and select the best to curate a dataset that is used in the next chapters.

In the background (Section 3.1) there is an introduction to the 'DREAM5 data-set'. This data-set is the focus of much of the rest of this thesis. In section 3.1.1 details are given of the type of protein binding micro-array experiment whose data comprise the DREAM5 data-set. An outline of some important steps in the protocol are given. In section 3.1.1.3 some closer attention is given to the *in situ* double stranding stage of the protocol. This stage appears to be one of the more critical. The quality and origins of the DREAM5 data-set will be discussed along with some of the consequences. The presence of spatial artefacts is discussed in section 3.1.2. The lack of availability of certain desirable data is talked about in section 3.1.3.

Section 3.2 is the methods section. First, in section 3.2.1, there is a discussion of the initial stage of the data normalisation process; the exclusion of outlying data. In the same section, a tool developed to visualise the protein binding micro-array data is introduced and its utility for the specific task of outlier exclusion is shown, there are further details in the appendix. In section 3.2.2, details are presented on the modelling of, and compensation for, spatial artefacts. The

DREAM5 data-set has clear evidence of spatial artefacts. These artefacts can be seen upon visual inspection of the distribution of intensities across their grid layout. Saturation at the highest intensity readings can also be observed on several of the DREAM5 micro-arrays. In section 3.2.3, methods are discussed that were used to deal with the saturation of signals at high intensities.

Section 3.3 is the results section. In section 3.3.1, measurements that assess the performance of spatial artefact removal are presented. In section 3.3.2, some measurements to assess the benefit of the correction of saturation artefacts are given.

In the conclusion (Section 3.4) there is a discussion of the usefulness of the techniques described in the preceding chapters.

3.1 Background

The DREAM5[1] protein binding micro-array data-set was made available by the organisers of the DREAM5 conference [71]. In aggregate, it consists of data for 172 protein binding micro-array experiments. Half of the experiments are performed on an array with a set of probes from one de Bruijn sequence, hereafter referred to as the 'HK design'. The other half of the experiments were performed on arrays with a set of probes taken from a different de Bruijn sequence, hereafter referred to as the 'ME design'. Over the 172 arrays a total of 82 proteins had binding characteristics measured. Every protein was characterised on each array design, HK and ME. Only a handful of proteins were afforded replicate experiments on the same array design. Two proteins, were afforded replicate experiments and a single protein was afforded triplicate experiments. Distributions of the latter are shown in figure 7.4. The aggregate of 172 protein binding micro-arrays is what will be referred to as the DREAM5 data-set for the rest of this thesis.

[1] DREAM is an acronym for 'Dialogue for Reverse Engineering Assessments and Methods'. The number 5 refers to it being the fifth such competition bearing this name. The evaluation meeting took place in November of 2010 in New York.

3.1.1 The protein binding micro-array experimental protocol

The type of protein binding micro-array technology under discussion here is an adaptation of gene expression array technology and was first described in Bulyk *et al.* [54].

A protocol for a protein binding micro-array experiment is detailed in Berger & Bulyk [53]. The protocol described there is not assumed to be significantly different from the protocol used for the production of the DREAM5 data-set.

3.1.1.1 Protein binding micro-array probe specification

The same Agilent $4 \times 44K$ oligonucleotide arrays, with customer specified probe sequences, are used for all experiments [77]. These single stranded oligonucleotide arrays have their single stranded DNA probes converted to double stranded probes in a solid phase primer extension reaction. The probe spot diameter is approximately 50 µm. The length of a probe is 60bp and so would be approximately '20 nm high' above the array [78]. Our picture, therefore, should be of each spot on the protein binding micro-array being a lawn of double stranded DNA probes.[1] Each probe contains a segment of a de Bruijn sequence. De Bruijn sequences and their application to protein binding micro-arrays is discussed at length in section 4.1.1.2.

The primer extension reaction used in creating the double stranded probes from the single stranded oligonucleotide arrays is described as being carefully optimized and yet remaining sensitive to temperature [53]. It is stated by the designers of the protocol that "it is important to monitor the accuracy of *each* primer extension reaction before using a micro-array in a protein-binding experiment"[2]. Unfortunately, the necessary data to perform or refine this step is mostly unavailable in the DREAM5 data-set, i.e. for only 8 of 172 arrays. Further discussion is in section 3.1.1.3.

Imaging of the array after the DNA binding protein has been added is done

[1] An analogous and well mown lawn, with blades of grass half a centimeter high, would have a diameter of about 12.5 meters.

[2] Emphasis added here.

using a sequence of increasing laser intensities. This sequence of measurements is combined using an algorithm 'Masliner' [79]. Unfortunately, the raw data to perform or refine this procedure is not available for the DREAM5 data-set. It is actually unclear whether this step has been performed. If this integration has not been performed, it is unclear which of the sequence of laser illuminations was chosen.

3.1.1.2 Preparation of the DNA binding protein

To investigate a DNA binding protein's affinity, the authors of the protocol [80] suggest it is adequate to clone only the binding domain of the protein. This amino acid sequence is combined with a GST tag that will later pair with the fluorescent Alexa 488 conjugated anti-GST. The authors suggest that purification from cellular lysate[1] is not always necessary since only the tagged proteins will emit a signal when excited by the laser illumination.

The authors state they have observed no difference between *in vitro* translation and the use of clones expressed in *E. coli*, and that the former allows the generation of more data, and so this is their preferred technique. It is stated that a few hundred nanograms of protein is sufficient and its concentration should be accurately estimated by Western blot or another method.

The authors of the protocol say that they use standard concentrations of protein i.e. $100\,\mathrm{nmol\,l^{-1}}$. The authors suggest that using rank based statistics makes the standardisation of protein concentration and salt concentration less important. They qualify this by stating that this might not be the case if a protein is 'particularly sensitive' to these parameters or to the presence of co-

[1] After over-expression in *E. coli* for instance.

factors[1].

3.1.1.3 Measuring the success of *in situ* double stranding through Cy3 incorporation

Each protein binding micro-array probe is attached to a glass slide. The locations of the spots on the glass slide are determined by inkjet printing initial nucleosides onto the slide. The initial nucleosides create an -OH bond with the slide substrate [77].

At the attached end of the 60bp probe a 25bp primer sequence is synthesised. This sequence is 25 base pairs long and is as follows,

```
CCTGTGTGAAATTGTTATCCGCTCT
```

It is the same on both array designs, i.e. the HK design and the ME design (Sections 4.1.1.3 and 4.1.1.4) and for every probe. The subsequent 35bp of a probe is specified uniquely for each array design.

The protocol dictates the addition of a small quantity of Cy3-conjugated dUTP to the primer extension reaction. The purpose of this is to indicate the quality of the double-stranding process. The amount of Cy3 incorporated into a double stranded probe should be in proportion to the number of adenine residues in the template strand. It was determined that the sequence context of an adenine residue on the template strand had a significant effect on the measured Cy3 signal. A linear regression upon all triplets was performed to create a model of expected Cy3 intensity for any given probe. A natural question that arises is,

[1] A simple biophysical model of binding says that the probability a probe is bound by a DNA binding protein, any given instant, depends upon the concentration of the protein compared to the concentration of potential binding sites. The experimental protocol here appears to dictate that high concentrations of protein compared to potential binding sites be used, i.e. there should be many protein molecules per array probe. The suggested importance of measuring protein concentration, and its standardisation at a fixed value, appears to be motivated by a desire to ensure a high enough concentration whilst using a minimal quantity of protein. This approach appears to be motivated by the desire to create a catalogue of protein binding micro-array data for a large number of DNA binding proteins. In this case the standardisation of such parameters seems sensible and perhaps necessary for reasons of economy.

does the sequence context of adenine residues determine measured Cy3 intensity by,

- affecting the proportion of Cy3 labelled dUTP *vs.* dTTP being incorporated into the probe, or,
- affecting the variation in measured Cy3 intensity due to, a more problematic, failure of the double stranding process.

An experiment to address this question was described in same paper as the description of the protocol [53]. Probes were prepared with a Zif268 binding site furthest from the array surface. The probe segments between the binding sites and the array were then varied in their di-nucleotide content whilst keeping mono-nucleotide proportions constant. 20 such probes were prepared and replicated on an array 16 times each. Since the measured amounts of Zif268 binding remained comparable, the authors concluded that it was incorporation of dUTP *vs.* dTTP, rather than failure of the double stranding process, that created the observed sequence dependencies of Cy3 signal. Despite this, the authors suggest that if a probe's measured Cy3 signal is above twice or below half its fitted value then it should be excluded from later analysis.

It is also suggested that "a run of five or more consecutive guanines are deleterious for primer extension reactions". The workaround for this is to replace each probe sequence containing such runs of guanines with its reverse complement. Whilst it is not mentioned in the protocol text it is clear that the probe sequence containing the 10-mer GGGGGCCCCC will pose a problem for this approach. This is discussed further in section 4.1.1.2. The relative merits of the different array designs are discussed in sections 4.1.1.3 and 4.1.1.3.

The DREAM5 data set includes information on the Cy3 readings for only[1] 8 of the 172 protein binding micro-arrays. Therefore the direct modelling of Cy3 effects has not been attempted in work described hereafter.

[1]It is unclear how the subset was chosen, it does not appear to be randomly selected though.

3.1.2 The presence of spatial artefacts

The grid layout of probes for each of the two micro-array designs is available for the DREAM5 data-set. The basic layout dimensions are given in table 3.1. Every probe has a location specified on a grid by pairs of integer indices. Figure 3.5 is an image of a section of a protein binding micro-array made after the laser excitation of fluorophores. This shows the regularity of the grid of spots and the distances between spots. Similar images for the protein binding micro-arrays in the DREAM5 data-set are not available. Figures 3.1, 3.2, 3.3 and 3.4 show

Table 3.1: Grid layout of protein binding micro-arrays.

Design	Number of Probe Spots	Empty spots	Spots up	Spots across
HK	40630	300	478	85
ME	40630	104	478	85

images that represent the sorts of spatial artefacts that can be seen for the protein binding micro-array foreground intensity readings. The images have been made by giving a colour to each probe according to its intensity reading and plotting that at its grid location. Each grid location is shown as a single pixel, though this may not be an accurate spatial representation in terms of proportions of the actual array. Pixels are displayed on a rectangular grid whereas a hexagonal grid is observed in figure 3.5. If the spots on the glass slide are distributed uniformly with gaps between spots of the same order of magnitude as the spot width, i.e. approximately 50 µm, then the images should be a reasonable representation. In any case, the presence of spatial artefacts is clearly observable. Stripes that run the length of the array are observed frequently, these are usually parallel to the edges of the array. Larger depressed or elevated regions can be observed, these are seen as broad regions of different colour. These regions may take up half of an array, perhaps with a gradient from one end to the other. Often the corners or ends of arrays have elevated or depressed regions.

The colour gradients in figures 3.1, 3.2, 3.3 and 3.4 have been chosen to highlight any spatial patterns. It should be noted that the range of probe intensity values that causes a spatial pattern in one of these images might be within the noise threshold of the array. That is to say, modelling and compensating for

these spatial patterns is not necessarily important given particular research objectives. Where curved lines are observed it could be speculated that uneven drying during one of the steps in the experimental protocol is to blame. For the straight edged spatial features it seems possible that the normalisation algorithm, 'Masliner' [79] could have introduced artefacts, i.e. when combining images from illumination with different laser intensities. It is unfortunately not possible to test these hypotheses given the absence of the raw data.

The pseudo randomness of the de Bruijn sequences from which the probes are constructed, together with the heterogeneous nature of the spatial artefacts, strongly suggests that the patterns observed are not related to the sequence content of the probes as arranged across the arrays[1].

It certainly seems reasonable that we should seek to remove any spatial biases, wherever possible, since they are unlikely to offer any information on the sequence specificity of the DNA binding protein being tested.

3.1.3 Availability of extra probe intensity information

The reading given for probe intensity is not raw data, it is a statistic obtained from a collection of data points for each probe spot on an array. The raw data would be the values from the imaging device that would comprise some number of pixels for every probe spot. A close look at figure 3.5 shows variation in pixel colour within array spots. This is the pixel data with which the foreground and background measurements are calculated. A selection of central pixels is used for the foreground signal and surrounding pixels are used for the background signal. Both mean and median statistics are available.

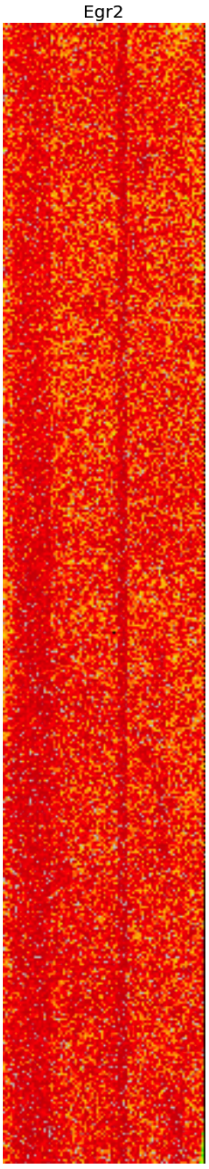
Of the 172 protein binding micro-arrays in the DREAM5 data-set only 106 have the foreground and background signal intensity readings available[2]. Unfortunately, raw images, similar to those in figure 3.5, are not available. Extra information would likely have been obtainable from these images.

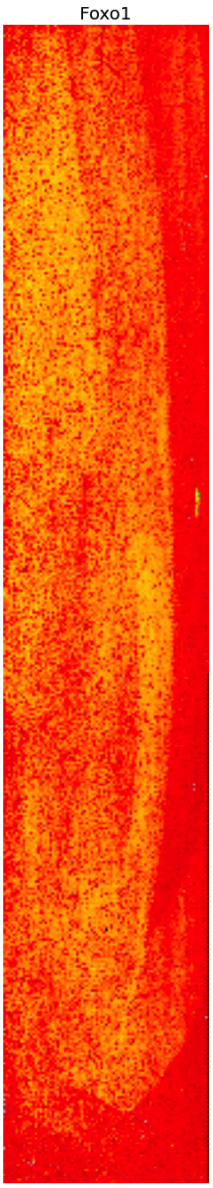
[1] Neighbouring probes are not neighbouring segments of the de Bruijn sequence from which they are taken. They appear to have been randomised over the layout as one would expect.

[2] The background signal intensity readings are missing for 66 of the arrays since the data for these 66 arrays was provided only as the 'correct answers' for the foreground signal prediction challenge.

Figure 3.1: Images constructed using a single pixel for each array spot. Spot intensity is shown using a colour gradient.

(a) A narrow, dark stripe parallel to the array edges together with a broad region to the left of the image that is consistently depressed relative to the rest of the array.

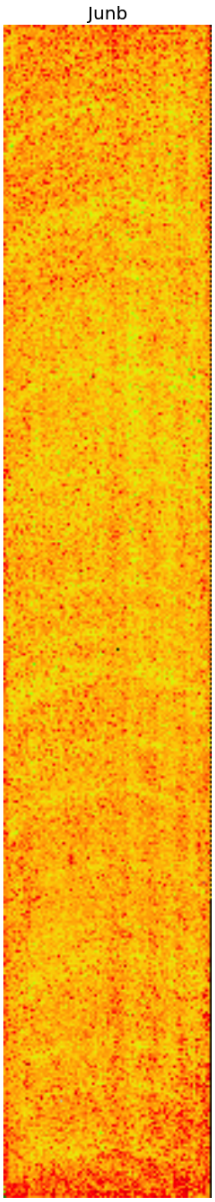


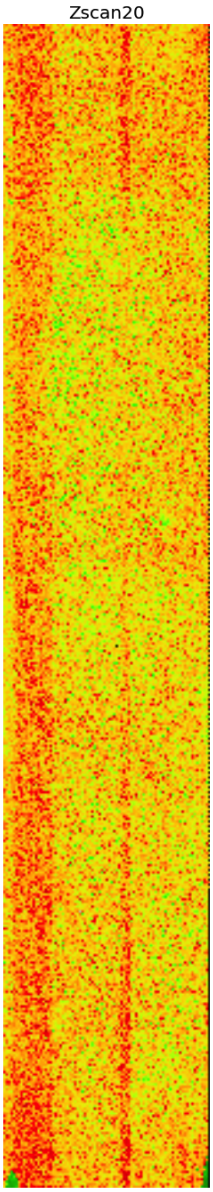


(b) Patterns of higher and lower intensity are observed that follow curves that are suggestive of artefacts formed in a liquid phase of the experiment.

Figure 3.2: Images constructed using a single pixel for each array spot. Spot intensity is shown using a colour gradient.

(a) Arcs are seen perpendicular to the long axis of the array. These arcs are reminiscent of a wave perhaps created in a viscous stage of the arrays drying process.

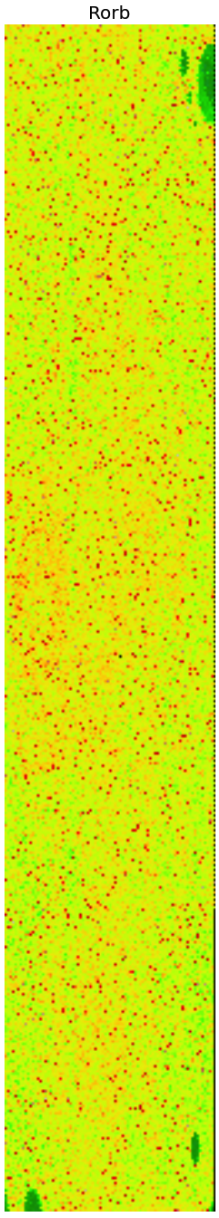


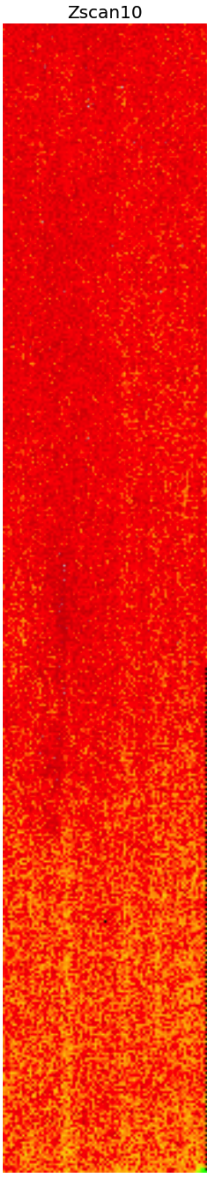


(b) A combination of artefacts including spots in the lower corners of the array.

Figure 3.3: Images constructed using a single pixel for each array spot. Spot intensity is shown using a colour gradient.

(a) Dark green spots are regions of low intensity.

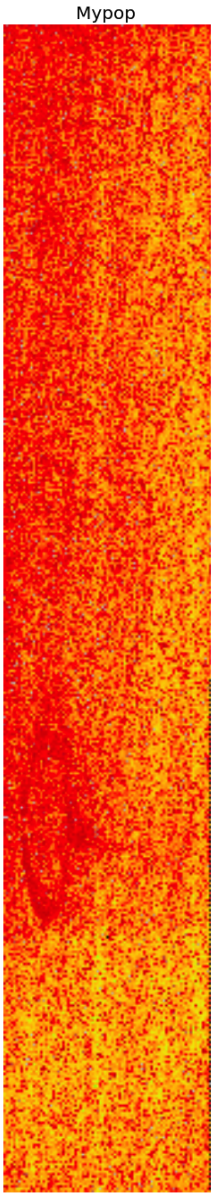


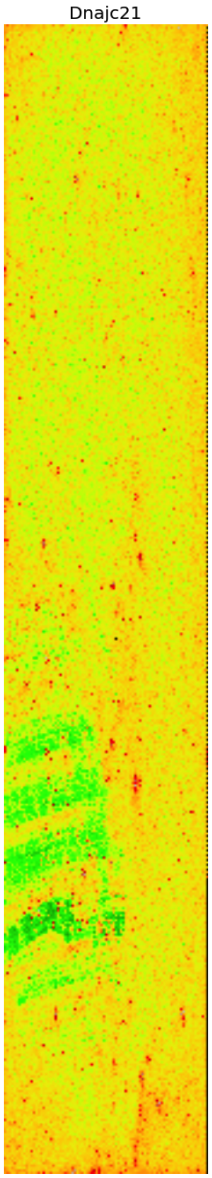


(b) The bottom third of this array has a consistently lower signal that the top two thirds.

Figure 3.4: Images constructed using a single pixel for each array spot. Spot intensity is shown using a colour gradient.

(a) An elliptically shaped artefact and also broad difference in intensity from the top left to bottom right of the array.





(b) This array shows some unusual patterning that might perhaps be better explained by physical contact rather than liquid drying phenomena as elsewhere.

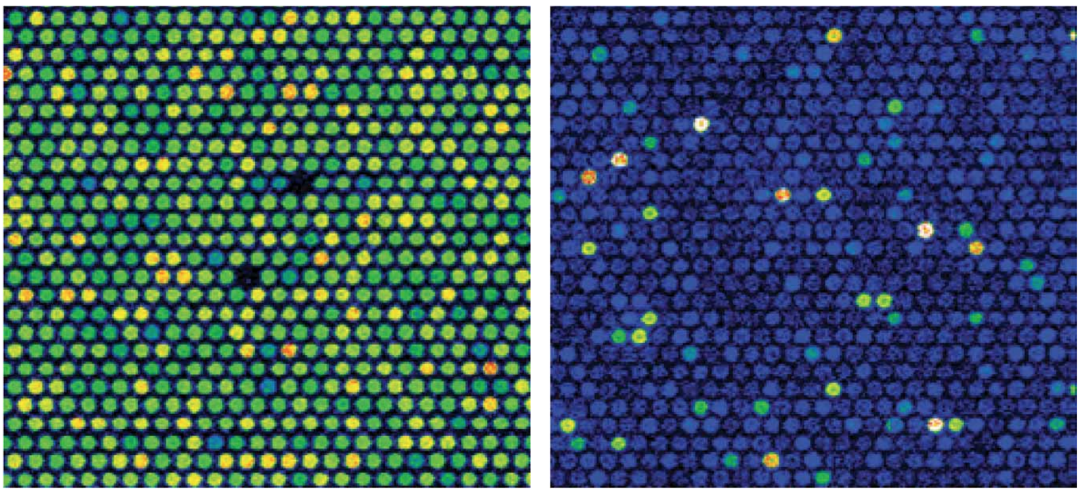


Figure 3.5: Digital images of protein binding micro-array spots after laser excitation of Cy3 and Alexa488 fluorophores. Taken from Berger & Bulyk [53].

3.2 Methods

A number of methods that have been developed to improve data normalisation and are described next.

3.2.1 Flagging of outlying probes

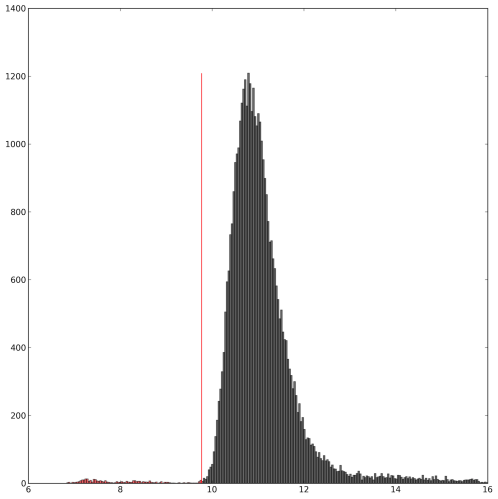
For several arrays there are probes that appear to be part of low intensity, spatial, artefacts. A method to exclude these is described by Annala *et al.* [70]. This paper accompanied an entry to the DREAM5 prediction challenge that produced the best predictions over an average of 6 metrics.

This method excludes probes, lying in histogram bins, that are to the left of the mode of the histogram (Figure 3.6). The mode of the histogram of intensities is located and then, after moving backwards towards lower intensities, a vertical line is drawn where the count has dropped below 0.005 of the peak count. All probes with lower intensity than this are excluded. This approach seems to work well in practice but does not give any intuition as to what sort of spatial artefacts might be being excluded. It is possible to gain an appreciation of this using the 3D data visualisation tool that was created as part of this work (Figure 3.7). The

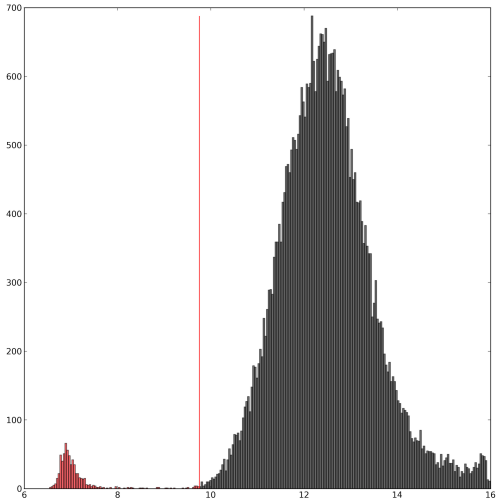
red plane is adjusted so that points to be excluded are below it and these are then flagged in the data base by clicking in a GUI, see Appendix for more information on this. The 172 arrays in the DREAM5 data set all had low intensity 'clouds' of points removed using the 3D viewer. In each case, the values at which the cut-off plane was set were compared to the values determined by the algorithm in Annala *et al.* [70] and were similar. The latter method often excluded a little more data from the low tail of the distribution but the differences are small. It is also worth noting that a bi-modal distribution could cause a blind application of the algorithm to go badly wrong, potentially excluding a large fraction of the data. It is necessary to look at the data.

Figure 3.6: Images showing low intensity probes excluded by method from Annala *et al.* [70].

(a) Low intensity probes excluded from Rorb protein binding microarray.



(b) Low intensity probes excluded from srebf protein binding microarray.

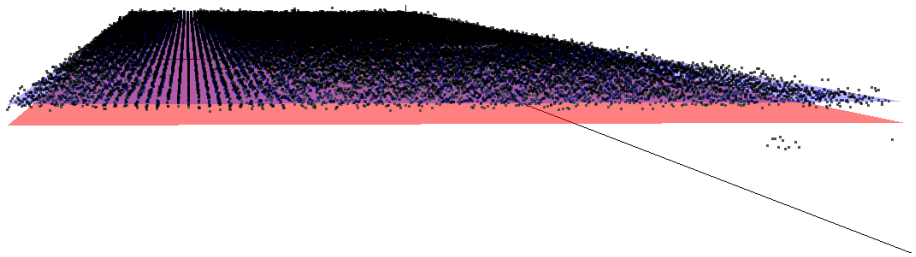


3.2.2 Methods to remove spatial artefacts

3.2.2.1 A moving window method to correct spatial artefacts

A method employed on the DREAM5 data-set was a 7×7 moving-median-window smoothing algorithm. This is computationally simple and has a positive effect on

Figure 3.7: 3D view of Sox14 protein binding micro-array. Some outlying values can be observed to the lower right corner in this view of the point cloud. The low intensity outliers can be clipped using the red plane and flagged in the sqlite database that backs the visualiser from this view.



the performance of a predictor, subsequently learned, for several DREAM5 arrays. A full list of before and after measurements is given in the supplementary material of Annala *et al.* [70]. Figure 3.8 is taken from this publication and illustrates the details of this method. A suite of software for protein binding micro-array processing, (perl scripts), is available from the Bulyk lab [53]. Amongst the routines is what appears to be an analogous moving window method.

Probes that are within three rows or columns of the edge of an array are corrected by the same amount as their nearest neighbour towards the interior of the array. Probes that have fewer than 25 other probes within their 7×7 moving window are excluded.

This method was re-implemented[1] and used to generate statistics for every array in the DREAM5 data-set. Two potential shortcomings for the moving window method are,

- the effect on values at the edges does not use information in one of the most

[1]The perl routine runs to 160 lines of code and takes a little under 30 s to complete the normalisation of a single array on a 2009 Apple Mac Book Pro (MBP). An implementation in the C++ language, using the Eigen linear algebra library [81] uses 60 lines of easily understandable code and takes 30 s to process 66 arrays in the DREAM5 data-set.

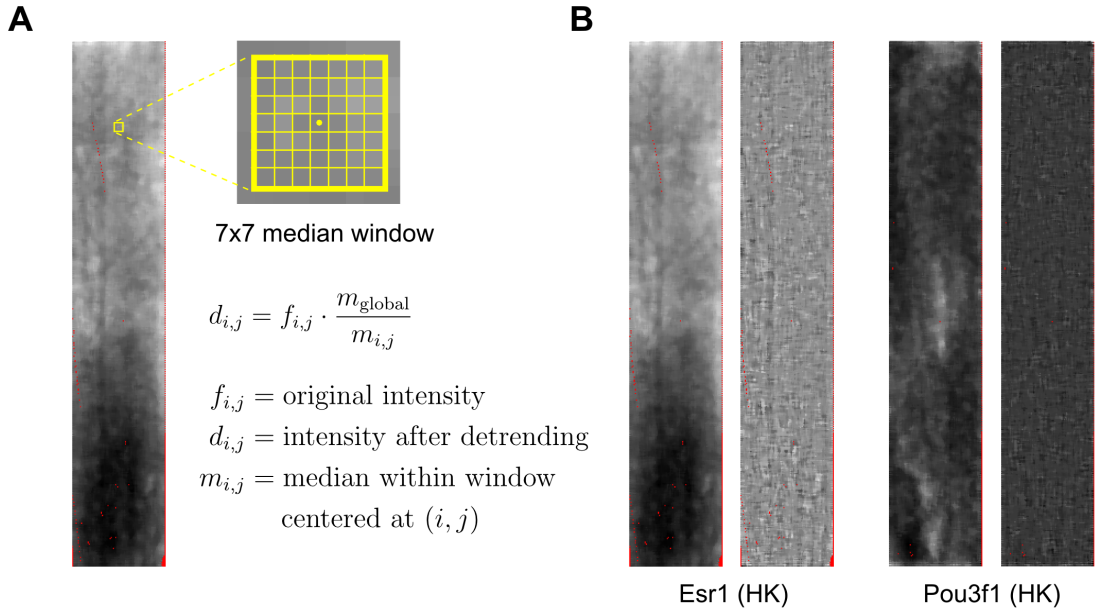


Figure 3.8: 7×7 median moving window approach to spatial artefact removal. Figure taken from Annala *et al.* [70]. **A** shows the model used to smooth the array. **B** shows before and after images for two selected arrays. The calculations shown are using intensity values that have not been log transformed.

critical places for spatial normalisation; some of the more severe spatial artefacts occur at the edges[1].

- a 2D contour plot is the only readily available way to view the effect of the normalisation. It is hard to compare this to the data and thereby gain a physical intuition of what is being affected.

A notable effect of this method is that, approximately, 1/49 probes will all have their corrected intensity set to the same value; that of the overall array median. To understand why this is true consider that roughly one out of 49 times the value $f_{i,j}$ will equal $m_{i,j}$ in equation 3.1, (taken from figure 3.8). Therefore $d_{i,j}$ is set to m_{global} for about 2% of the data.

$$d_{i,j} = f_{i,j}\frac{m_{global}}{m_{i,j}} \tag{3.1}$$

If we believe that there is useful information in most probe measurements then this seems undesirable.

3.2.2.2 LOWESS correction of spatial artefacts

Another method that obtains point estimates and is popular in micro-array analysis [82] is LOWESS [83]. LOWESS is a weighted linear regression method that fits a polynomial smooth of the data to each point with a weight function that has a local cutoff (Equation 3.2). The parameter λ controls the number of neighbouring points that contribute to the point x_0's value.

$$w(x) = (1 - |x - x_0|^3)^3 \mathbb{1}\{|x - x_0| < \lambda\} \tag{3.2}$$

Calculating the kd-tree necessary for a LOWESS fit is expensive computationally and makes interactive exploration of fits difficult. The LOWESS method offers only a single parameter that can be adjusted in contrast to spline approaches

[1] The edges of the arrays present a problem for any method since the data there seem to be some of the most prone to spatial artefacts and, at the same time, there is less local information available from neighbouring probes to use for normalisation.

described next. For these reasons only exploratory images were made using this method.

3.2.2.3 Splines for correction of spatial artefacts

Splines are a standard method used to obtain compact and efficient representations for smooth functions, including surfaces of physical objects [84]. Their use is ubiquitous in the modelling and computer aided design.

Splines are polynomial approximations to functions that are specified to be differentiable to some degree, often they are designed to have continuous third or second derivatives. A spline of degree 1 is simply a linear, or piecewise linear function.

Because splines are differentiable functions it is possible to calculate tangents and normals to surfaces and lines and also to select a particular parametrisation that has desired tangents and normals. B-splines have been used for normalising data in the micro-array context with reported success [85, 86].

In greater generality we can describe a function[1], f, as a linear combination of basis functions h_i (Equation 3.3). The weights, β_i are to be determined by an optimisation method such as minimising squared errors. The particular basis functions are chosen to simplify the task at hand.

$$f(x) = \sum_{i=0}^{n-1} \beta_i h_i(x) \tag{3.3}$$

An important difference between choices of basis function is whether they are local or global in their support[2]. Thin plate splines, and natural splines are both examples where the basis functions have global support. Thin plate splines and natural splines do provide a differentiable function with the desired smoothness properties but have trouble modelling the rather heterogeneous and locally vari-

[1] When modelling spatial artefacts on a protein binding micro-array the function we are interested in defines a surface, $f(x, y) = z$. (x, y) is the location of a probe and z is its intensity at this location.

[2] What is meant by local support is that each basis function is non-zero between only a few adjoining knots. Having global support means each basis function is non-zero on the entire domain.

able artefacts of the protein binding micro-arrays. For this reason, after some exploratory work these methods were abandoned in favour of B-splines.

A B-spline basis is local yet the result of fitting to the data is a continuous and differentiable function parameterised over a unit interval or square in the case of a surface. B-splines can also be parameterised on their degree d and on the number and position of the knots (Equation 3.4). The B-spline basis of degree d with knots ξ_i is constructed in such a way that the approximated function has d continuous derivatives, except at a knot, where it has $d-1$. B-splines can be constructed recursively (Equation 3.7) starting with indicator functions over the intervals defined by the knots. A key feature of B-splines is that, through careful placement of the knots, particular differentiability constraints can be achieved. By the repetition of knots at the endpoints of an interval the Runge effect [87], can be avoided. The polynomial approximation is constrained to be linear at the edges; the B-spline curve will pass through its last control point and have a tangent that interpolates its final two control points.

For the modelling of the protein binding micro-arrays, various numbers of basis knots and different degrees, d, were investigated in order to find a surface that models the spatial artefacts in a helpful way. Methods that involve penalising parameters were used (Section 3.2.2.4).

The d knots in equation 3.4, at either end of the range, take the same values, ξ_0 on the left and ξ_{N-1} on the right, enforcing linearity at the end points, these knots are denoted using τ.

$$\tau_0, \tau_1, \ldots, \tau_d, \xi_0, \xi_1, \ldots, \xi_{N-1}, \tau_N, \ldots, \tau_{N+d-1} \tag{3.4}$$

$$\tau_0, \ldots, \tau_d = \xi_0 \tag{3.5}$$

$$\tau_N, \ldots, \tau_{N+d-1} = \xi_{N-1} \tag{3.6}$$

The recursive definition of the basis functions is as follows,

$$B_{0,i}(x) = \mathbb{1}_{[\xi_i,\xi_{i+1})}(x) \tag{3.7}$$

$$B_{1,i}(x) = \frac{x-\xi_i}{\xi_{i+1}-\xi_i}B_{0,i}(x) + \frac{\xi_{i+2}-x}{\xi_{i+2}-\xi_{i+1}}B_{0,i+1}(x) \tag{3.8}$$

$$B_{r,i}(x) = \frac{x-\xi_r}{\xi_{i+r}-\xi_i}B_{r-1,i}(x) + \frac{\xi_{i+r+1}-x}{\xi_{i+r+2}-\xi_{i+1}}B_{r-1,i+1}(x) \tag{3.9}$$

By allowing a knot for every data point it is possible to obtain a smoothing spline [88]. In this case either an L_2 or L_1 penalty applied to the parameter vector offer regularisation and sparsity.

3.2.2.4 B-spline surface fitting

The B-spline surface is fitted to the protein binding micro-array data by using the product basis (Equation 3.10).

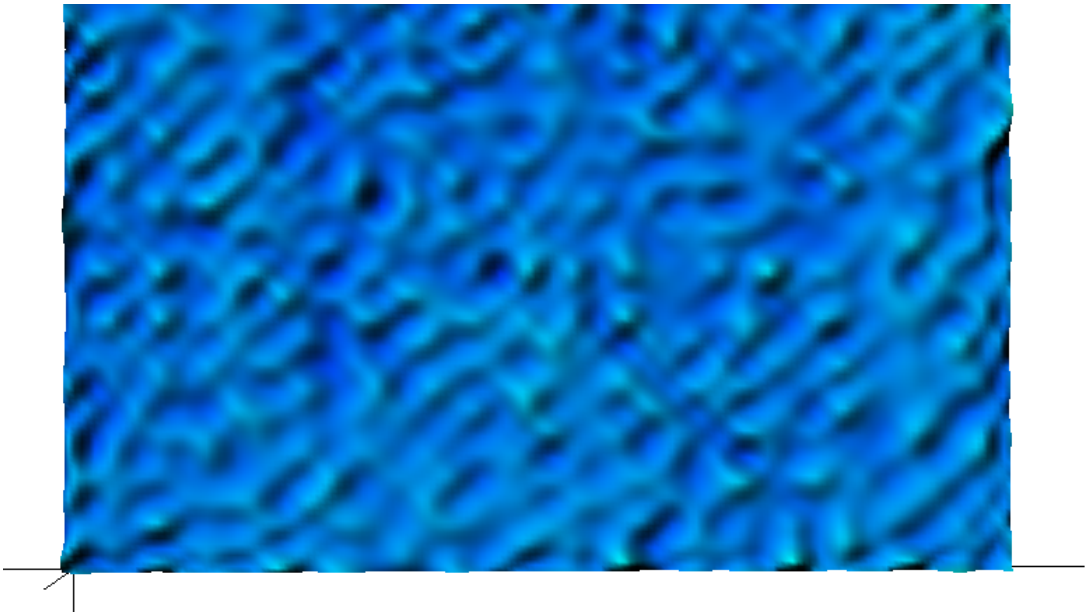
$$f(x,y) = \sum_{i=0,j=0}^{I,J} \beta_{ij}h_i(x)k_j(y) \tag{3.10}$$

In this case we have I B-spline basis functions h_i in the x-axis direction and J B-spline basis functions k_j in the y-axis direction. The parameters β_{ij} are to be obtained by optimising an error function, such as least squares.

The number of control points can be selected and their positions are also independently positionable. For instance if there is a missing data value, or values, then corresponding control points can be omitted. To model a hexagonal grid rather than a rectangular one the control points could be placed in a hexagonal grid. The latter was not attempted and would be an interesting further experiment; a possible hexagonal pattern was sometimes observed on fitted surfaces (Figure 3.9).

When fitting fewer control points than array probes a standard linear regression was used to obtain values for the control points. In the case where a greater number of control points than array probes were fitted, then both ridge regression and lasso were used. Ridge regression penalises control points that bias them towards being zero. This is helpful to avoid probes being overfitted as well as

Figure 3.9: Example of a possible hexagonal lattice pattern in a B-spline surface.



regularising an otherwise over-parametrised model. The lasso regularises but also obtains sparsity, if we have a prior belief that most of the probes should not have their intensities corrected, then this seems a valid option.

To fit the 2D B-spline surface, rendered in 3D space, we require a row in the model matrix for every pair of values, (x, y), at which the surface is to be evaluated. There are 4×4 non-zero weights for this 'patch', there will therefore be 16 non-zero weights in each row of the model matrix. The protein binding micro-array grid layout gives us 85 values in the x direction and 478 in the y direction meaning that the model matrix will have 40630 columns. A sparse matrix representation is efficient in terms of memory requirements and worked well in practice.

It is easier to understand what the effects of data smoothing are when we are able to visualise its effects in 3D (Figures**??**). In some cases it is clear that data would be better flagged and excluded rather than corrected with a surface fitting

Figure 3.10: 3D view of protein binding micro-array. In this image the data has been overfitted. In particular, the surface tangent, at the edges, at the rear right, can be seen to vary quite wildly. The blue plane shows the grand mean of the data.

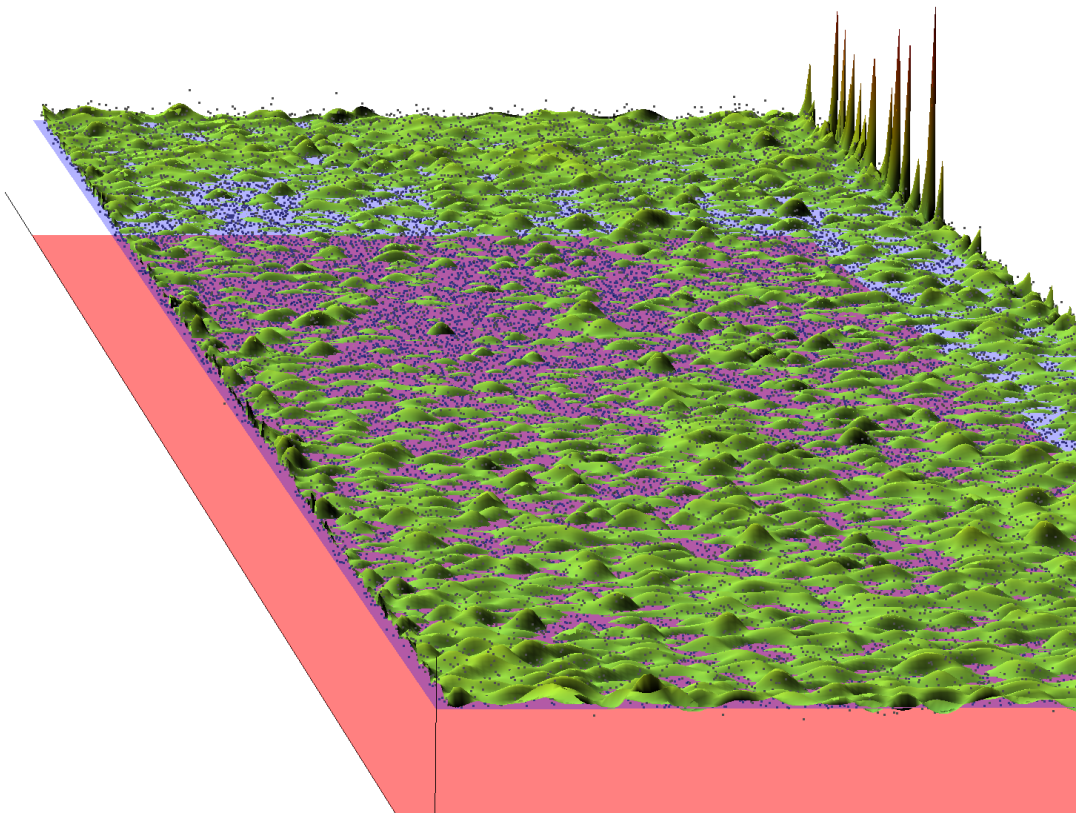


Figure 3.11: 3D view of Sox14 protein binding micro-array with data seen as a point cloud. Spatial artefacts are detectable with this representation.

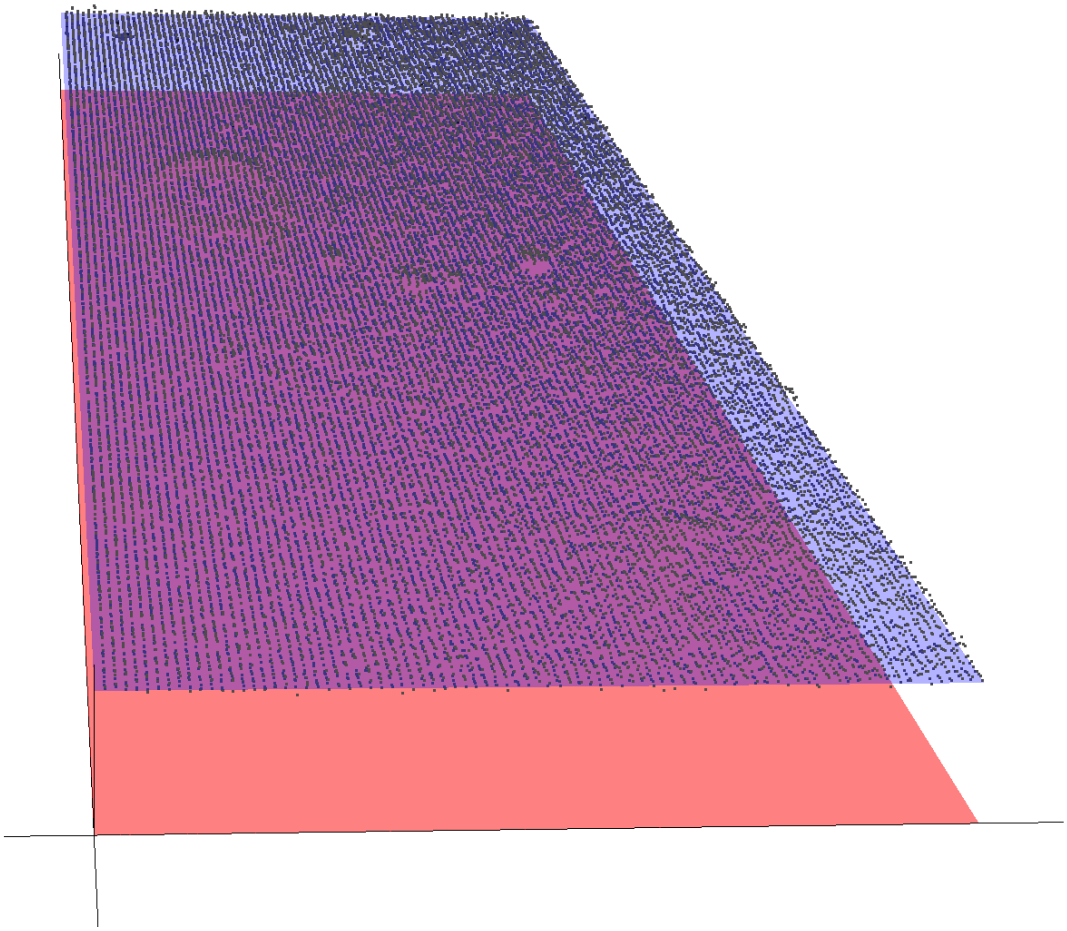


Figure 3.12: 3D view of Sox14 protein binding micro-array. A B-spline surface models uneven spatial distribution of intensities. The data and the surface are plotted together here.

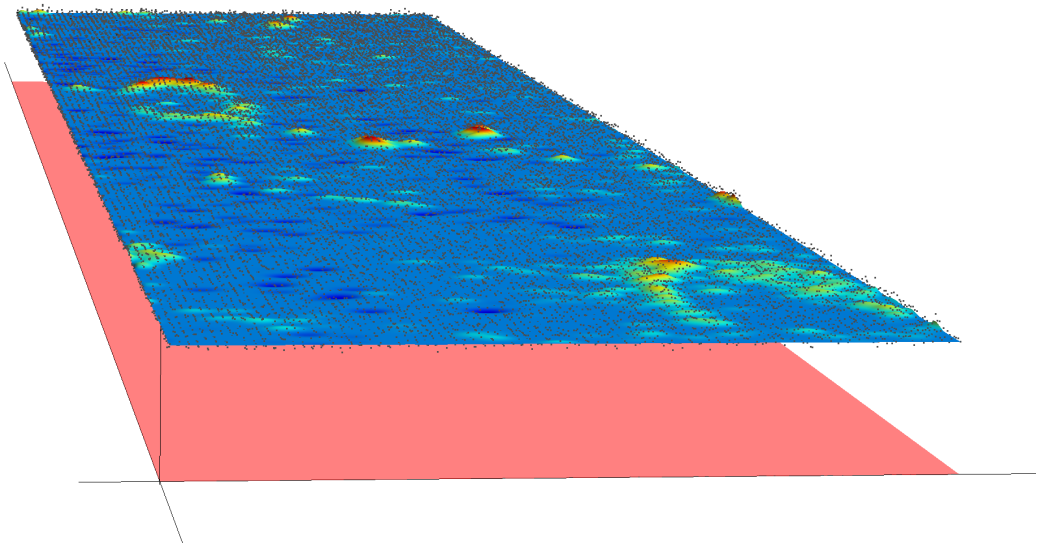
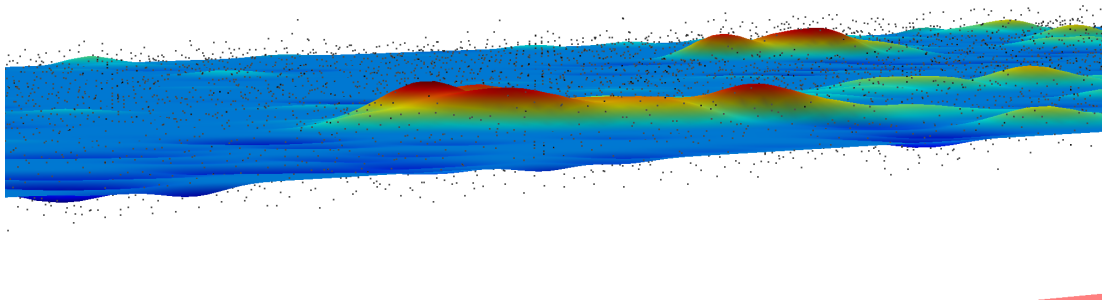


Figure 3.13: 3D view of Sox14 protein binding micro-array. This view gives a sense of the relative importance of the spatial artefacts to the data. In this case it is clear that probes in the vicinity of the spatial artefact have a strong distortion on their intensity reading.



procedure[1]. In other cases, particularly the edges, it is important to be able to see that the smoothing is working as planned and not introducing artefacts of its own (Figure 3.10). Figure 3.14 offers a strong impression of the balance that needs to be struck between overfitting *vs.* correcting spatial aberration. Figure 3.15 shows a B-spline surface that has been fitted to the data with fewer knots, i.e. 10 along the narrow edge of the array and 60 along the long edge. Smaller spatial artefacts are perhaps less easily corrected on this scale. Figures 3.16 and 3.17 give good examples of the physical intuition that one can obtain of the data using the 3D visualisation of the data and the B-spline surface fitting. It is instantly clear that the data have a good dynamic range compared to other arrays and also compared to the surface that is fitted to the spatial artefacts. But it is also clear that the data are saturated at the highest intensity level. There is good utility of being able to view the data in this way. The sparse matrix representation of a B-spline smoother is easily manageable when using a control point for every array spot. The thin plate spline does not afford this sparse representation and is therefore more computationally challenging.

There are a potentially large number of parameters involved in the surface fitting process, i.e. for every knot there can be a single parameter and the degree of the B-spline basis can also be chosen. It would be easy to overfit the data, the result of such overfitting is reduced predictive performance since 'signal' is removed from the data before the learner has a chance to use it.

Different methods have been tried to avoid this overfitting including reducing the number of knots, and therefore the number of parameters, directly and also using a regularised regression for parameter estimation. The sparse parameter selection method, employed later to fit model parameters, the lasso, was also tested for its utility in fitting the surface without overfitting.

Having every data point drawn as an individual vertex is easily achievable using OpenGL and scales to a million data points or more with current, non-expensive hardware, e.g. 2009 Apple Mac Book Pro. Tricks involving textures

[1]Cubic splines are an optimal fit to the data subject to a constraint on the second derivative of the fitted function [88]. If a region of our data has a step difference in intensity as compared to the neighbouring data then our prior belief that the second derivative should be small is violated. Clearly this is the case for 'clouds' of disjoint data such as those observed in figure 3.7

Figure 3.14: Three B-spline surfaces that have a knot for every data point and penalised control points. From the top picture to bottom picture the penalty is increased, leading to a smoother surface and therefore smaller corrections to the data. The rough surface at the top leads to greater corrections of the data and therefore smoother data. In the bottom right hand corner of each picture, where there is missing data, the combination of penalised control points and repeated knots, at the edges of the array, results in the surfaces tangent to being flat.

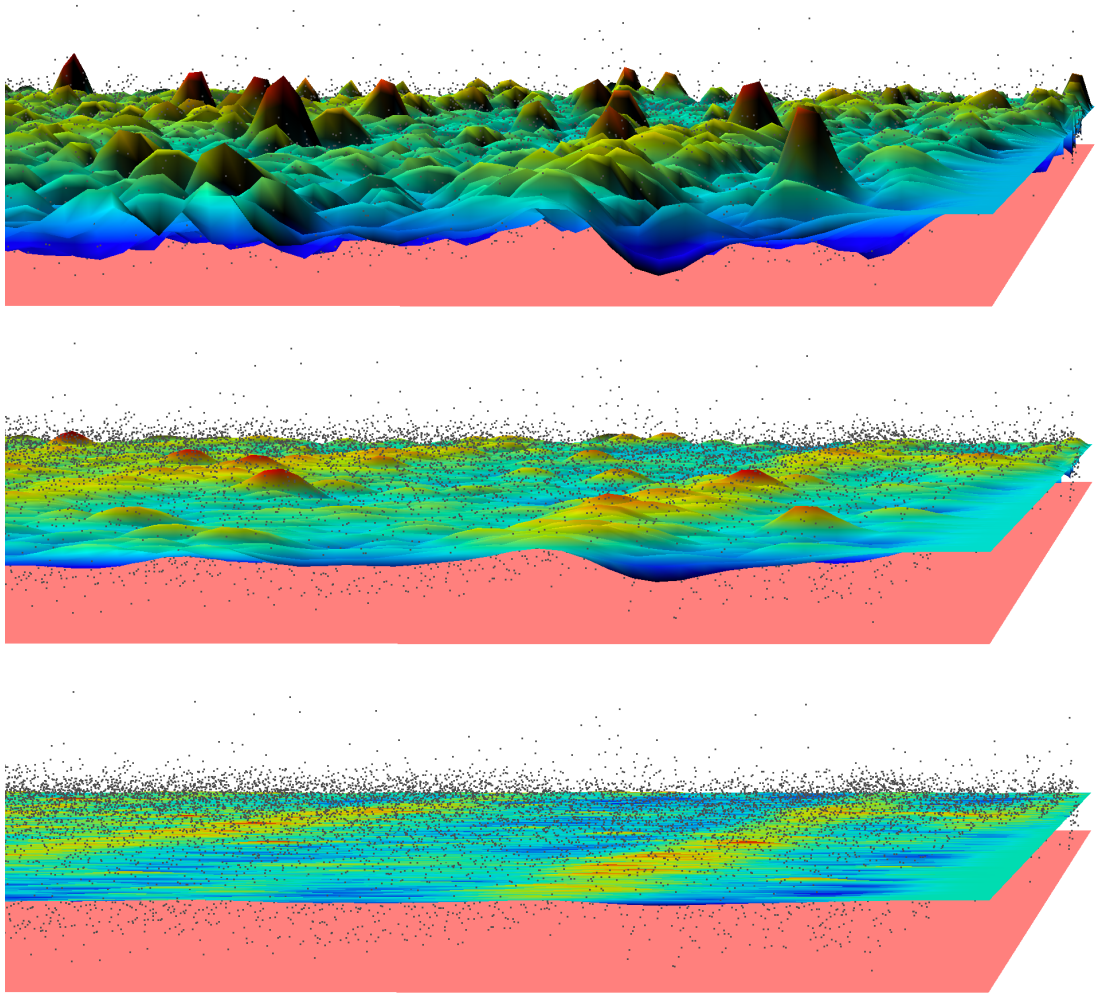


Figure 3.15: This surface has fewer knots, 10 across and 60 up. It can be seen that the control points near the missing data, in the bottom right corner, have been moved considerably.

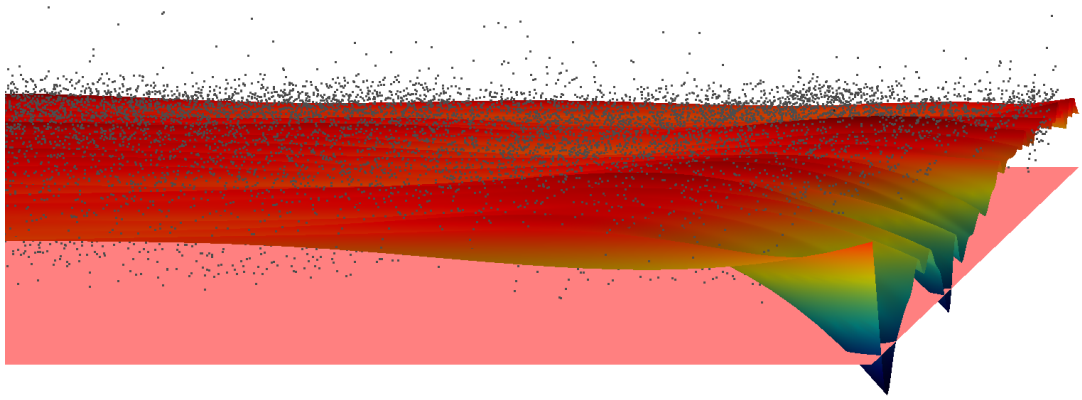


Figure 3.16: 3D view of Gmeb2 protein binding micro-array showing fitted surface.

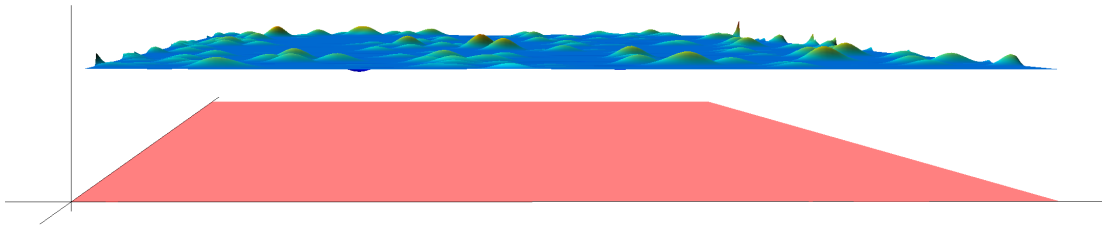
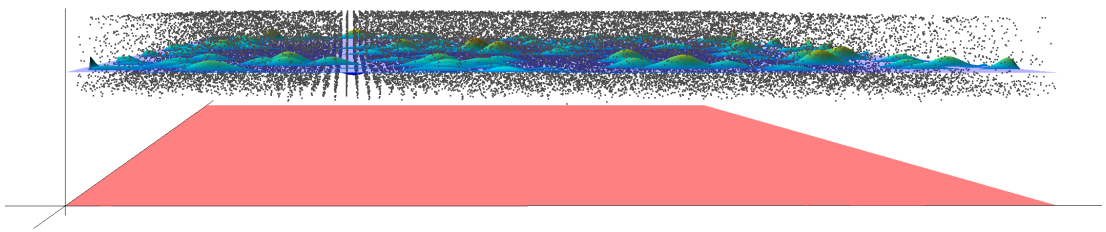


Figure 3.17: 3D view of Gmeb2 protein binding micro-array showing large dynamic range of data and also saturation at highest intensity.



allow the rendering of surfaces with far fewer resources and data transfer but preclude the interactive filtering and full 3D exploration of the data that has been achieved here. The software written to draw, filter and normalise the data has all been in the C++ language apart from the GUI that uses the OSX Cocoa API. C++ bindings to the Apple implementation of OpenGL were written to create an easy interface to the Eigen linear algebra library [81]. The Eigen library offers both sparse and dense matrix APIs and also matrix decompositions and solvers used in the surface fitting. A conjugate gradient solver was used for the ridge regression fits but the FORTRAN language glmnet algorithm was used for the lasso fits [89].

3.2.3 Methods to mitigate probe signal saturation

The histogram (Figure 3.21) of a protein binding micro-array for Egr2 shows a probe intensity distribution spike at the $\log_2$ value of 16. This corresponds to a 16 bit digital value being saturated. Also, the point cloud image for the Gmeb2 micro-array (Figure 3.17) shows an excellent dynamic range but saturation at the highest intensity level that the image sensor can record, i.e. 16 given as log base 2.

We might believe that the highest intensity readings for an array correspond to probes that have the most significant binding sites for a transcription factor. Differences in the binding specificities to probes with the highest signals may contain some of the most interesting information. It is of value therefore to have methods that can recover some of the signals lost in the saturated data.

3.2.3.1 All array quantile normalisation

In Annala *et al.* [70], a quantile normalisation across all the protein binding micro-arrays in the DREAM5 data-set was made. The authors comment that an assumption implicit when making this transformation is that the distributions of probe intensities, for each array, is the same. Array quantile normalisation is a much used method for the pre-processing of gene expression arrays [90]. The method used is to calculate a consensus distribution (Figure 3.18) across all micro-arrays. Each probe, according to its rank, is given the intensity of a

probe, with the same rank, in the consensus distribution. The authors comment

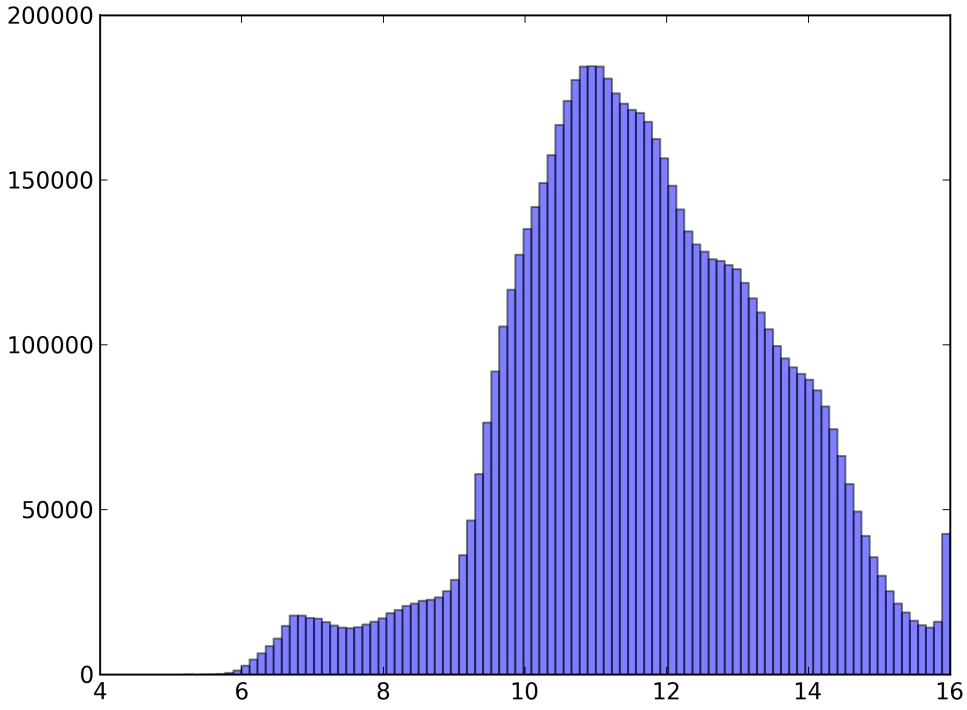


Figure 3.18: The consensus distribution for the 172 protein binding microarrays in the DREAM5 data-set. Clearly the consensus looks nothing like any individual array distribution.

that an assumption of equal distributions is 'subject to debate'. The authors suggest that the improvement in predictor performance achieved by this all array quantile normalisation is likely due to its positive impact on the most saturated arrays. It is clear, and potentially of relevance biologically, that the distributions of intensities for different transcription factors are, in fact, not equal. Figure 3.19 illustrates some of these points.

3.2.3.2 Quantile normalisation with background data

In Annala *et al.* [70] it is suggested that a slightly more discriminating approach, rather than performing all array normalisation, would be to select only those

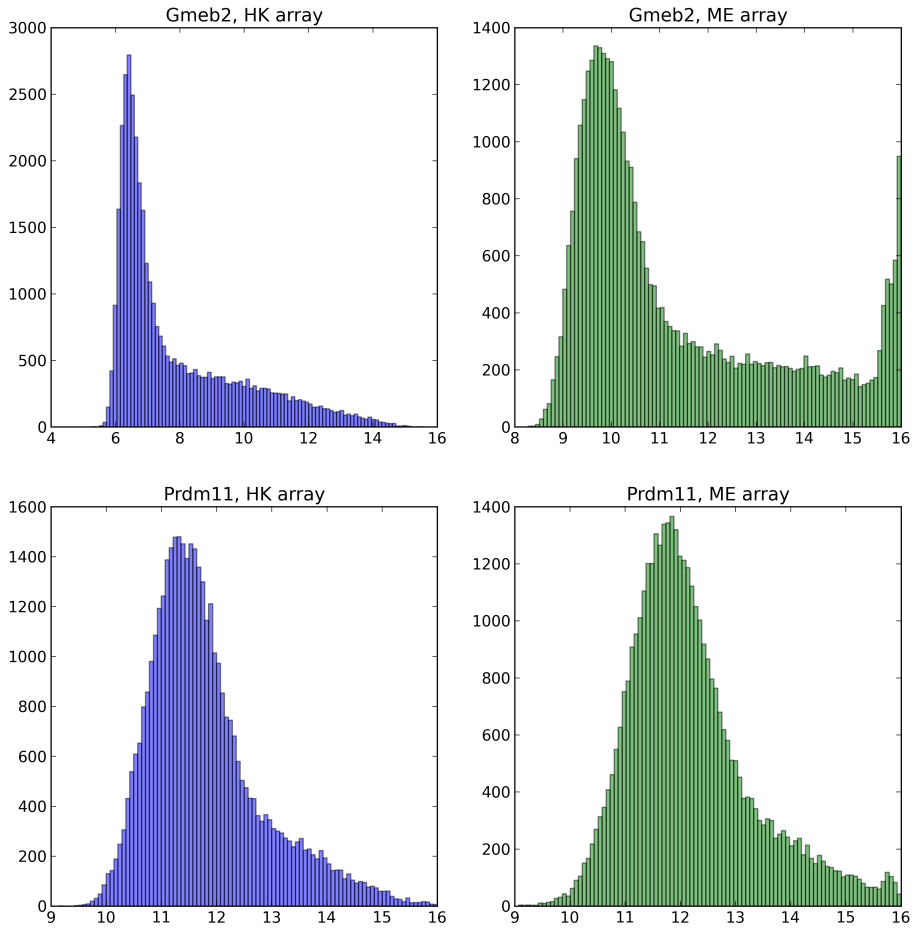


Figure 3.19: Distributions of intensity signals for two pairs of protein binding micro-array experiments. On the top row, distributions for the transcription factor Gmeb2 are observed, with a distinctive shape, that is different to those on the bottom row, which are for the transcription factor Prdm11. Each row shows the distribution of intensities for a pair of experiments, i.e. one on each of the array designs, HK and ME. It is of note that the ME array, for the transcription factor Gmeb2, has a significantly saturated signal at the highest intensities.

arrays that have a non-saturated signal to form the consensus distribution.

An even more selective alternative is to quantile normalise the foreground probe intensity information with the background probe intensities for a single experiment. This is only possible where background probe intensity information is available of course. The histograms for Irf2's foreground and background signals (Figure 3.20) over two experiments, show a case where the signals are saturated in the foreground but not in the background. The idea then, is to use the information in the background signal to separate those foreground signals that have been 'squashed together' at the saturated peak. If we believe that the background signals contain less information than the foreground signals then we would like to keep the good information in the foreground signals whilst using the background signals to 'correct' the saturated signals in the foreground. Considering figure 3.23 it seems likely that we can recover some of the correct shape of the distorted distribution of high intensity signals using the background distribution.

In figure 3.21 a similar set of foreground and background signals for a pair of experiments is seen. In this case the background distributions do not have such a similar appearance to the foreground distributions.

3.2.4 Implementation details

Data for every array is stored in an Sqlite database and is mapped using the object relational mapping software ODB. It is possible to pull in all necessary information on any array in less than a second. For example 40000 intensity readings and their grid references are pulled in and rendered in under a second. The database is approximately 0.5GB in size, depending slightly on how many summary statistics are calculated and stored therein.

The implementation of a B-spline curve is greatly simplified when one is able to use regularly spaced knots. A single $(d+1) \times (d+1)$ sized matrix can be used to calculate weights for every control point. The boundary conditions add an extra complication.

Basis functions for B-splines are calculated on the CPU and control points are then optimised for using either a normal linear regression, a penalised linear regression or the elastic net algorithm.

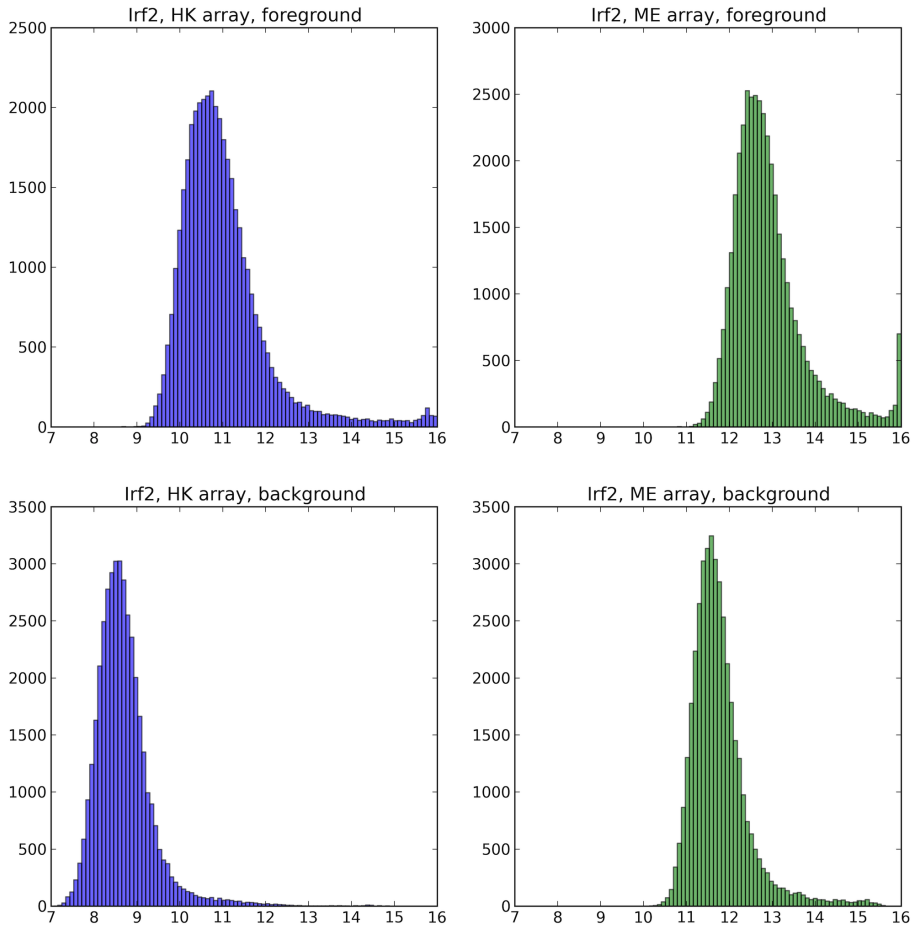


Figure 3.20: Intensity distributions for the transcription factor Irf2. Each histogram in the top row shows the distribution of intensities for a different array design. The HK array, to the left, shows a small amount of saturation at the highest signal. Whilst the array to the right, the ME array, shows several hundreds of probes saturated at the highest intensities. In each case it can be seen that the shape of the background distribution closely follows that of the foreground distribution.

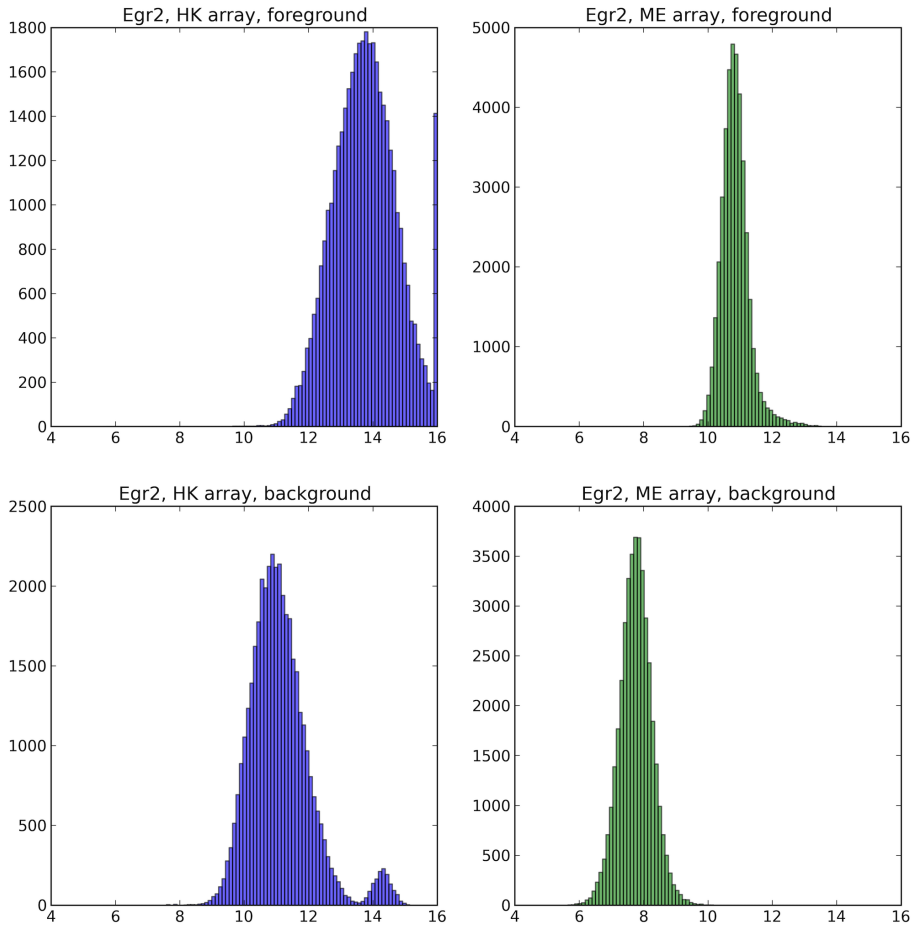


Figure 3.21: Intensity distributions for the transcription factor Egr2. Each histogram in the top row shows the distribution of intensities for a different array design. The HK array, to the left, shows a small amount of saturation at the highest signal. Whilst the array to the right, the ME array, shows several hundreds of probes saturated at the highest intensities. It is not clear what the bump to the right of the Egr2 background distribution is.

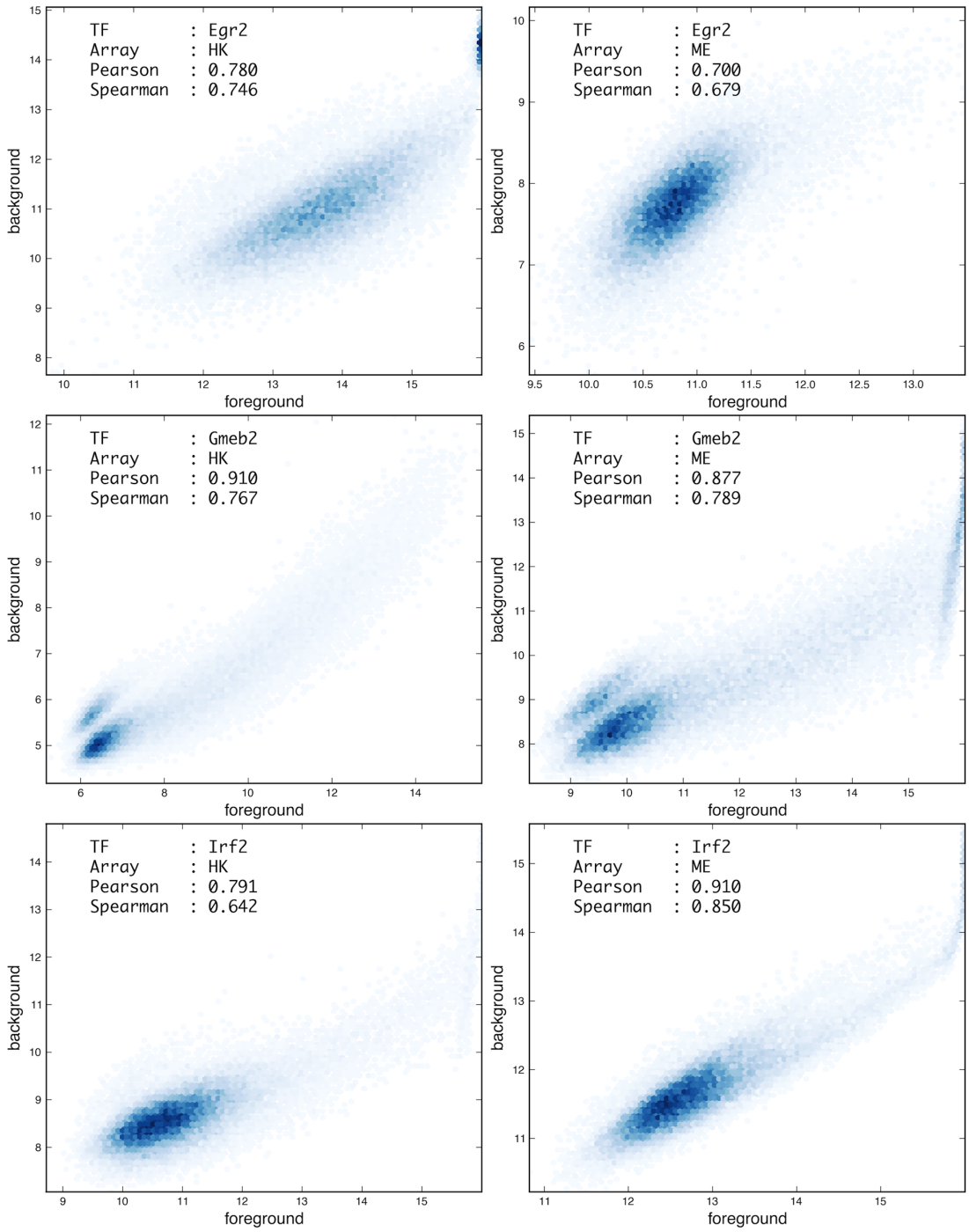


Figure 3.22: Scatterplots showing the relationship between foreground and background intensities for a selection of transcription factors. The plots on the top left, middle right and bottom right show arrays that have saturated foreground signal intensity. In each case, there is a good correlation between the background and foreground signals and the background signals are not saturated. It is not clear what the 'twin clouds', at low intensities, in the Gmeb2 plots are.

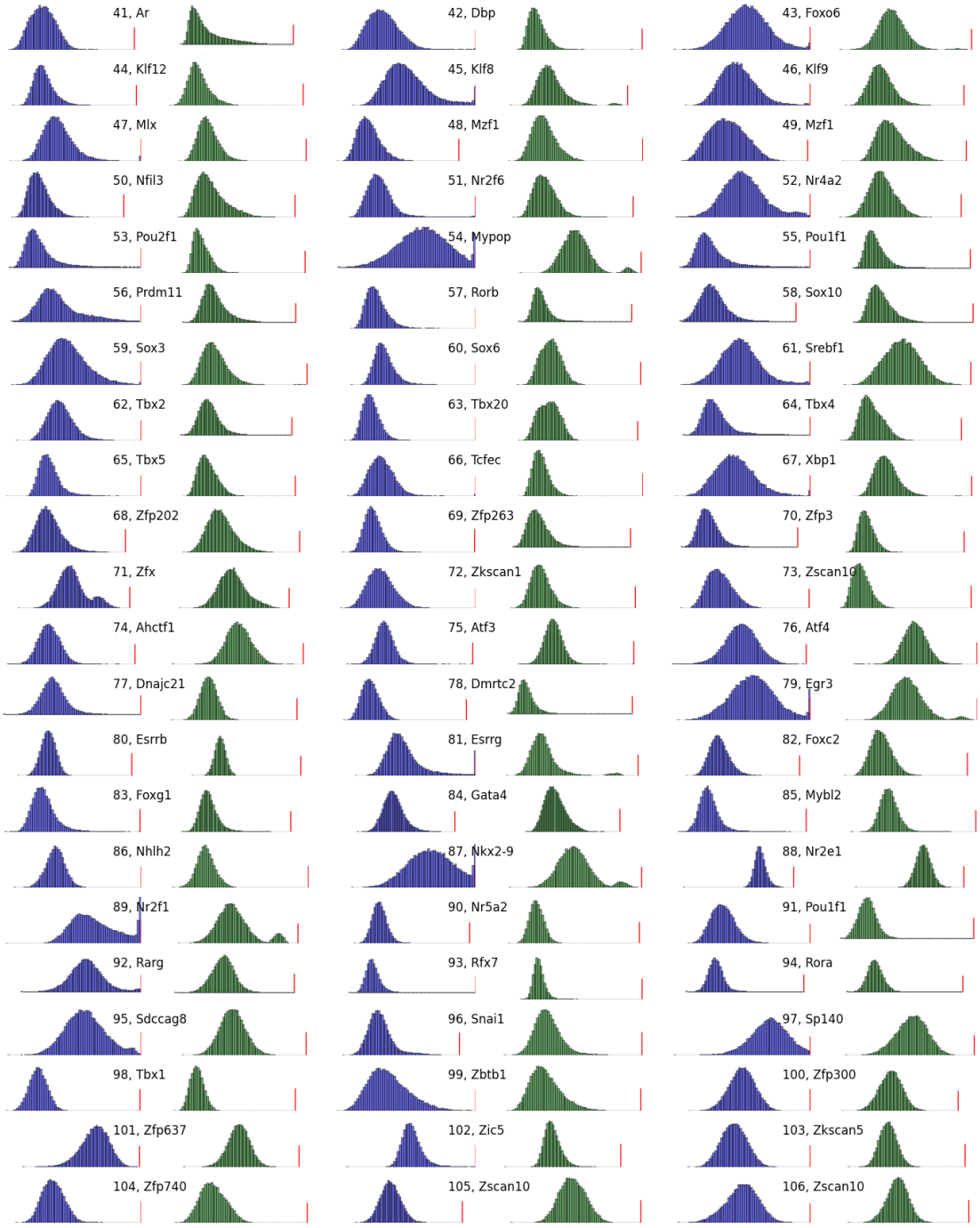


Figure 3.23: Histograms showing the foreground and background intensity distributions for half of the DREAM5 arrays. The red bars show the locations of the highest intensities that are otherwise hard to observe in the histogram. The blue histograms show the foreground intensities and the green show the background intensities.

Once vertices and their positions have been calculated they are shifted to the GPU for rendering using the OpenGL 3.3 API.

3.3 Results

We wanted to evaluate possible improvements made by spacial normalisation and distributional normalisation of the probe intensities. Prior to this evaluation, flagging of outlier probes was carried out using the GUI tool for each of the arrays (Section 3.2.1) with flags stored in a database and used to exclude the data for all following steps. One approach to measure success in the correction of spatial and saturation artefacts is to measure our ability to predict the measured probe signals on a counterpart array[1].

The prediction method is to be discussed in detail in chapter 5, for this demonstration a vanilla set of parameters will be fixed in all cases. The predictor will be referred to as the lasso predictor. The rationale for this approach to measuring efficacy of spatial and saturation corrections is that any spatial and saturation corrections that enable us to better predict an independent array are improving the signal to noise ratio on the training array.

Pearson correlation between predicted and true values was measured before and after making the spatial and background corrections. The same set of 66 arrays were chosen that were the arrays initially part of the DREAM5 challenge. This helps with comparisons to previously published results.

3.3.1 Spatial normalisation results

Figures 3.24, 3.25 and 3.26 show the small differences caused by surface normalisation on lasso predictions for a subset of the protein binding micro-arrays. Table 3.2 has an overall comparison. Both the median window method and the B-spline method, with 10×60 knots, are comparable. The latter does slightly better

[1]Here, a counterpart array is a second protein binding micro-array with a different de Bruijn sequence and the same DNA binding protein. In the DREAM5 data-set every transcription factor was added to two arrays, one HK design array and one ME design array. These two arrays are described as counterparts.

over-all. The penalised B-spline method improved upon no spatial normalisation but comes in behind the other methods.

In Annala *et al.* [70] improvements afforded by spatial corrections were noted as roughly a 0.003 mean improvement in Pearson correlation. The same median method implemented here, and assessed via its effect on the lasso predictor, is very slightly improved. The 10×60 knot B-spline surface smooth slightly improves the predictor performance again.

Table 3.2: Comparison of spatial artefact correction reported in Annala *et al.* [70], a similar method re-implemented here and also the B-spline approach implemented here. Before and after Pearson correlations are given and the relative changes.

	Annala et al predictor	**Lasso predictor**	
Smooth Type	Median	Median	B-Spline, $\mathbf{10 \times 60}$
Uncorrected	.6094	.6866	.6866
Corrected	.6119	.6899	.6904
Change	.0025	.0033	.0038

Given the results in table 3.2, in the following work described in this thesis, the data-set used to build predictive models is all based upon data normalised with the B-spline spatial artefact correction using lasso penalisation of surface parameters.

3.3.2 Saturation normalisation results

Figures 3.27 and 3.28 show the effects on the lasso predictor's performance of the quantile normalisation of foreground intensities to background intensities on a subset of the protein binding micro-arrays. Changes in predictor performance are observable in both directions and sometimes, in contrast to the spatial corrections, these changes can be substantial. For this reason it is better to apply this normalisation selectively to arrays that clearly show saturation.

Three transcription factors that have clear saturation artefacts are shown in figures 3.19, 3.20 and 3.21, Egr2 on the HK array, Irf2 on the ME array and Gmeb2 on the ME array. Table 3.3 shows that quantile normalisation with background signals consistently offers an improvement on this hand picked selection of arrays.

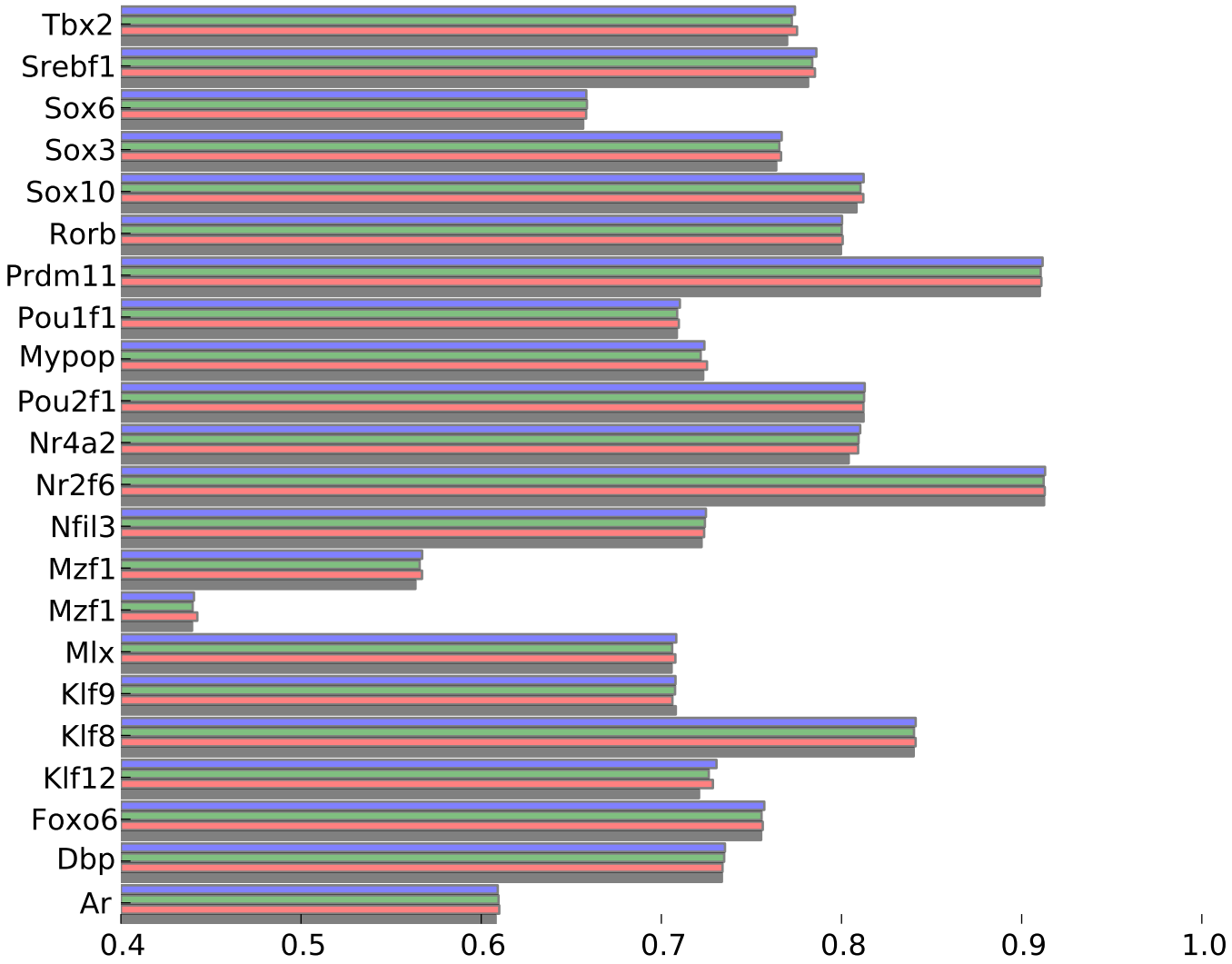


Figure 3.24: Bar chart showing correlations of predicted values with true values after different types of surface normalisation,

- 10×60 knot B-spline
- penalised B-spline
- median window
- None

This chart shows the first 22 of the 66 DREAM5 challenge arrays, these arrays are all of the HK design.

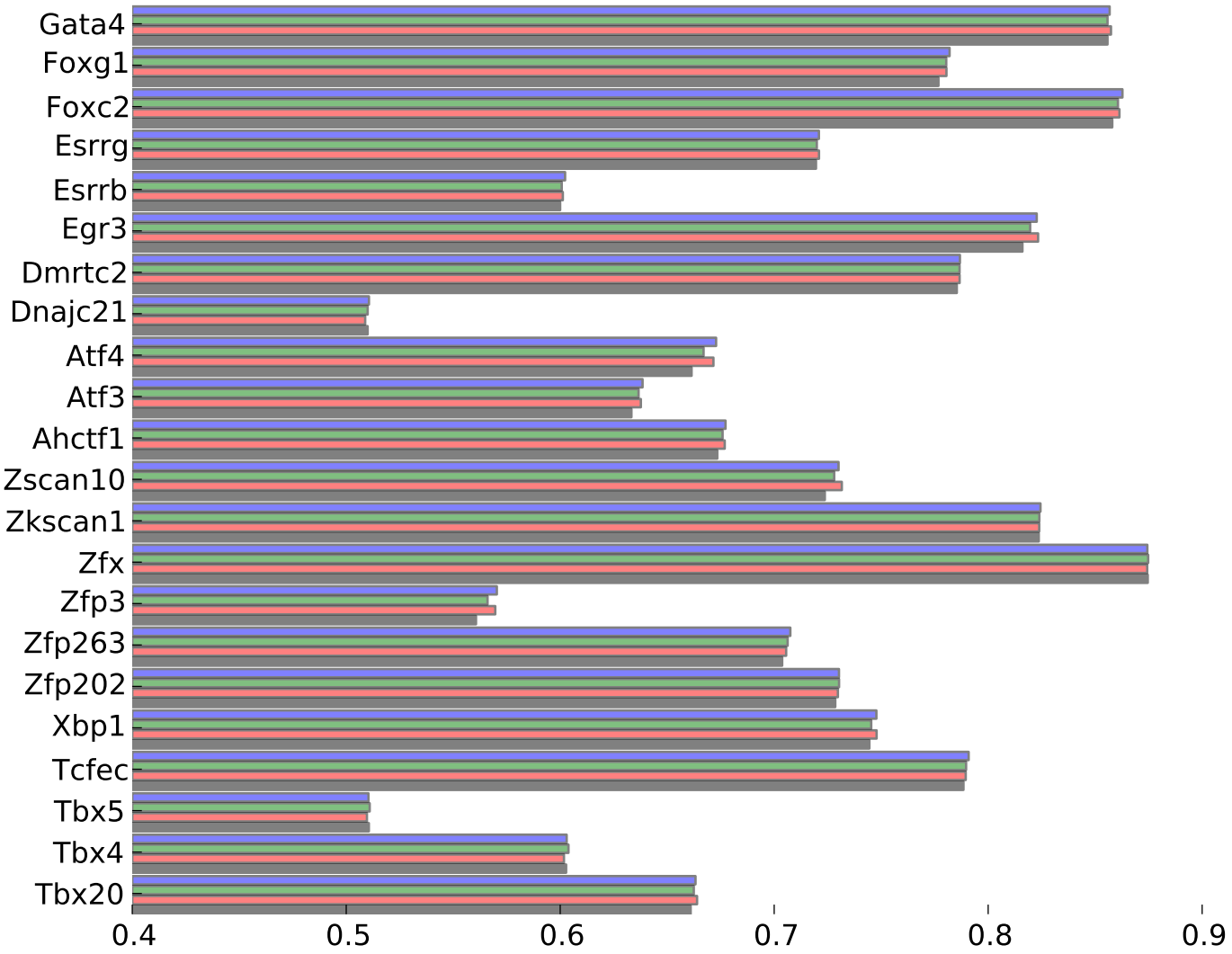


Figure 3.25: Bar chart showing correlations of predicted values with true values after different types of surface normalisation,

- 10×60 knot B-spline
- penalised B-spline
- median window
- None

This chart shows the second 22 of the 66 DREAM5 challenge arrays, the first 11 of these arrays are of the HK design the second 11 are of the ME design.

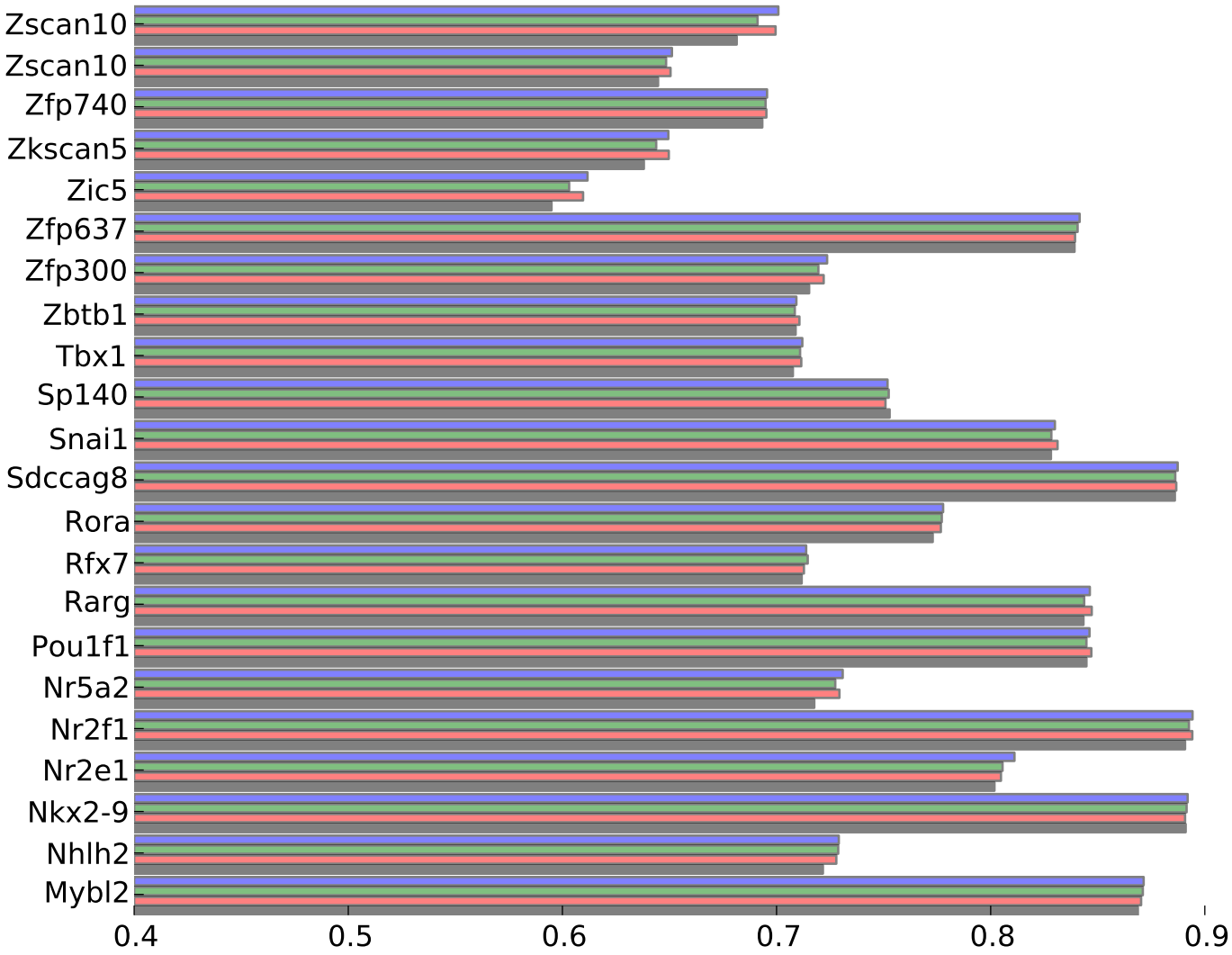


Figure 3.26: Bar chart showing correlations of predicted values with true values after different types of surface normalisation,

- 10×60 knot B-spline
- penalised B-spline
- median window
- None

This chart shows the third 22 of the 66 DREAM5 challenge arrays, these arrays are all of the ME design.

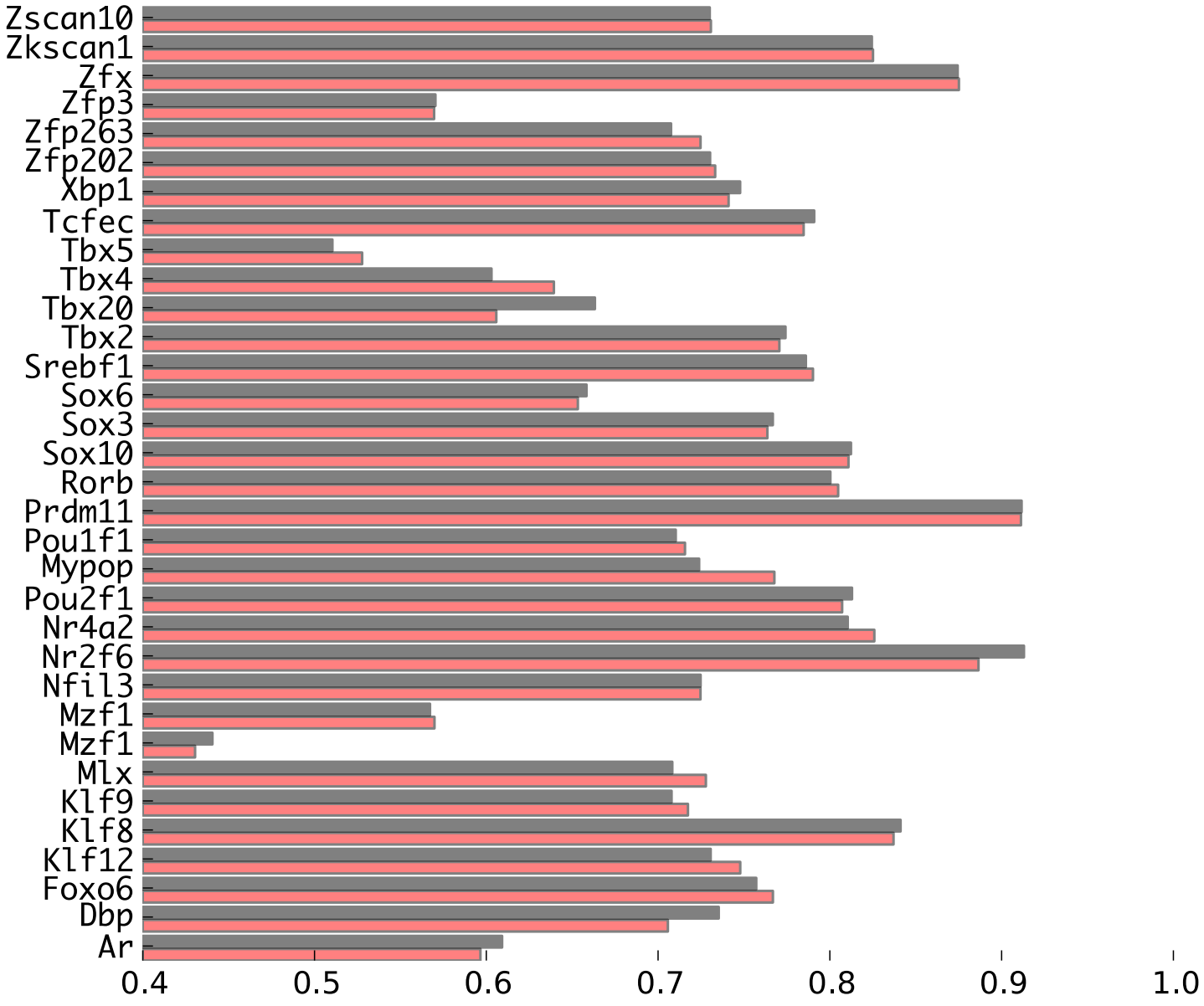


Figure 3.27: Effect of saturation correction on 33 HK arrays.

- No saturation correction
- Background quantile correction

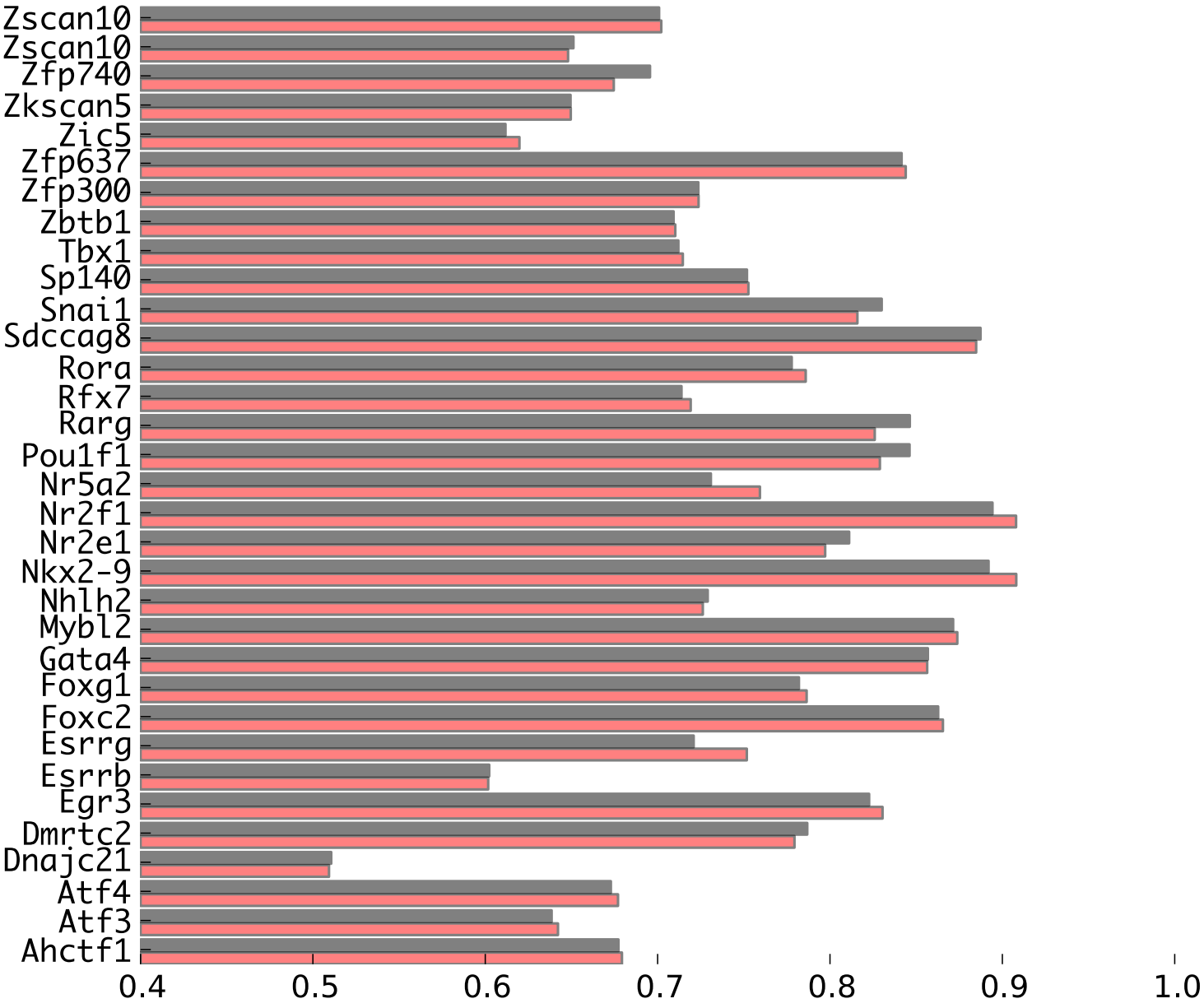


Figure 3.28: Effect of saturation correction on 33 ME arrays.

- No saturation correction
- Background quantile correction

It is easy to make this selection by visual inspection of a foreground intensity histogram (Figure 3.23).

Table 3.3 shows that the all array quantile normalisation has caused a decrease in predictive performance on the array Irf2 in the previously published results from Annala *et al.* [70]. On the other the normalisation with background intensities developed here improves predictive perfomance in all the examples shown.

Table 3.3: Comparison of all array quantile normalisation used in Annala *et al.* [70] against quantile normalisation used only with background signals. The Pearson correlation coefficient is given in each case.

	Annala et al predictor		**Lasso predictor**	
	No QN	QN	No QN	QN
Egr2	.763	.801	.808	.827
Gmeb2	.629	.775	.952	.953
Irf2	.677	.640	.810	.814

Given the results in the table 3.3, in the following work described in this thesis, the data-set used to build predictive models has had saturation normalisation applied to arrays selected by visual inspection to have significant foreground saturation.

3.4 Conclusions

In this chapter, some measurements on the effects of normalisation procedures are shown. A tool was produced that allows the easy exploration of the heterogenous data and the building of an intuition about the data. This is described further in appendix A. A number of methods were evaluated through their effects on predictor performance. The most effective were selected and applied to create a cleaned and normalised data-set that is used throughout the rest of this thesis.

Most of the differences observed in the overall improvement of correlation are small, for both spatial and saturation corrections. Useful questions to ask are,

- What is the purpose of optimising overall correlation?

- How might the corrective measures applied here have improved other metrics?

For the latter, measurements have not been made in this thesis to address these matters adequately. The size of corrections made to individual probe intensities is large in many cases when compared to the dynamic range of the data. If we want to compare the intensity readings of a pair of probes to make a quantitative statement about their relative binding affinity then the corrections made could indeed be significant.

It seems that the protein binding micro-array platform could be used for more quantitative modelling of protein DNA interactions. This would involve many replicate experiments with the same transcription factor on the same array design, then several replicates at different concentrations, and then replicates with different de Bruijn sequences. This subject is expanded in the conclusions to this thesis.

Chapter 4

Probe Analysis and Model Matrix Construction

This chapter turns to the investigation of the protein binding micro-array probes and the construction of the 'model matrix' used in subsequent parts of this thesis.

In section 4.1.1.1 there is a summary of the probe sequences as published for the DREAM5 competition. After this look at the data there is an introduction to de-Bruijn sequences (Section 4.1.1.2). An understanding of de-Bruijn sequences is necessary to properly appreciate the HK and ME array designs, background on each of these designs is given in sections 4.1.1.3 and 4.1.1.4.

Armed with our understanding of the composition of the probes and their statistical properties we go about the important task of feature selection and model matrix construction, background on this is given in section 4.1.2.

The methods section is split into methods used in the analysis of the probes and their underlying de-Bruijn sequences (Section 4.2.1) and methods used in the construction of the model matrix (Section 4.2.2).

The results section is split in a similar way. Firstly results on the analysis of the array designs are given (Section 4.3.1) and secondly empirical results on properties of the model matrix (Section 4.3.2).

As part of the careful investigation into the properties of the HK and ME array designs the following techniques, whose application to this domain is novel, were developed,

- A method to calculate the generating polynomial of the HK array design is described and used to retrieve it. This adaptation of the Berlekamp-Massey algorithm is used to show the probes on the array have been modified in certain positions compared to the generated sequence.
- The published method for the construction of the ME design is shown to be missing some key information. This observation was the subject of Orenstein & Shamir [69], published during the writing of this thesis.

Conclusions for this chapter are found in section 4.4.

4.1 Background

4.1.1 Array Probe Analysis

4.1.1.1 Description of a protein binding micro-array's probes

The sequences that are used on the HK and ME arrays[1] are different, though they have important properties in common. Each design has probes that are segments from sequences that compactly represent all possible 10-tuples, each sequence is produced by a different method and has different properties. Descriptions of these methods are given in sections 4.1.1.3 and 4.1.1.4. Implications for the modelling and analysis of the protein binding micro-arrays are discussed.

Figure 4.1 is a sample of the probe information from the HK array design as given in a data file provided by the organisers of the DREAM5 challenge. These are 35 base pair sequences that correspond to the template strands of the double stranded DNA probes that are attached to the glass slides of the protein binding micro-arrays. The linker sequence that connects the 35bp sequence to the array, which is itself 25bp long, is omitted here.

[1]Each of the DREAM5 protein binding micro-arrays has probes from one of two designs; these designs are referred to as HK and ME.

Figure 4.1: A sample of the probes described in a data file for the HK array design.

```
probe_id probe_sequence
HK00001  TAAAAGTCAAGGATAAGTTTCCGGCACCGCAAATA
HK00002  CCGCAAATATGGGATTAGCCATAGTCTTGCATAGC
HK00003  TTGCATAGCAAAAATGCATGTGTGCTCGATGAAAA
HK00004  CGCAGATCATTCCCCGCGGGACGGAGTTTTCATCG
HK00005  TGATCTGCGCTGTCATGCAAAGCAAGCATAATAGT
```

Figure 4.2: A segment of a de Bruijn sequence obtained by aligning the probes from figure 4.1. The ellipses are for the central segments, of each probe, omitted here to facilitate display.

```
TAAAAGTCA...ACCGCAAATA
            CCGCAAATAT...CTTGCATAGC
                        TTGCATAGCA...TCGATGAAAA
                                    CGATGAAAAC...ATGATCTGCG
                                                TGATCTGCGC...
```

Concatenating these 35bp sequences in the given order, with a nine base pair overlap, gives us the alignment in figure 4.2, but with one complication. The sequence in red, HK00004 in figure 4.1, has to be reverse complemented to enable the sequences HK00003, HK00004 and HK00005 to be aligned as shown in figure 4.2. The reverse complement of probe HK00004 is shown in green in this alignment. By using reverse complements where necessary it is possible to concatenate all probe sequences in the data file into a single sequence of length $4^{10} + 10 - 1$. This is the de Bruijn sequence, from which the probes are taken, for the HK design.

In the case of the ME array design a concatenated sequence can be produced in the same way. Though in this case, the sequence has approximately half the length due to the redundancy of reverse complements having been removed, this is discussed in more detail in section 4.1.1.3.

4.1.1.2 Introduction to de Bruijn Sequences

De Bruijn sequences are very well studied as mathematical objects in their own right. They also have several important applications such as position sensing [91], functional magnetic resonance imaging (fMRI) [92], DNA sequence assembly [93] and DNA synthesis for synthetic biology applications [94]. Their properties are needed for the construction of protein binding micro-arrays since they offer the most efficient way to represent all possible 10-tuples. An appreciation of the properties of de Bruijn sequences gives us an understanding of the statistics of the sub-sequences contained within an array's probes. Efficient methods to generate and analyse the sequences are afforded by an appreciation of the mathematics behind them, hence the somewhat detailed investigation that follows.

A de Bruijn sequence is a maximally compact representation of all possible strings of symbols of a given length from a finite alphabet. Of particular interest to us is the alphabet $\{A, C, G, T\}$ and strings of a length that might correspond to a transcription factor recognition site. A de Bruijn sequence over an alphabet of c characters, with every possible length u string contained therein, will be described as a 'c-ary de Bruijn sequence of span u'. By 'maximally compact' what is meant is that, in a c-ary de Bruijn sequence of span u, each length u sub-string will occur *exactly* once. For example, given the alphabet of nucleotide bases, $\{A, C, G, T\}$, and nucleotide sequences of length 10, we would speak of a 4-ary de Bruijn sequence of span 10. This 4-ary de Bruijn sequence will have 4^{10} elements when viewed as a periodic sequence or $4^{10} + 9$ elements when taken as a linear sequence.

It may not seem immediately obvious that a de Bruijn sequence will necessarily exist for any given span and alphabet size. In fact, such sequences exist for all sizes of alphabet and all spans. Moreover, the number of such sequences is very large. One way to demonstrate that such sequences exist is through the properties of a de Bruijn graph (Figure 4.3).

General de Bruijn sequences can be obtained from a graph traversal. The nodes on the graph are the set of possible length u strings from an alphabet of size c and directed edges from a node x (Equation 4.1) to y (Equation 4.2) whenever

the last $u-1$ symbols of x overlap the first $u-1$ members of y (Equation 4.3).

$$x = x_0 x_1 \dots x_{u-1} \tag{4.1}$$

$$y = y_0 y_1 \dots y_{u-1} \tag{4.2}$$

$$x \to y \iff x_1 x_2 \dots x_{u-1} = y_0 y_1 \dots y_{u-2} \tag{4.3}$$

A de Bruijn sequence can be represented as a circular or periodic sequence (Figure 4.5). A linear representation of a span u de Bruijn sequence requires extra elements added to the end in order to contain every possible length u sub-string. For a 2-ary de Bruijn sequence of span 3 the following demonstrates this detail; between any pair of colons the sub-string 001 is missing but is present when taken as a periodic or circular sequence.

```
...00:010111000:010111000:010111000:01...
```

This need to 'add back' extra elements to ensure all sequences are represented is important to appreciate in the construction of the protein binding micro-array probes. Each probe is a short segment of a larger sequence and so extra elements have to be 'added back' to the end of each probe to make sure that every 10bp sequence is represented, unbroken, on some probe of the array.[1]

[1]This does mean that a 9bp sequence that is a suffix of a probe will be repeated as a prefix of the next probe. This could provide an interesting method to test the importance of probe position on 9bp sequences.

Figure 4.3: A binary de Bruijn graph of span 3. Any Hamiltonian cycle through this graph, i.e. a cycle visiting each node exactly once, will describe a binary de Bruijn sequence of span 3. The sequence can be read off by overlapping successive nodes on the Hamiltonian cycle, the last two bits are overlapped with the first two of the next node. The overlap is also described by the blue labels, a bit is left shifted into the least significant bit when traveling between nodes. E.g. $000 \rightarrow 001$ is labelled with a blue 1 since a one is shifted into the binary 3-tuple from the right. The existence of an Eulerian cycle, i.e. visiting each edge exactly once, is guaranteed by the even in-degree and even out-degree of each vertex. An Eulerian cycle on this graph describes a de Bruijn sequence of span 4. In this case the sequence is read as the sequence of blue labels as the Eulerian cycle is followed.

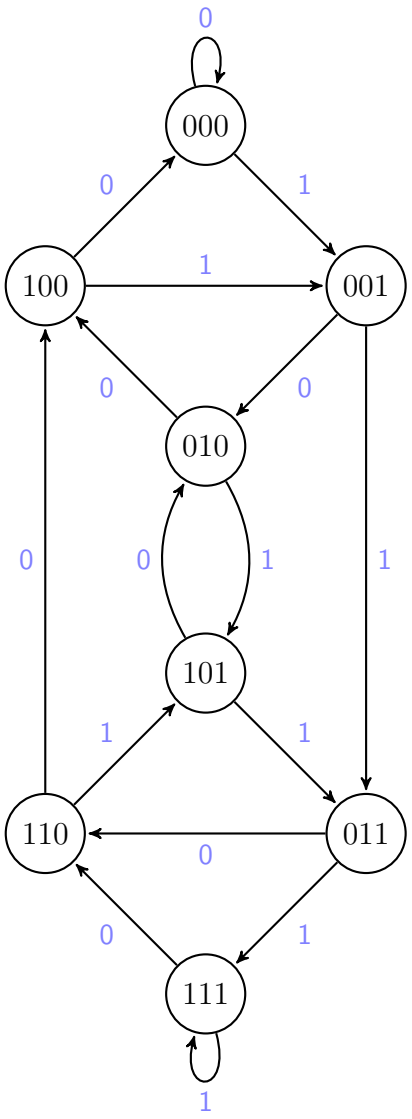
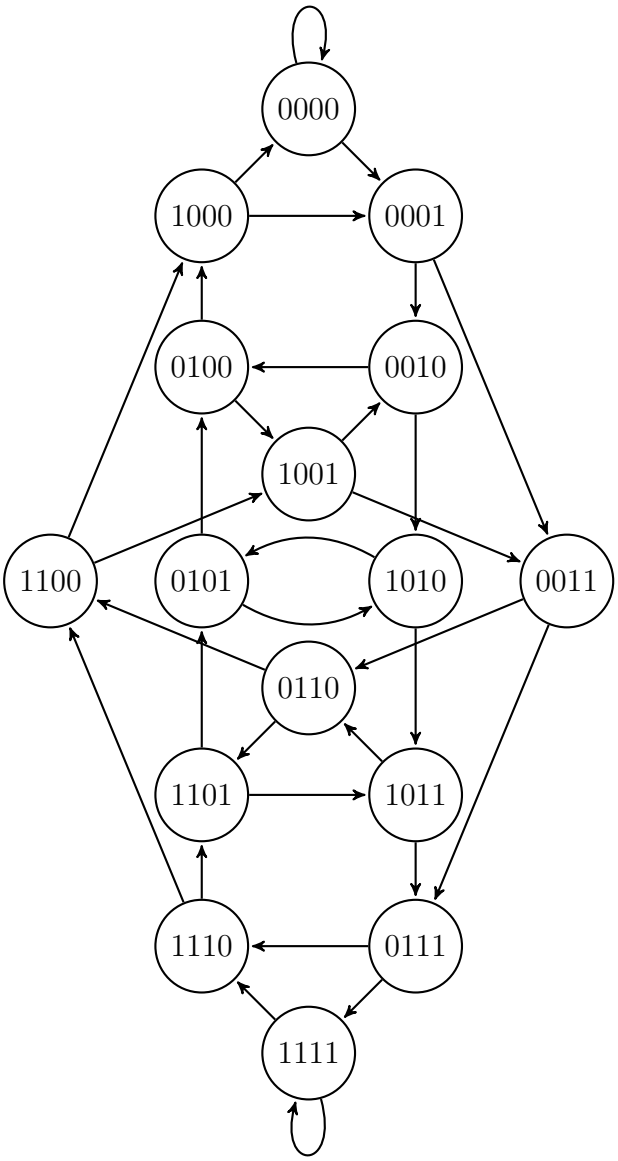


Figure 4.4: A binary de Bruijn graph of span 4. Any Hamiltonian cycle through this graph will describe a binary de Bruijn sequence of span 4. This graph can be formed from the line graph of the graph in figure 4.3. The line graph of a graph is the new graph created by taking edges from the original graph as nodes in the new graph. E.g. the node 000 in figure 4.3 has an outward edge labelled with a 0 and an outward edge labelled with 1. These edges become, respectively, the nodes 0000 and 0001 in the graph below. Edges between these nodes are then added whenever a single node in the old graph separated the edges from the old graph that have become nodes in the new graph.



A de Bruijn sequence over the alphabet $\{A, C, G, T\}$ that contains all possible u-tuples will have length,

$$4^u \tag{4.4}$$

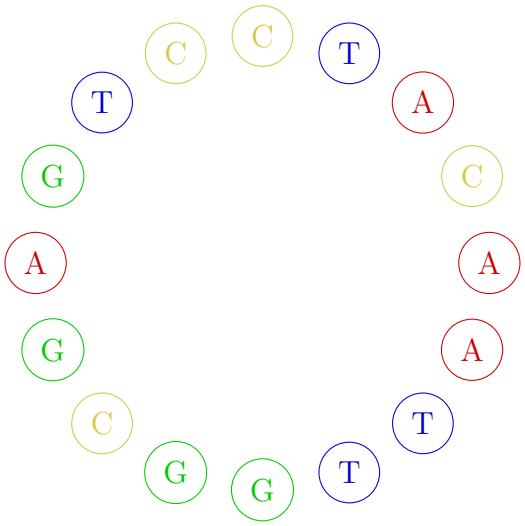
if viewed as a periodic sequence (Figure 4.5) or

$$4^u + u - 1 \tag{4.5}$$

when taken as a linear sequence. This follows from the Eulerian tour traversal of the de Bruijn graph,

- the traversal is known to exist since the out-degree equals the in-degree for every node
- the Eulerian tour formulation has c^{u-1} nodes and c out-edges for every node, hence there are c^u edges and the same number of symbols produced by a Eulerian tour traversal

Figure 4.5: The 4^2 symbols of a de Bruijn sequence covering all di-nucleotides. Each pair of nucleotides appears exactly once in the circular sequence. To represent each pair of symbols independently would require 2×4^2 new symbols. The saving is greater for de Bruijn sequences of greater span.



The total number of such sequences grows very quickly with the span of the sequence. The number of de Bruijn sequences over an alphabet of c symbols

with every u-tuple represented was determined by van Aardenne-Ehrenfest & de Bruijn [95] and is given by the formula,

$$\frac{(c!)^{c^{u-1}}}{c^u} \tag{4.6}$$

The number of c-ary de Bruijn sequences of span u that can be generated by a linear feedback shift register[1] are the same as the number of primitive polynomials of degree u over the finite field with c elements,

$$\frac{\phi(c^u - 1)}{u} \tag{4.7}$$

where ϕ is Euler's totient function [96]. For a protein binding micro-array containing all DNA u-tuples the ME array design (Section 4.1.1.3) uses a four letter alphabet and an algorithm on a span $u-1$ de Bruijn graph to construct its probes. The HK array design (Section 4.1.1.4) uses a binary alphabet and a span $2 \times u$ de Bruijn sequence, generated by a linear feedback shift register, to construct its probes. Table 4.1 illustrates the double exponential growth in the number of both general and linear feedback shift register type de Bruijn sequences. Properties of the binary linear feedback shift register type sequences that are of biological interest are discussed in Philippakis *et al.* [97]. These properties include that their sequences represent all 'gapped u-mers'. The authors in Philippakis *et al.* [97] argue that this constrained set of de Bruijn sequences are *just the right* set of sequences for the protein binding micro-array experiment. This point will be raised again in section 4.1.1.4.

[1] A linear feedback shift register over an alphabet can be pictured as an array, (register), of characters that, at each step are shifted one space over, (to the left say). The character that is shifted 'off the end' to the left is the output of the step. The character that is added to the right is determined by adding a fixed subset of the characters in the array. The fact that the characters are added determines that the register is of 'linear' type, i.e. if they were multiplied this would not be a non-linear shift register. When we speak of adding characters what is understood is that the characters are elements of a finite field. In our case the field with 2 elements or the field with 4 elements. The latter being identified with nucleotide alphabet.

4.1.1.3 The ME array design

The ME array design [68] has probes that are a compact representation of every possible 10-mer. The sequence used on the protein binding micro-array is not a de Bruijn sequence in the strict sense but contains either a 10-mer or its reverse complement exactly once. A 10-mer that is its own reverse complement occurs exactly once in the constructed sequence. The sequence is obtained by following a pseudo-Eulerian cycle on a de Bruijn graph whose vertices represent all possible 9-mers. By 'pseudo-Eulerian cycle' what is meant is that each time an edge is traversed its reverse complement edge is eliminated from the de Bruijn graph (Figure 4.6).

This approach requires only 20264 probes of length 35 according to Mintseris & Eisen [68] and therefore is a more compact representation of all 10-tuples than the HK design. The oligonucleotide arrays used in the DREAM5 data set allow 40630 probes and so the reverse of the pseudo-Eulerian sequence is also included on an array.[1] A meta-data file for the ME design protein binding micro-array, in the DREAM5 data-set, gives a template for each of the probes. By taking these probe templates it is possible to re-assemble a contiguous sequence. It appears that randomisation has been used in deciding whether a particular segment or its reverse complement is used as the template in each case. Because of this it is necessary to reverse complement about half the time when aligning the probes to re-assemble the sequence.

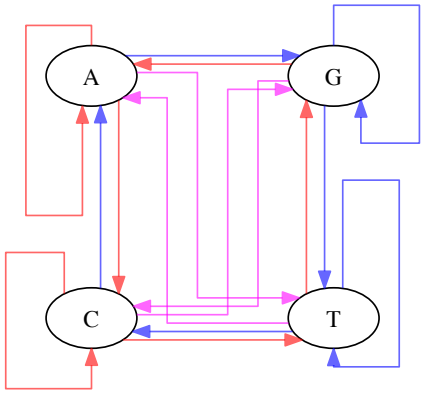
Empirically, for the DREAM5, ME design data, it is also observable that several 10-tuples are duplicated within the template sequences. This is contrary to the design specification. For instance, there are 788 duplicated 10-tuples on each of the forward and reverse sequences on the ME array design. It was initially unclear why this occurs. It is problematic when designing algorithms working with the sequences. It indicates that an optimally compact sequence, if such a construction is possible, has not been achieved in this case.

As it appears on the DREAM5 arrays, and compared to the HK design, a key difference for the ME design is that a DNA word of length 10 *and* its reverse

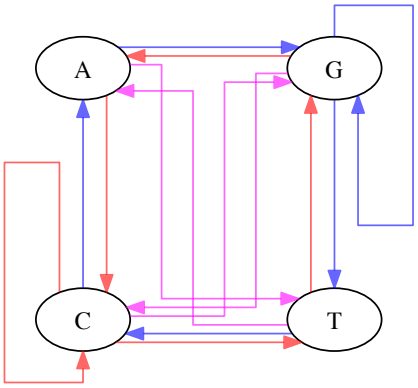
[1] Given twice the number of probes, i.e. 81260, the ME design would allow every 11-tuple to be represented on the array

Figure 4.6: Steps in the construction of a pair of pseudo-Eulerian tours

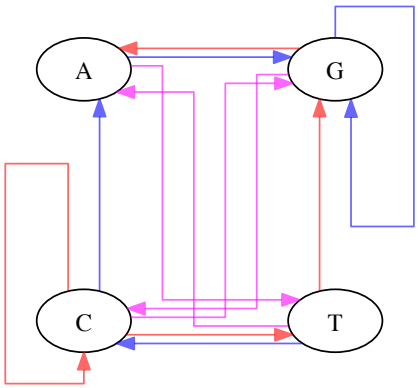
START



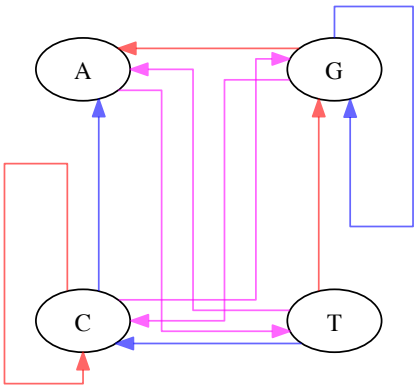
AA TT



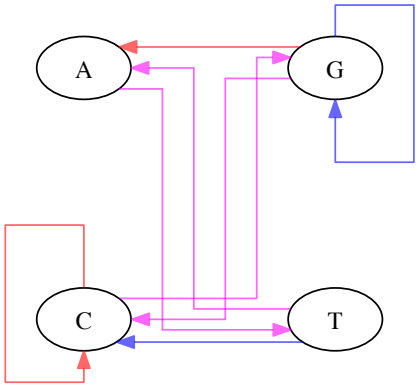
AC GT



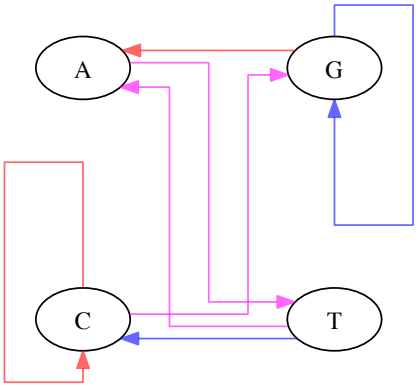
CT AG

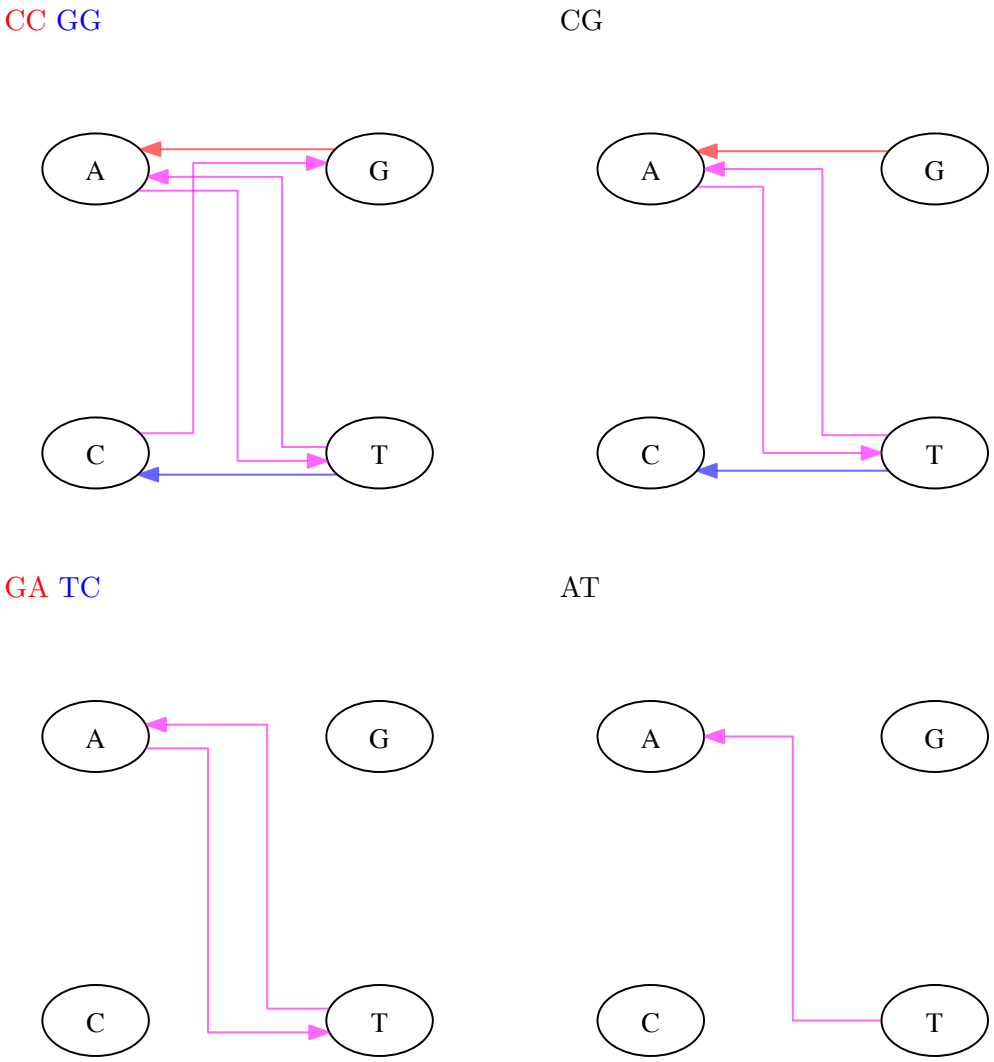


TG CA



GC

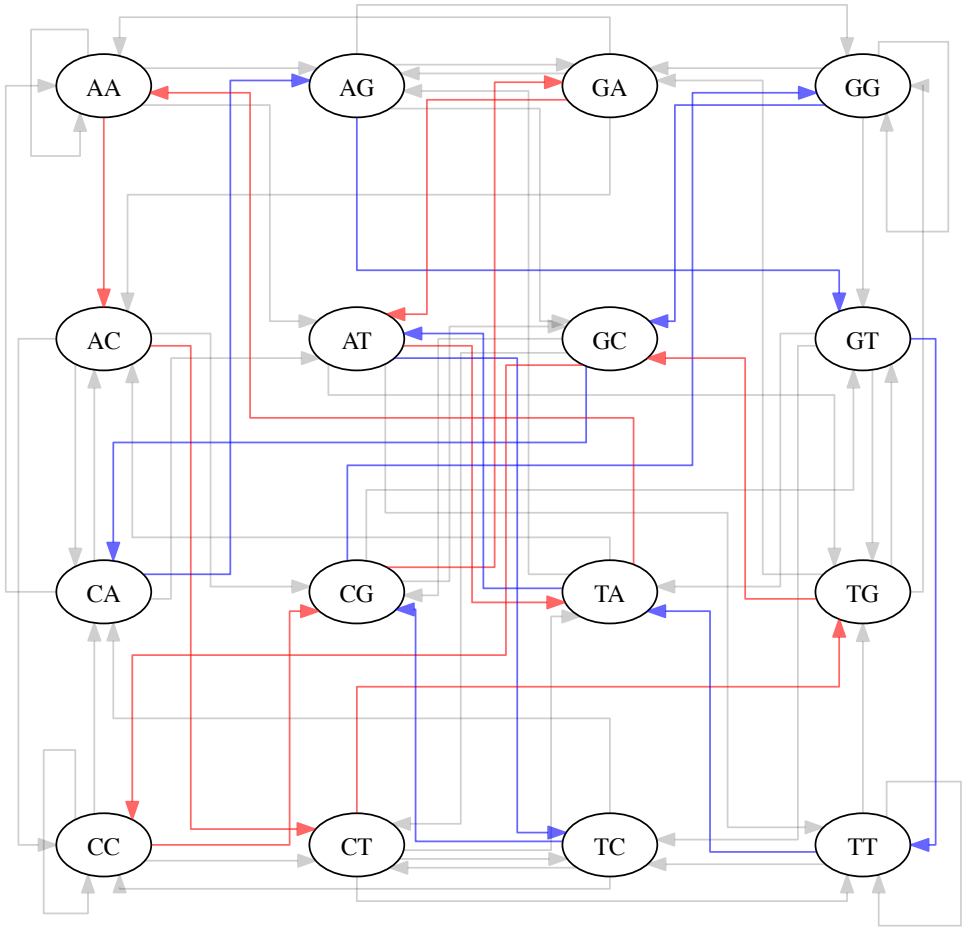




complement will not appear on the template strand. For example, if the 10 letter nucleotide word `AAAAAAAAAA` appears on the template strand of the ME array design then we know that `TTTTTTTTTT` will not appear on the template strand anywhere on the array[1]. This is in contrast to the HK array where every 10-mer appears on the template strand. This has implications for the modelling

[1]Since the reverse of the pseudo de Bruijn sequence is included on the ME array design. In this case no 10-tuple in the template side of the reverse sequence will provide the reverse complement of `AAAAAAAAAA` because `TTTTTTTTTT` is still in the non-template strand.

Figure 4.8: Figure showing a pair of pseudo-Hamiltonian tours for DNA two-tuples, i.e. the blue tour and the red tour. The graph, including the red, blue and grey lines, is a de Bruijn graph for all DNA two-tuples. Each of the two-tuples that is its own reverse complement, (the four central nodes), is visited on both of the tours. Otherwise each node is visited exactly once on exclusively the blue tour or the red tour.



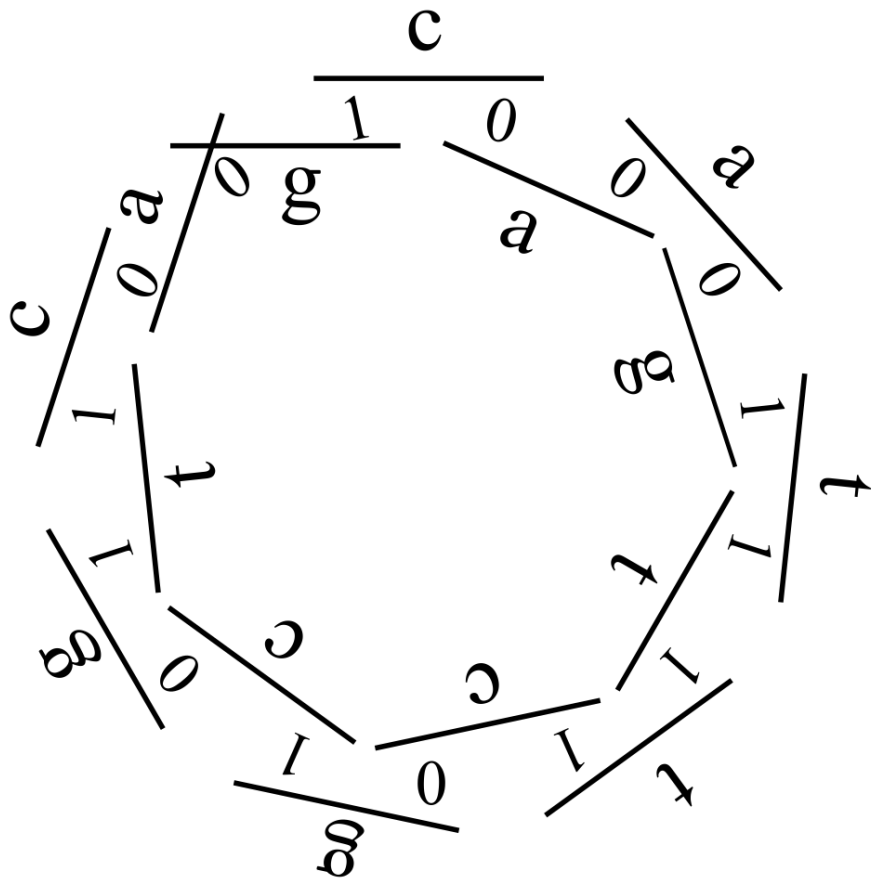
and prediction of an ME design protein binding micro-array and also for possible asymmetry in representation of binding sites caused by the incomplete double stranding of array probes (Section 3.1.1.3).

4.1.1.4 The HK array design

The HK array design [97] has probes that are the output of a linear feedback shift register.

When using a linear feedback shift register over the binary alphabet we must use a register of length $2 \times u$ if we wish to obtain all span u de Bruijn sequences over the 4-ary alphabet. Also, the 4-ary sequence must be read from each reading frame of the binary sequence. Since the length of the output from a linear feedback shift register, over the binary alphabet, always has odd period, this works in a natural way (Figure 4.9).

Figure 4.9: The HK array design approach to probe construction. In this example picture, taken from Philippakis *et al.* [97], we see binary pairs translated from each reading frame when using the output of a linear feedback shift register. Note the absence of the sequence 0000. This is not a valid state of the LFSR. Its omission guarantees the binary sequence will be odd in length and therefore that reading through both frames occurs naturally as shown. The missing 'a' nucleotide is added back afterwards to either the left or right of any existing 'a' in the constructed sequence.



Each possible 10-mer occurs exactly once in the de Bruijn sequence. This means that each 10-mer occurs exactly twice on the protein binding micro-array since the reverse complement is also present in each case. A 10-mer that is its own reverse complement should therefore occur 4 times on this design of protein binding micro-array.

The template strand probes are given in a meta-data file and from these it is possible to stitch back together the de-Bruijn sequence from which the probes

have been cut. 3481 of the probes have been reverse complemented from their original orientation in the de Bruijn sequence from which they have been taken. It appears that this has been done to avoid runs of 4 or more consecutive guanine residues in the template strand. Obviously this is not possible in all cases. The algorithm used has not been published.

In Philippakis *et al.* [97] it is shown that a maximal number of 'gapped 10-tuples' is present in a de Bruijn sequence generated in this way. This means that every 11-tuple, with some unspecified base at any internal position, will be present on the array. Another way of describing this is that for every 10-tuple there will be 9 11-tuples on the array that match with a single 'insertion'.

4.1.2 Feature Selection and the Model Matrix

Feature selection is done to obtain a set of probe sub-sequences that represent the content of any given probe. Features that describe the positions of sub-sequences within probes are also used.

The model matrix is used when learning parameters and predicting both probe intensities and genomic binding sites. The term 'model matrix' is taken from the standard expression for a linear model in statistics, i.e. the X in the following expression,

$$Y = X\beta + \epsilon \tag{4.8}$$

There is further discussion of this and similar linear models in section 5.2.3.2.

The matrix is designed using only the probe sequences, this is independent of any protein binding experiment results, hence the description finds itself in this chapter. Mathematical and computational methods for its construction are given in section 4.2.2. The results of some benchmarks and other investigations of this model matrix are given in section 4.3.2.1.

4.2 Methods

4.2.1 Array Probe Analysis

4.2.1.1 De-Bruijn sequence retrieval from probes

The probes for each design were analysed to gain a more complete understanding of their sequence content. The method used to reconstruct the de-Bruijn sequence from the probes is described in the introduction section 4.1.1.1.

4.2.1.2 Retrieval of generating polynomial

The Berlekamp-Massey algorithm was used to reverse engineer the generating polynomial representation of the de-Bruijn sequence of the HK array design. This was after transforming the DNA sequence to its binary form (Section 4.1.1.4).

4.2.2 Model matrix construction

The method used to form the model of the binding specificity to a probe is to split the probe into 'nucleotide words' of various lengths and then find weights for these words using a least squares penalty. This procedure, and some extensions, are described next.

4.2.2.1 Decomposing protein binding micro-array probes

Each probe is split into a number of features, let's call these features 'nucleotide words'. A toy example of a probe is used here to aid description. Consider the example probe,

```
AACGGTTT
```

We first split this probe into single nucleotides,

```
AACGGTTT
A A C G G T T T
```

then into di-nucleotides,

```
AACGGTTT
AA AC CG GG GT TT TT
```

then into tri-nucleotides,

```
AACGGTTT
AAC ACG CGG GGT GTT TTT
```

fours,

```
AACGGTTT
AACG ACGG CGGT GGTT GTTT
```

and so on,

```
...
```

The actual probes on the arrays have 35bp of variable sequence and a constant 25bp linker sequence. Different amounts of the linker sequence were included in decompositions. Since a protein could bind partly to the variable sequence and partly to the linker sequence it made sense to try incorporating part of the linker sequence into the model. If the longest DNA word being incorporated into the model is 10bp then we allow 9bp of linker sequence to be included in our decomposition since at least 1bp will be variable, and therefore provide useful information to the regression.

4.2.2.2 Encoding the presence of DNA words within probes

Each nucleotide word can be encoded as an integer using the lexicographical ordering,

$$\begin{aligned}
\texttt{A} &= 0 \\
\texttt{C} &= 1 \\
\texttt{G} &= 2 \\
\texttt{T} &= 3 \\
&\\
\texttt{AA} &= 4 \\
\texttt{AC} &= 5 \\
&\vdots \\
\texttt{TG} &= 18 \\
\texttt{TT} &= 19 \\
&\vdots
\end{aligned} \tag{4.9}$$

The inverse function, from DNA strings to integers, is easily computed allowing the recovery of a unique nucleotide word from an integer index, which is also the index of the word's column in the model matrix.

Each probe on the protein binding micro-array is described by a row of the model matrix. If a probe contains a particular nucleotide word then an entry will be made in the column of the model matrix corresponding to the nucleotides word's index. The particular entry that is made is a parameter that can be chosen from,

- the digit one, for one or more occurrence of the word within the probe
- the number of occurrences of the word within the probe
- in addition to the above, an entry in the column of the nucleotide word's reverse complement can be made

4.2.2.3 Model matrix implementation

The number of columns is calculated as,

$$4^m + 4^{m+1} + \cdots + 4^M = \frac{4^{M+1} - 4^m}{3} \tag{4.10}$$

where M is the longest DNA word used in the model and m is the shortest. The algorithm that converts the probe sequences into the model matrix takes a pair of parameters to select which features to include. A typical range, and one that often returned good results, was for $m = 3$ and $M = 8$.

The model matrix will have a nominal number of columns of ≈ 65000. The number of rows in the model matrix is the same as the number of probes, i.e. ≈ 40000. We see from this that a naive implementation would create a matrix with $\approx 2.6 \times 10^9$ elements. Taking 4 byte floats as elements we see that this pushes the limits of what can be done in RAM on a typical workstation.

The C++ language, linear algebra library Eigen [81] is used extensively throughout the data analysis presented in this thesis. An efficient, performant and convenient sparse matrix API is included. It is also convenient to interface with the FORTRAN language optimisation routines discussed later.

Construction of the model matrix from probe sequences is done at runtime by decomposing one probe at a time and looping over all the probes in an array. Storing the sparse matrix representation as a text format file is possible but this takes 100s of MB of storage. It turns out that the loading of the sequence probes from an sqlite database and on the fly sparse matrix construction can be done in about a second and this approach is adopted instead. Many variations of model matrices have been tried and the number of stored files quickly becomes unwieldy and hard to track using the former approach. Because probes can be flagged the number of rows in the model matrix varies between each array. Loading a serialised sparse matrix from file and removing unnecessary rows is also expensive and more complex than selecting the desired probes from the sqlite database. Some benchmark numbers for the settled upon approach are given in the results (Section 4.3).

4.2.2.4 Encoding a word's position within a probe

There is evidence that the location of a DNA word within a probe affects the intensity of the fluorescent signal for that word. In Berger *et al.* [80] it is claimed that increased intensity is recorded for a DNA word when it appears towards the end of the probe, i.e. furthest from the glass slide.

The model matrix appears to be of full rank for models including all 8bp DNA words (Section 4.3.2.2). Nevertheless further basis expansion seems to improve predictor performance. This might be interpreted in terms of offering the learning algorithm more relevant features to select from. The lasso predictor is not a linear function of the data and this must be a key ingredient since a simple linear regression could not be improved by adding extra columns to a matrix of full rank.

In order to pass information on word location to the learning algorithm, e.g. the lasso, it must be somehow encoded in the model matrix. One way to do this is to add a column for every possible DNA word and for every possible probe position. This would result in a model matrix consuming roughly 30 times more memory (Section 4.3.2.1) than the matrix that encodes only the presence, and possibly multiplicity, of a DNA word. Another approach is to add one extra column to the model matrix for each DNA word, but in this column add a value that encodes the feature's position within the probe. The latter approach is complicated by the fact that, particularly for shorter features, a DNA word may appear more than once in any given probe. In this case it must be decided which copy of the DNA word should have its positional information encoded. One approach is to accumulate the positional information by adding the values. Another approach would be to pick the location closest to one end, or the other, of the probe. Several functions have been tried to encode the positional information and these are described next (Figures 4.10, 4.11, 4.12 and 4.13). Which of these techniques is best will be addressed in the following chapter (Section 5.4.2). For each of the functions it needs to be decided what to do for features that occur multiple times in a single probe, simply adding the values is one approach.

Linear function of distance from probe end

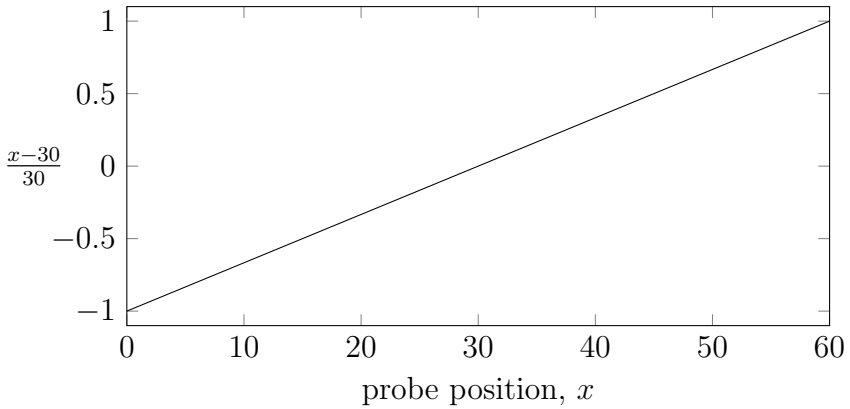


Figure 4.10: A simple linear function of probe position. A prior belief in using this function is that there is a linear decrease or increase binding propensity as one moves from one end of a probe to the other.

Tanh function of distance from probe end

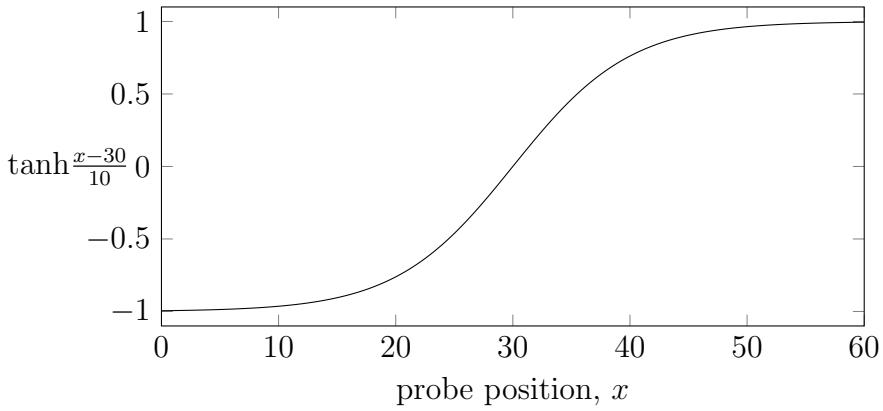


Figure 4.11: A hyperbolic tangent function of probe position. A refinement of the linear function, a prior belief in using this function would be that the location most critical at the line $x = 30$ and less important towards either end of the probe. As with the linear function, it assumes that one end of the probe has the opposite effect to the other.

Log of distance from probe end

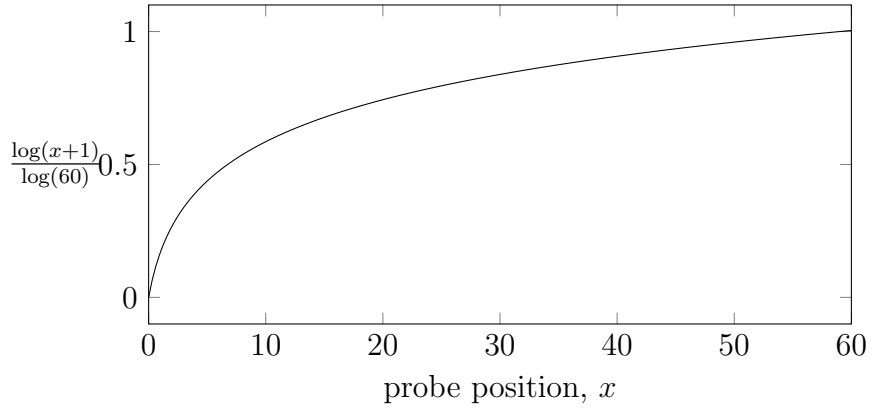


Figure 4.12: A log function of probe position. This function can be reflected in the line $x = 30$ to yield another candidate position encoding that was experimented with. The prior belief is that one end of the probe has the opposite effect to the other. Here though, the position changes are more important at one end than the other.

Triangle function of distance from probe end

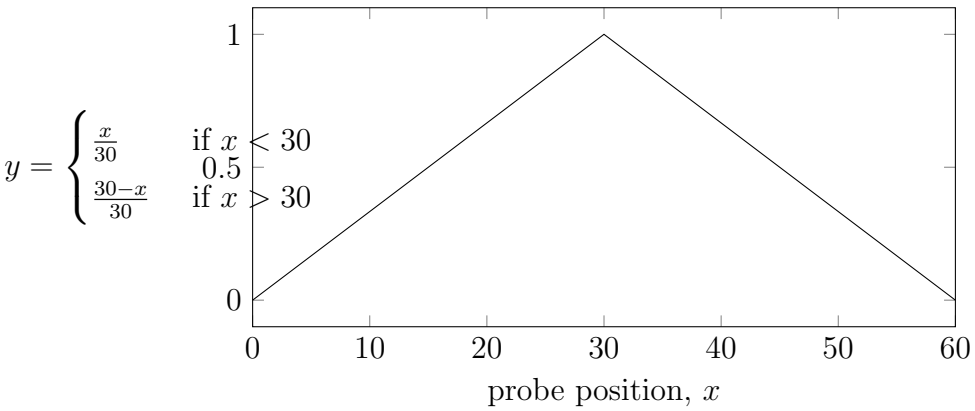


Figure 4.13: The triangle shown here is symmetric and centered at $x = 30$, asymmetric triangle functions, centred at other probe positions were also tried. This function allows that there might be a 'sweet spot' somewhere on a probe for a binding site to occur. The prior belief could be that positions close to the array are sub optimal due to steric hindrance and positions at end of the probe furthest from the array are also sub-optimal for binding due to the increased probability that a probe has not been successfully double stranded.[1]

4.2.2.5 Locating word positions

During analysis and visualisation of protein binding micro-array data a mapping between individual DNA words and their containing probes, and vice versa, was frequently wanted. A way to select all intensities corresponding to a particular DNA word is available through the model matrix (Section 4.2.2). Essentially, one can travel down a column of the sparse matrix and find all the non-zero rows. These rows correspond to the probes in which the DNA word appears. This does not give information on the position within the probe though.

A way to find the position of a DNA word within its containing probes is a 'decoding' of the de Bruijn sequence. What is meant by decoding in this context is the ability to efficiently locate a DNA word in the de Bruijn sequence, and therefore within any of its containing probes. Efficient methods exist to decode de Bruijn sequences generated from linear feedback shift registers [98]. A description of the particular shift register used in the HK design was calculated and this is given in the results (Section 4.3.1.2). This method was not used but its potential utility is interesting to note and could be useful in building a more complex model.

Alternatively, we can keep a mapping from every possible DNA word of a given length to its locations on the protein binding micro-array. If we encode all 10bp DNA words in this way then we require a 32bit integer for every word and integers for each location of the word on the micro-array. Using an efficient data-structure it is possible to store such information in a few MB of RAM. In contrast to the LFSR decoding approach, this approach works for the ME array design as well. Storing the locations of shorter words presents a different problem; a typical 3bp word might appear in a large proportion of probes. In this case we need to keep lists of 10s of thousands of probes and their locations within probes for every 3bp sequence.

[1] In LeProust [77] the yield of full length cDNA probes is given as 80%. In this case, fewer than 80% of double stranded DNA probes on a protein binding micro-array will have a double stranded binding site fully formed at the probe end most distant from the array.

Table 4.1: The number of de Bruijn sequences for alphabet sizes 2 and 4, for general and linear feedback shift register types, for a selection of spans.

DNA word length		linear	general		linear	general
	span	$\|\Sigma\| = 4$	$\|\Sigma\| = 4$	span	$\|\Sigma\| = 2$	$\|\Sigma\| = 2$
1	1	2	6	2	1	1
2	2	4	20736	4	2	16
3	3	12	$\approx 10^{20}$	6	6	67108864
8	8	4096	$\approx 10^{22000}$	16	2048	$\approx 10^{9800}$
9	9	15552	$\approx 10^{90000}$	18	7776	$\approx 10^{39000}$
10	10	48000	$\approx 10^{360000}$	20	24000	$\approx 10^{157000}$

4.3 Results

4.3.1 Probe Analysis

4.3.1.1 Comparison of number of de-Bruijn sequences for each array design

These quantities are calculated for alphabets of size four and two using equations 4.6 and 4.7. A point of interest (Table 4.1) is that the number of relevant de Bruijn sequences that can be obtained via doubling the binary alphabet is smaller than that that can be obtained using the 4-ary alphabet. The number of linear feedback shift register type sequences is constrained further. The sequence space for the binary linear feedback shift register de Bruijn sequences is relatively small compared to that of the more general 4-ary sequences. It may be that the de Bruijn sequences available from the binary linear feedback shift register construction are sufficient to represent the biologically significant sequence information on the micro-array platform.

4.3.1.2 HK design LFSR recovery

One of the results of this chapter is the 'stitching together' of the sequences of the HK and ME designs as described in the introduction (Section 4.1.1.1). For the HK design this allowed the creation of a single circular sequence that was the de Bruijn sequence used in the HK array construction, apart from the reverse complemented probes. It was not possible to predict when a probe would be

reverse complemented apart from those cases that had poly-guanine sequences. These were always reverse complemented to contain a poly-cytosine instead. An algorithm for this has not been published. It also seems that some arbitrary decisions have to be made, e.g. what happens if a probe sequence has a poly-guanine at the beginning of the probe and a poly-cytosine at the end? Taking the reverse complement results in another poly-guanine.

The fact that efficient algorithms are available for decoding shift register generated sequences is stated in section 4.1.1.2. Running the Berlekamp-Massey algorithm [99] on the sequence obtained from the data file, and assuming the binary scheme of encoding, the following characteristic polynomial was obtained,

$$x^{21} + x^{13} + x^{11} + 1 \tag{4.11}$$

This shift register successfully reconstructs the HK de Bruijn sequence, up to the reverse complemented probes. Establishing this LFSR description of the HK de Bruijn sequence was partly verification of the sequence content of the HK array and was in itself a worthwhile task. The utility[1] of having the LFSR available has not been exploited any further in this thesis though.

4.3.1.3 ME sequence type construction

An effort was made to construct this type of 'pseudo de Bruijn sequence' following the specification of the ME design [68].

If we require all 10bp nucleotide sequences to be represented in a sequence exactly once, allowing for reverse complements, then the number required is as follows,

$$4^{10} = 1048576 \tag{4.12}$$

$$4^{5} = 1024 \tag{4.13}$$

$$(1048576 - 1024)/2 + 1024 = 524800 \tag{4.14}$$

i.e. we divide all the 10-tuples that are not their own reverse complements by 2.

[1] This information is sufficient to construct the mapping from DNA words to probe location in a compact and efficient manner [98].

We then add back all those that are their own reverse complement. This gives a total sequence length of 524800. With 26 independent base pairs on a probe this would require $524800/26 = 20185$ probes.

The following observations were made,

- This number is less than what is stated by the authors in Mintseris & Eisen [68].
- There are several 10bp words that appear more than once on the ME array design.

The authors in Mintseris & Eisen [68] do not give a complete description of their algorithm. There seems to be some confusion between Eulerian cycles and Hamiltonian cycles and the construction is incorrectly stated to contain each k-mer[1] exactly once. Whilst writing this thesis a paper that addresses the ambiguity in Mintseris & Eisen [68] was published [69]. The authors give accurate bounds for the efficiency of the 'psuedo Eulerian' construction and a description of an optimal algorithm.

As part of earlier investigations into this matter explorative scripts were written that generate ME type sequences along the lines of the description given in figure 4.6. Basic components of the algorithm,

- Depth first graph traversal
- Tests for connectedness
- Tests for strongly connected components
- Counting connected components
- Eulerian tour detection

were all implemented. The development of this code was suspended when it was found that the description in Mintseris & Eisen [68] was ambiguous and essentially not possible to implement as stated.

[1] k being 10 in our case.

4.3.2 Model Matrix

4.3.2.1 The sparse model matrix

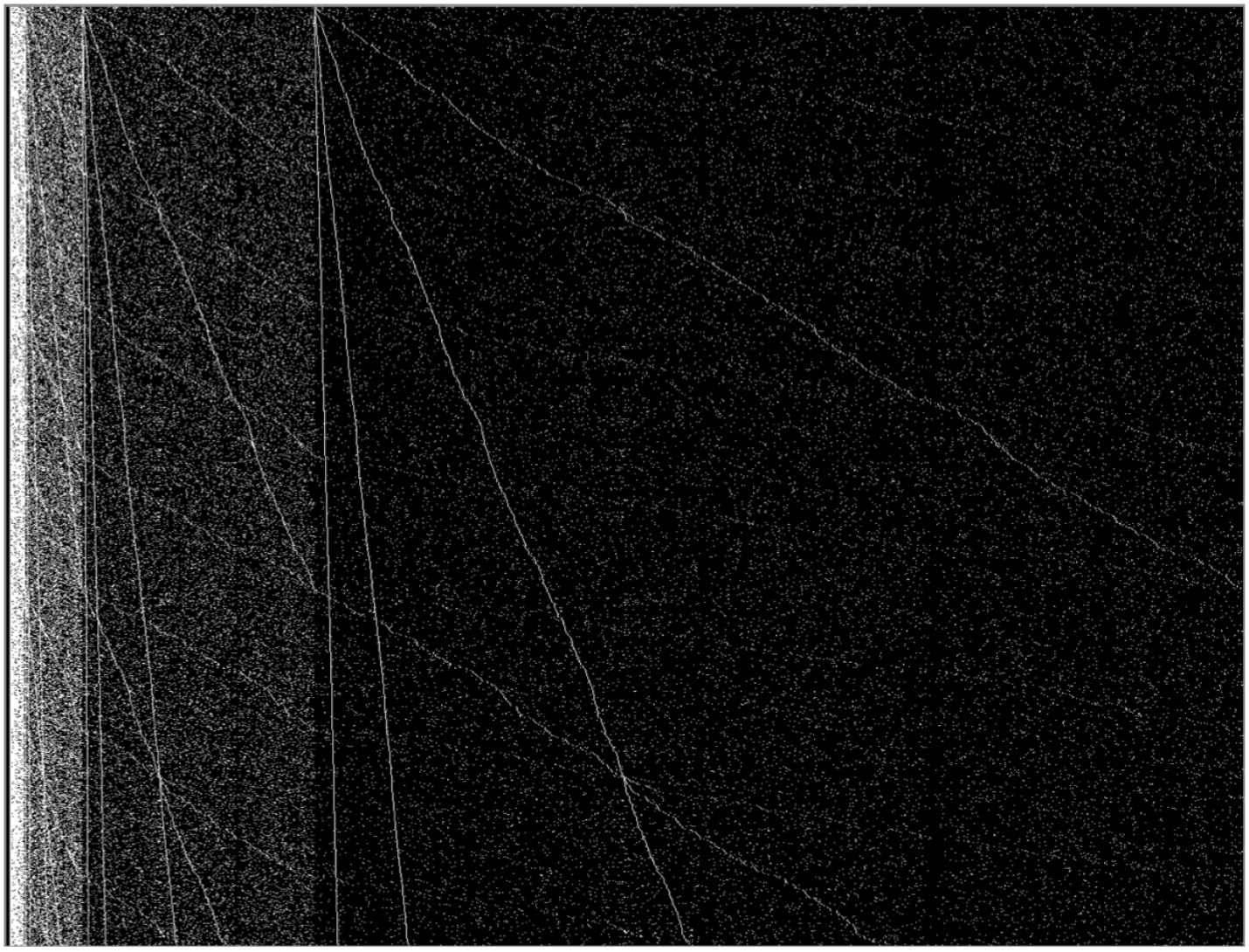
Figure 4.14 gives a visual representation of one of the sparse model matrices. This is, in fact, just the top left corner of a matrix but it does give the correct impression for the entire matrix. Each column of white dots corresponds to a DNA word contained in the array's probes, each probe being represented as a row. For example, the first and most dense columns represent the presence of DNA 2-tuples in each probe. It is a nearly solid block of white pixels but not entirely; in any given probe all DNA 2-tuples will not necessarily be present. This is a consideration when making a model; if multiplicities of DNA 2-tuples are not counted then many probes will have the same DNA 2-tuple description from the perspective of the regression. On the other hand, if we record the numbers of each DNA 2-tuple, in each probe, in the model matrix then each probe is more likely to have a distinct 2-tuple description. The amount of memory used to construct a model matrix depends upon the number of non-zero elements in the matrix. The memory requirements for a model matrix also depends upon whether it uses column major or row-major storage to a small extent.

Assuming 45bp of each probe word is decomposed into each row of the model matrix we will have 4 possible non-zero entries for 1bp features, 16 non-zeros for 2bp words, up to 43 non-zero entries for 3bp words, 42 for 4bp words and so on, up to 36 non-zero entries for 10bp words. Due to the properties of the de Bruijn sequences we know that there will be 36 distinct entries for 10bp words. On the other hand we might not have 43 distinct 3bp words in any given probe and so will have fewer non-zero elements in the corresponding row of the matrix. If we include all 4bp words through to 10bp words then a maximum number of non-zeros per row is,

$$43 + 42 + 41 + 40 + 39 + 38 + 37 + 36 = 316 \tag{4.15}$$

For 40000 probes this would imply 12640000 non-zero entries. Upon construction of a model matrix for a typical array the empirical numbers of non-zero elements are given in the second column of table 4.2. The memory requirements in table 4.2

Figure 4.14: Sparse matrix representation. Each non-zero entry in the matrix is represented as a white dot. The diagonal lines that can be observed are an artefact of the probe sequences being sorted lexicographically before being decomposed into the model matrix.



Word sizes	Non-zeros	Memory used	Time to construct
4-8	7 700 000	92MB	1.1s
4-9	9 200 000	111MB	1.38s
4-10	10 600 000	127MB	1.4s

Table 4.2: Data structure characteristics of a typical model matrix for a selection of contained features' sizes.

are calculated as the number of non-zero elements multiplied by 12 bytes. The 12 bytes consist of 8 bytes for the value and 4 bytes for the value pointer. The row pointers have been disregarded since they are a relatively small component[1].

4.3.2.2 Observations on model matrix rank

An important consideration is the rank of the model matrix. Since we are working with a regularised linear model it is not a problem that the number of columns is greater than the number of rows, '$n < p$'. Repeated, identical columns are also not a problem for the algorithm in the same way they would be in a typical '$n \geq p$' linear regression setup.

If we include long enough word lengths then the matrix will have full rank and we will be able to obtain a zero-residual error solution to the problem. e.g. if we allowed all 10bp DNA words, then each probe would be 'labelled' uniquely by some 10bp word and the weight for each of these 10bp words can be allowed to equal the corresponding intensity. Another way to picture this is that, taking the matrix in figure 4.14, under some permutation of rows we would see a $\approx 40000 \times 40000$ identity matrix appear within the larger matrix.

On the other hand, it is possible to have many more columns in the model matrix than rows and yet still not obtain a zero residual. This is caused by the matrix having a column space of dimension less than the number of rows.

Model matrices were constructed that included all 10bp DNA words, this required 4^{10} columns in the model matrix and ≈ 40000 rows. Because this matrix is sparse it was handled with relative ease.

The SuiteSparse QR matrix decomposition library [100] was used to obtain the ranks of some typical model matrices. The actual rank of a matrix can depend upon how many flagged probes there are for an array. The calculated ranks shown in table 4.3 show that the rank of the matrix for a model including all 7bp DNA words has rank less than that of the number of columns/DNA words, but only by 20.

The rank of the matrix for 1-8bp matrix is equal to the number of rows for

[1]The sparse matrix representation is the standard setup with an array for values, a same sized array for value indices and a third array giving row or column indices.

Word lengths included	Number of words	Matrix rank
1-6	5460	5460
1-7	21844	21824
1-8	87380	39754

Table 4.3: Ranks of model matrices for a selection of contained features.

this array, i.e. there are 39754 unflagged probes on the array from which this matrix was built. For a standard linear regression we would be able to fit the data with zero residual error in this case.

4.3.2.3 Data retrieval performance

The sequence data and intensity data are stored in normalised sqlite database tables. The ODB object relational mapping translates between the database tables and the retrieved objects. The sequences are retrieved into STL[1] vectors of strings and the intensities are retrieved into STL vectors of doubles. The intensities are being pulled from an indexed table of approximately 7 million rows.

- Retrieval of the filtered probe sequences for a pair of arrays takes about 0.3s
- Retrieval of a set of $\approx$ 40000 intensities from the table of 7 million takes about 0.15s
- This rate of data retrieval allows interactive exploration.

4.4 Conclusion

In this chapter a description of the probes from two protein binding micro-array designs has been given. Some measurements of their properties have been made that have not been explicitly stated elsewhere. A comparison of the two sequence designs is also made in more detail here than elsewhere.

[1]Standard Template Library of the C++ programming language.

A way of modelling the probe sequences is introduced that will be used in chapters 5 and 6. The sparse model matrix, as described above, could be extended to allow this model to be implemented, using reasonable compute resources, for greater DNA word lengths if a protein binding micro-array with greater DNA word lengths were developed.

The adding of extra columns to the model matrix that 'encode positional information' is also described. This is found to have a positive effect on the performance of a predictor built from this model. The prediction is the topic of chapter 5.

The HK designed array has desirable properties that make it the favourite of the author. These properties include a very concise description (Section 4.3.1.2) and the availability of efficient decoding algorithms.

The development of an algorithm to efficiently navigate the de Bruijn sequence of the HK array design using a 'decoding method' was not completed. This would have been useful in creating a more detailed, yet efficient, model of probe position effects and would have utility in drawing visualisations of the data. The remaining work to be done is to implement the decoding algorithm described in Paterson [98] whilst accounting for the idiosyncratic 'flipping' of probes in this data-set.

The algorithm for the design of the ME type array has been more clearly described in Orenstein & Shamir [69]. The source code for this software is not available, only a Java binary that the authors claim produces such sequences. Writing an efficient, open-source, implementation in C would be an interesting and useful extension of the Python code described at the end of section 4.3.1.3.

Chapter 5

Model Fitting and Array Prediction

In this chapter we use the model matrix (Section 4.2.2) and the data-set published as part of the DREAM5 competition (Section 5.2.1) to fit parameters and predict held back data. We compare the performance of our predictor to others from DREAM5. We also look at sets of DNA words selected in some of the models and comment on what we might infer on DNA binding behaviour. Models with enforced sparsity are demonstrated to provide interesting perspectives on the data (Section 5.4.2.2).

In this chapter there is also a demonstration of a simple alignment approach (Section 5.4.1) that gives us a feel for the data and what can and cannot be achieved before resorting to more complicated methods.

5.1 Chapter Outline

In the background section of this chapter (Section 5.2.1) there is a presentation of the results of the DREAM5 challenge. There is a brief discussion of an algorithm that was used to make an entry to the competition, this will be referred to as 'the RVM entry'. Following this (Section 5.2.3) is some background on the lasso algorithm that was used in the model fitting procedure and in making the predictions that are the subject of the rest of this chapter.

In the methods (Section 5.3) details of the use of the lasso optimisation routine are given along with its use for prediction. There is a brief discussion of the methods used to do the multiple sequence alignment.

In the results section there is a presentation of the simple alignment approach to finding significant subsequences in protein binding micro-array probes (Section 5.4.1). Following this are a number of representations of the output of the lasso predictor (Section 5.4.2).

In the conclusions a review of the results of the lasso predictor is made (Section 5.5).

5.2 Background

5.2.1 DREAM5 protein binding micro-array prediction challenge results

The DREAM5 data-set was released as part of a prediction challenge. The data-set was originally released as a set of 20 pairs of protein binding micro-arrays for training purposes. These arrays were for each of 20 proteins on an array of each design, (counterpart arrays). A further 66 arrays were provided with the identity of each array's protein concealed and the intensity readings for one of the counterparts concealed. The 66 arrays were 33 of the HK design and 33 of the ME design (Sections 4.1.1.3 and 4.1.1.4). Teams submitted predictions for each of the 66 arrays, each probe having to be assigned its predicted intensity.

Predictions were ranked according to the following metrics,

- Pearson correlation of predicted intensity against true value
- Pearson correlation of predicted log intensity against true value
- Spearman correlation of intensity against true value
- 8mer area under precision recall curve
- 8mer area under receiver operator curve

Team	Aggregate Rank	Probe Evaluation Pearson	Probe Evaluation Spearman	Probe Evaluation PearsonLOG	8mers Evaluation AUPR	8mers Evaluation AUROC
696	1	0.6413	0.6394	0.6742	0.6997	0.9942
824	2	0.6103	0.6555	0.6732	0.5446	0.9764
690	3	0.6375	0.6735	0.6936	0.5223	0.9524
863	4	0.5728	0.5735	0.6207	0.6739	0.9942
662	5	0.6117	0.6227	0.65	0.5244	0.9649
689	6	0.5814	0.6921	0.6475	0.306	0.9395
853	7	0.4688	0.3669	0.4171	0.6759	0.9906
763	7	0.5177	0.4837	0.5227	0.5304	0.9747
755	9	0.4973	0.5617	0.575	0.2484	0.9405
775	10	0.5335	0.431	0.4605	0.5837	0.9246
873	11	0.4612	0.5313	0.5402	0.1559	0.9304
872	12	0.4611	0.5382	0.5438	0.1503	0.9293
730	13	0.2667	0.1003	0.189	0.4617	0.8908
71	14	-0.0002	0.0004	0.0004	0.003	0.487

Table 5.1: Results for the 14 entries to the DREAM5 challenge

The last two metrics were based upon a 'gold standard set of 8mers' that it was deemed should be predicted. These gold standard sets were of varying sizes for each array and had an unknown provenance. Due to the apparent arbitrariness of this metric it will not be discussed much further here. It has been observed, here and elsewhere, that PWM models fare worse under these metrics than models based on dictionaries of DNA words [71]. The lasso model, presented later (Section 5.2.3.2), predicts these gold standard sets better than other models as well as out-performing other methods in the correlation metrics.

Table 5.1 gives the results, as reported, for the original challenge. The relevance vector machine (RVM) entry is team 755 and the winning entry, team 696 is that of Annala *et al.* [70]. The winning entry to the challenge was that described in Annala *et al.* [70]. This model is discussed further in section 5.2.3.1.

5.2.2 Background to the RVM algorithm

An entry to the competition was made using the machine learning software described in Down [101]. The relevance vector machine uses a combination of position weight matrices, learned with the NestedMICA motif finding tool [65,

102] and statistics of sequence composition, such as di-nucleotide counts. A background model is trained in advance and the algorithm optimises over the parameters in the matrices.

This approach was very computationally expensive; it took 48 hours for the algorithm to run to completion on some arrays. Problems such as the Java virtual machine running out of memory or mysteriously aborting on certain compute farm nodes made the process somewhat laborious and hard to parametrise and test.

The predictor's performance was also not particularly good and hence was not pursued any further.

5.2.3 Background to lasso algorithm

5.2.3.1 The linear model of Annala et. al.

The winning entry to the DREAM5 protein binding micro-array prediction challenge was that of Annala *et al.* [70]. The regularisation method used by this group was to include only 7bp and 8bp words with the highest median intensities averaged across their containing probes. As stated by the authors, an assumption in this approach is that the probes with the highest signals are the "most informative in terms of protein binding". This is a natural assumption, though the alternative point of view, i.e. that we might learn something about protein binding from the particular set of probes that have low intensity, is also an interesting question. All 4, 5 and 6bp words were included in their model. The authors suggest that the lasso would be an alternative regularisation method but that it could not be made to "run in a practical amount of time for a system of this scale", although this is contrary to the findings presented in section 5.2.3.2.

5.2.3.2 Lasso model

The lasso linear model shares some similarities with the linear model described in section 5.2.3.1. This model is obtained by optimising the objective function in equation 5.1. This is the usual least squares objective function. Provided that the model matrix X is of full rank the parameter vector β can be obtained by solving a linear matrix equation.

In the following the L_1 norm, $\|.\|_1$, is given by $\Sigma|x_i|$ and the L_2 norm, $\|.\|_2$, is given by $(\Sigma|x_i|^2)^{1/2}$. In each of the objective functions Y is the vector of probe intensities and X is the model matrix.

$$\|Y - X\beta\|_2^2 \tag{5.1}$$

In application to our protein binding micro-arrays, if it is desired to obtain a weight for all possible octomers, then the model matrix will not have full rank; for an experiment with 40 000 measurements/probes there would be $4^8 = 65536$ parameters to fit. This is often referred to as the case $p \gg n$. Methods to overcome this problem are collectively known as regularisation.

Ridge regression (Equation 5.2) is a technique used for the purpose of regularisation. It does not offer sparsity, e.g. all octomers would have non-zero weights, but it has the benefit of resolving to a linear matrix equation (Equation 5.1).

$$\|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2 \tag{5.2}$$

The lasso [103] offers regularisation and also sparsity to the parameter vector, β, in the objective function (Equation 5.3). The difference between lasso regression and ridge regression is that the penalty on the parameters is the L_1 norm, rather than the L_2.

$$\|Y - X\beta\|_2^2 + \lambda\|\beta\|_1 \tag{5.3}$$

The objective function (Equation 5.4) is the elastic net penalty [104]. It combines the ridge penalty and lasso penalty via an extra parameter α. With α set to zero we have the lasso model.

$$\|Y - X\beta\|_2^2 + \lambda(\alpha\|\beta\|_2^2 + (1-\alpha)\|\beta\|_1) \tag{5.4}$$

The lasso will set one of a pair of linearly related variables to zero, giving a non-zero weight to only one. Making the α parameter positive can mitigate this behaviour if it is thought undesirable. Whilst this option has been extensively tested, no clear conclusions on its utility have been made and the data presented in this report should be assumed to have the α parameter set to zero.

The λ parameter in equations 5.2, 5.3 and 5.4 is a 'tuning' parameter that allows control of the penalty imposed on the parameter vector β. This is obtained by cross-validation, the usual 10-fold scheme was used here, as described in Hastie *et al.* [88]. In our case this means taking the approximately 40000 probes on an array, select 36000 to train with and predicting the remaining 4000 probes with varying values λ. In fact the glmnet algorithm[1] fits a range of values of λ by default and so it is more a case of *selecting* the best value of λ from those [89].

5.3 Methods

5.3.1 Alignment methods

A simple approach that we can take to explore the array data is to perform alignments. This adds an interesting perspective to the data, two different methods of increasing complexity are used,

- alignment of entire probes
- alignment of words from probes

A description of the algorithms for these two approaches is given next. The alignments themselves perhaps give a clearer picture of the methods and these are given in the results (Section 5.4.1).

For the first approach we select the 10 probes with the highest intensities and pass them to the alignment algorithm.

For the second approach we take every possible 8bp word[2], find all the probes in which it occurs, and then calculate the mean score for the word. We select the 8bp words with the highest scores and do a multiple alignment upon these words.

[1] 'glmnet' is the name given to the algorithm that optimises the objective function in equation 5.4 by its authors.

[2] We might assume that how much a protein binds to a probe depends upon whether a particular 8bp sequence is present in that probe.

5.3.2 lasso methods

An efficient, FORTRAN language, implementation of this procedure is available [89]. This computes the lasso path solution in a few seconds for each microarray.

The algorithm computes a sequence of models of decreasing sparsity using a cyclical coordinate descent. The Intel ifort FORTRAN compiler was used to compile the source which was linked with a C++ wrapper. The original code is written in MORTRAN rather than FORTRAN. MORTRAN has FORTRAN as a compile target and appears to offer some syntactic sugar to FORTRAN. The FORTRAN code was linked against the C++ wrapper relying on the GCC ABI surrounding argument passing and symbol name mangling.

The same model matrix design that is used to model the probes on the training array is used to predict the probe intensities on the counterpart array. For instance if the training model matrix contains columns for positional information then the model matrix for prediction will have these columns too. The probes on the counterpart array are decomposed as described in section 4.2.2.1. The model matrix obtained in this way is then multiplied by the parameter vector, fitted using the training array, to predict the counterpart intensities.

Probes that are flagged are not used in the construction of the model matrix and so the model matrix will have fewer rows if the training array has flagged probes.

When doing cross validation the probes on a training array are split in to 10 randomly assigned groups. For each of the groups the probes are predicted by training a model on the 9 other groups. In this way we estimate the performance of the predictor on the counterpart array and, importantly, select the amount of sparsity that is best to avoid over-fitting.

5.4 Results

5.4.1 Alignment results

The objective of the presentation is to show a very simple and intuitive way of looking at the data. The data presented in this section is limited to that of

```
TGACGTCA 19.147
ATGACTCA 18.8728
TGAGTCAT 18.8267
TGACTCAC 18.8084
GATGACGT 18.8003
TTACGTCA 18.7382
ATGAGTCA 18.7301
TGACGTAA 18.6948
TGACTCAT 18.6944
ATGACGTA 18.6578


CLUSTAL 2.1 multiple sequence alignment


3               --TGAC-TCAC 8
8               --TGAC-TCAT 8
1               -ATGAC-TCA- 8
2               --TGAG-TCAT 8
6               -ATGAG-TCA- 8
0               --TGACGTCA- 8
5               --TTACGTCA- 8
4               GATGACGT--- 8
9               -ATGACGTA-- 8
7               --TGACGTAA- 8
                  * *  *
```

Figure 5.1: Alignment of the highest scoring 8bp sequences for the Jundm2 transcription factor.

a single protein. This was the first thing done in the investigation of the data. Hundreds of similar examples were created and could be given here but this single example gives a fair idea of what can and cannot be done using this approach.

Figure 5.1 shows the highest scoring 8bp sequences for the transcription factor Jundm2 taken from PBM data published in Badis *et al.* [105].

Figure 5.2 shows a multiple alignment of the 10 probes with the highest intensity readings for the same transcription factor. We see that most of the 10 highest scoring probes on this array contain the 8bp sequence, or a single substitute thereof, that we might have expected from figure 5.1. It would be reasonable to assume that such an alignment would not be possible, based upon the assump-

```
CGTATGACGTCACTTTGAATGTCCGGCGAGCCGTA 19.9169
CTTCATGTGACCGTGACGTCACGCCATCTCTCTCA 19.9111
CACAATGACGTCATAAATGAGGTGACATAGAGCTG 19.9108
ATTGATGACGTCAGGGGATACGTATTCGCTCACAC 19.9101
GTCATTACGTCACCTGCCCGACTGGGAGGATGATT 19.896
TAGTGAGATGACGTAATTGGGACCATTCAGCGTGT 19.8826
TGTCGATGATGTCACAAGATTTGAGTATCTTTTCG 19.8652
ACCTTGGAGATGAGTCATCCTGACTCCCTCAAGCC 19.8297
CACGCAGAAGTCTTACGTCATCCACCGTGCAGTCC 19.8155
CCAGGCATGATGTCATTTCCCTGGACGTCTCCCAC 19.812

CLUSTAL 2.1 multiple sequence alignment


0               --------CGTA------TGACGTCACTT---TGAATGTC-CGGC---GAGCCGTA--- 35
2               --------CACAA-----TGACGTCA------TAAATGAGGTGACATAGAGCTG----- 35
4               ---------GTCA-----TTACGTCACCTGCCCGACTGGGAGGAT---GATT------- 35
1               CTTCATGTGACCG-----TGACGTCAC-----GCCATCT-----------CTCTCA--- 35
6               -------TGTCGA-----TGATGTCAC-----AAGATTTGAGTAT----CTTTTCG--- 35
3               --------ATTGA-----TGACGTCAG-----GGGATACGTATTC----GCTCACAC-- 35
8               ------CACGCAGAAGTCTTACGTCAT---------CCA----CCGTGCAGTCC----- 35
9               -----CCAGGCA------TGATGTCAT---------TTC----CCTGGACGTCTCCCAC 35
7               ----ACCTTGGAGA----TGA-GTCAT---------CCTGACTCCCTCAAGCC------ 35
5               ------TAGTGAGA----TGACGTAAT---------TGGGA--CCATTCAGCGTGT--- 35
                             * * ** *
```

Figure 5.2: Alignment of the 10 highest scoring probes for the Jundm2 transcription factor.

tion that the distribution of the position of high scoring features within a probe would be uniform. What we observe though is that the 9bp sequence appears towards the left hand end of the probe sequence. This is the end of the probe furthest from the array.

This could be evidence that the position of a feature within a probe is in fact important to the relative binding efficiency of this DNA binding protein. There are hundreds of protein binding micro-array data-sets available from the UniPROBE database [106]. Similar alignments were made for many of these protein binding micro-array experiments and in many cases a similar effect can be seen. In many cases, any alignment made between the highest scoring probes is less clear (Figure 5.3). This is the same DNA-binding protein but with measurements made on a counterpart array. This is illustrative of the noisiness and lack of reproducibility of protein binding micro-array experiments. It shows that making general statements about this type of data is difficult, especially when replication of experiments is limited to a few or none. A point of interest with the transcription factor Jundm2 is that, as claimed in Badis *et al.* [105], it appears to recognise the motifs `TGACTCA` and `TGACGTCA`. This is referred to as a 'variable spacer length'. Jundm2 is a di-meric leucine zipper type protein (Figure 1.3). It can be speculated that flexibility in the angle between the di-mers allows the possibility of an extra inserted base. This example shows that 'interesting effects' can be seen using this very simple method. We should be honest in assessing what is gained by using more sophisticated machinery.

5.4.2 Lasso results

The lasso algorithm was run thousands of times with hundreds of different parameters. For each run a path of solutions is generated[1]. What is presented here is a selection of representative results.

[1]The path is created as the parameters are fitted during a cyclical coordinate descent.

```
ATGCATAATGACGATCCACGCAGCAAAACGTTACA 19.4652
GGCGGCTAAGGCAAGCCAGGTGACGTCACAGCCCA 18.8871
ATCATTATGCTGTGACGTCATCTAGTAAAATACGG 18.7715
AGGGAGGGAGGGAGGGAGGGAGGGAGGGAGGGAGG 18.7432
GCCGGTTGCTTGAAAGCCTGACGTCACCCGGTCTG 18.6248
TTTGGTTACCCGTTCGATTACGTCATCCTGTGCGT 18.6231
GGAAGCTGGCAATGACGTCACTATTATTCAGCCGT 18.5334
GCGGAGAGACGTAACTTGATGACGTCAGCGCGTTG 18.4948
CAGGAAAGTCTGCTTGTCCGTGATGTCACCTGTAC 18.4686
TGTTCCGCACACATGACGTCATGCGAGTACATGAA 18.4057

CLUSTAL 2.1 multiple sequence alignment


0               ------------ATGCATAATG---ACGAT-CCACGC-AGCAAAACGTTACA- 35
2               ------------AT-CATTATGCT-GTGACGTCATCT-AGTAAAA---TACGG 35
9               -----------TGTTCCGCACAC--ATGACGTCATGCGAGTACA----TGAA- 35
5               --------TTTGGTTACCCGTTCG-ATTACGTCATCC-TGTGCGT-------- 35
8               CAGGAAAGTCTGCTTGTCCGT--G-AT---GTCA-CC-TGTAC---------- 35
4               --GCCG------GTTGCTTGAAAGCCTGACGTCACCC-GGTCTG--------- 35
7               --GCGGAGAGACGTAACTTGA-----TGACGTCAGCG-CGT-TG--------- 35
6               -----------GGAAGCTGGCA---ATGACGTCACTATTATTCAGCCGT---- 35
1               --GGCGGCTAAGGCAAGCCAGG----TGACGTCAC---AGCCCA--------- 35
3               ---------AGGGAGGGAGGGAGGGAGGGAGGGAGGGAGG--------- 35
                                                *
```

Figure 5.3: Alignment of the 10 highest scoring probes for the Jundm2 transcription factor from a counterpart array.

5.4.2.1 Cross-validation prediction compared to counterpart array prediction

In order to select a model from a lasso path that is likely to be the best predictor a cross validation process is used. Ten fold cross validation was performed. The model that gives the best predicted performance on kept back data is used to predict the counterpart array. A comparison of the performance of the predictor on counterpart arrays compared to the cross-validation performance is shown in figures 5.4, 5.5 and 5.6.

The following points are observed,

- In most cases the kept-back data is predicted with a Pearson correlation of above 0.7 and in many cases is above 0.8.
- In some cases the counterpart array is predicted poorly as compared to kept-back data on the training array.
- In some cases the counterpart array is predicted slightly better than the kept back data, though only by a small amount.

It is unclear why some counterpart arrays are predicted poorly compared to kept back data on the training array. It seems plausible that this is due to experimental artefacts, e.g. poorly controlled experiments.

5.4.2.2 Predicting with very sparse models

The lasso algorithm returns a path of solutions. Each of these solutions can be used to obtain predictions of the counterpart array. The solutions are of decreasing sparsity. This means the first few solutions of the path are very sparse, i.e. they consist of only a handful of weighted DNA words.

With a small number of weighted features in the model, the plot of predicted probe intensities against actual probe intensities shows a banding pattern (Figures 5.7, 5.8, 5.9 and 5.10). The banding is explained by multiplicities of the features occurring within probes. This gives clear and quantitative evidence of multiple TF binding events for most probes. For example, in figure 5.7 the two

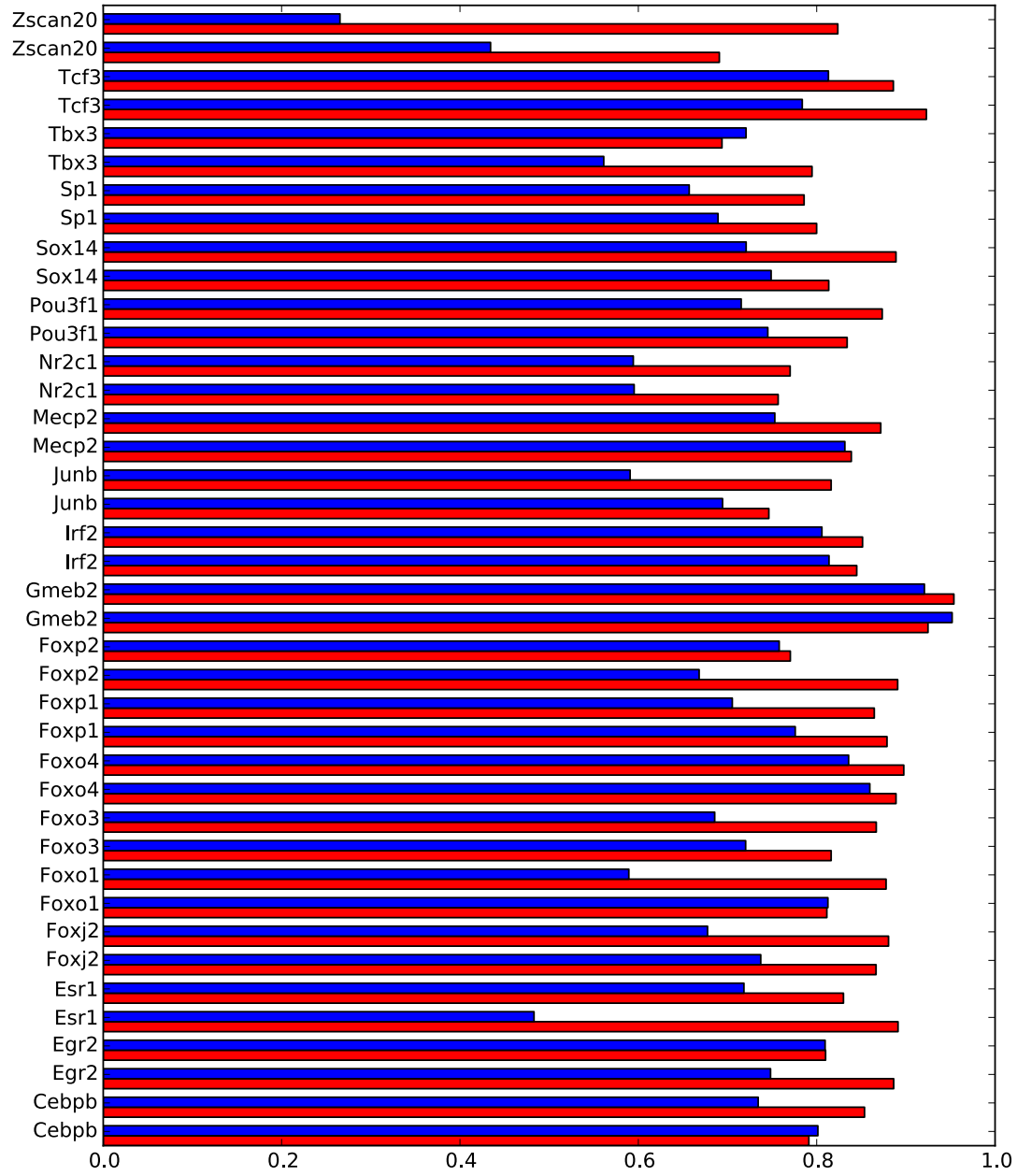


Figure 5.4: Cross-validation prediction versus counterpart array prediction. The blue bars are the counterpart correlations and red bars are the cross-validation prediction correlations. These arrays were the training arrays for the DREAM5 challenge. These arrays alternate in design, HK to ME.

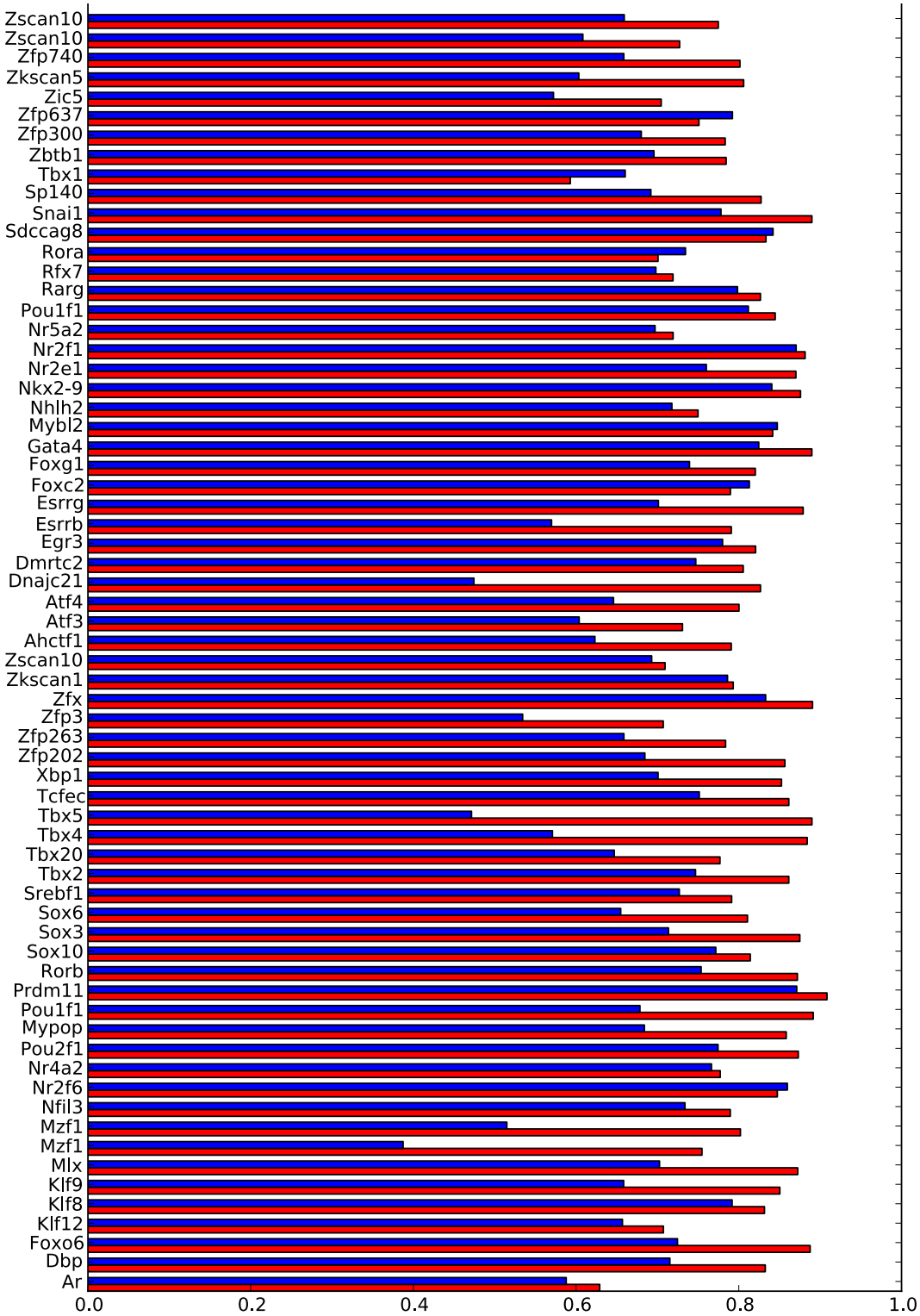


Figure 5.5: Cross-validation prediction versus counterpart prediction. The blue bars are counterpart correlations and red bars are the cross-validation correlations. The top 33 arrays are of the ME design in this picture, the bottom are of the HK design. Training was done on the original challenge arrays.

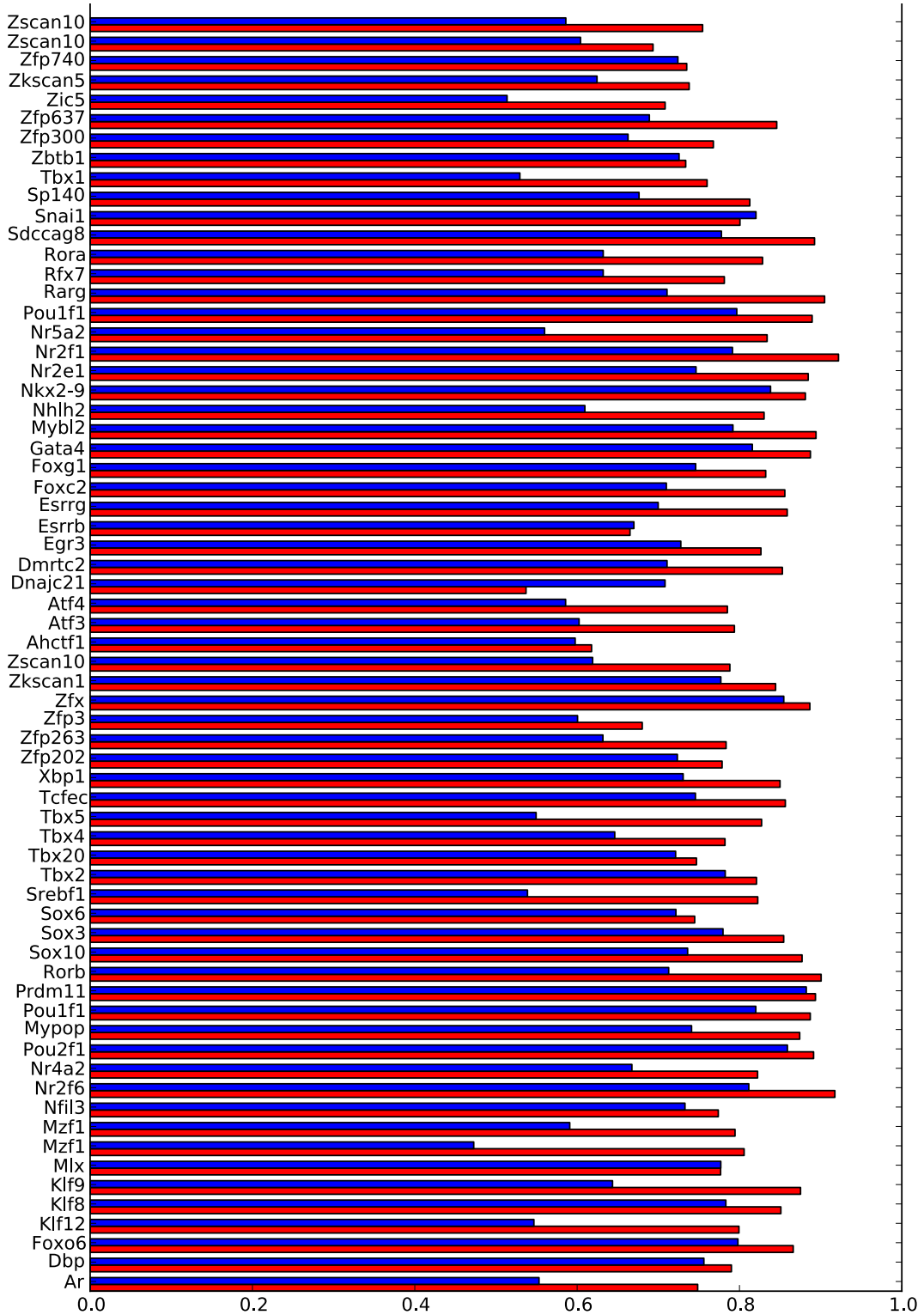


Figure 5.6: Cross-validation prediction versus counterpart array prediction. The blue bars are the counterpart correlations and red bars are the cross-validation prediction correlations. The top 33 arrays are of the HK design in this picture, the bottom are of the ME design. Prediction was done on the original challenge arrays.

TF	2bp	2-9bp
Sdccag8	0.7076	0.8199
Foxp2	0.7065	0.7170
Foxc2	0.6915	0.7739
Dmrtc2	0.6894	0.7078
Zfp637	0.6798	0.7357
Mecp2	0.6719	0.7817
Foxo6	0.6619	0.7538
Foxo3	0.6605	0.6927
Dmrtc2	0.6595	0.7078
Pou1f1	0.6524	0.7851

Table 5.2: The proteins most predictable using only 2bp DNA words. The correlation of a 2-9bp model is shown for comparison. This is the same data as table 5.3 but organised to show the most predictable proteins using a 2bp model.

DNA words, `ACG` and `CGT` can be seen to offer a correlation of 0.7147 between actual and predicted values.

o

In figure 5.8 we see a correlation of 0.7515 provided by a predictor using only 4 DNA words. All of these words are short and there is, similar to figure 5.7, a positive relationship between the number of occurrences of such words and the intensity recorded. Similar effects are observed for the very sparse models of prdm11 in figures 5.10 and 5.9.

This is strong evidence in favour of multiple binding events on each probe and also a surprisingly strong predictive performance for such sparse models.

5.4.2.3 Some arrays can be predicted well with only 2-mers

Table 5.2 shows a selection of lasso models that were restricted to contain only 2bp DNA words. It is interesting to observe that relatively good predictive performance is achieved in these cases. All of the possible 16 2bp words were given positive weights in these models. See table 5.3 for the performance of each model ranked by the improvement made by including more features. Table 5.3 shows that some arrays are very dependent on longer features for predictability.

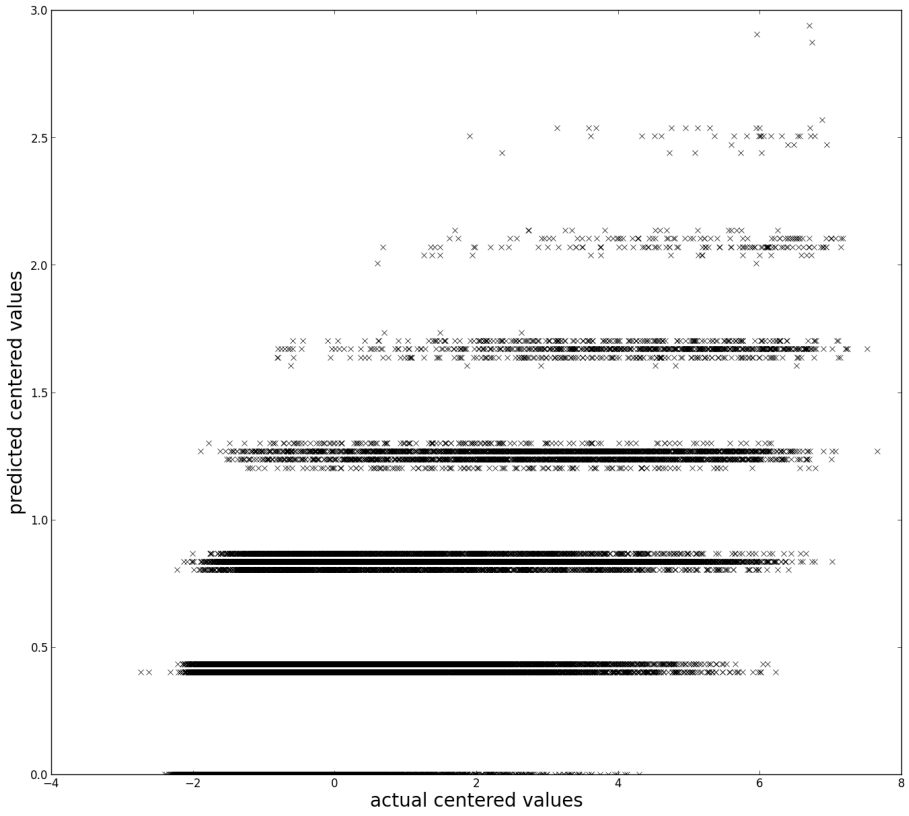


Figure 5.7: Predictions for the Gmeb2 protein using a model built on the HK array and predicting probes on the ME array. This model uses only 2 features and provided a correlation of 0.7147. Looking at the weights of the features in the table below we can see that the lowest pair of horizontal lines of points in the plot correspond to probes that have ACG for the lower line and CGT for the higher. The triple of horizontal lines that appear above are explained by probes that have two copies of ACG for the lowest line, one copy of ACG and one copy of CGT for the line of points in the middle and two copies of CGT for the line above.

Feature	**Weight**
ACG	0.4013
CGT	0.4336

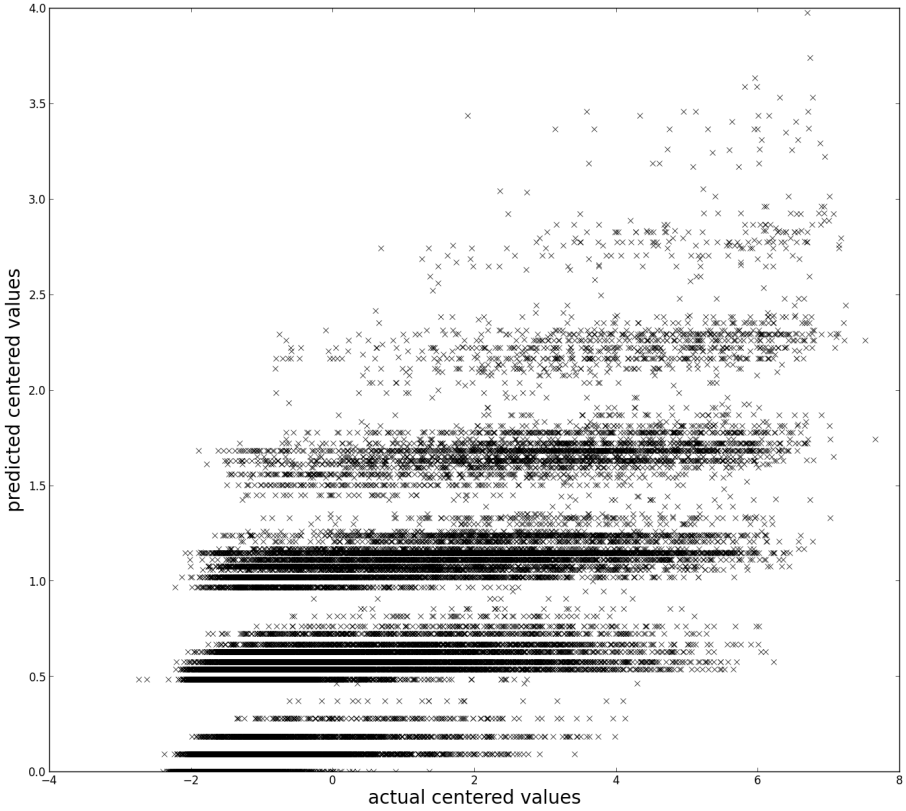


Figure 5.8: Predictions for the Gmeb2 protein using a model built on the HK array and predicting probes on the ME array. This model uses only 4 features and achieved an overall correlation of 0.7515. The banding pattern in this plot has an analagous explanation to that given in figure 5.7

Feature	**Weight**
ACG	0.4827
ACGT	0.1263
ACG	0.0924
CGT	0.5368

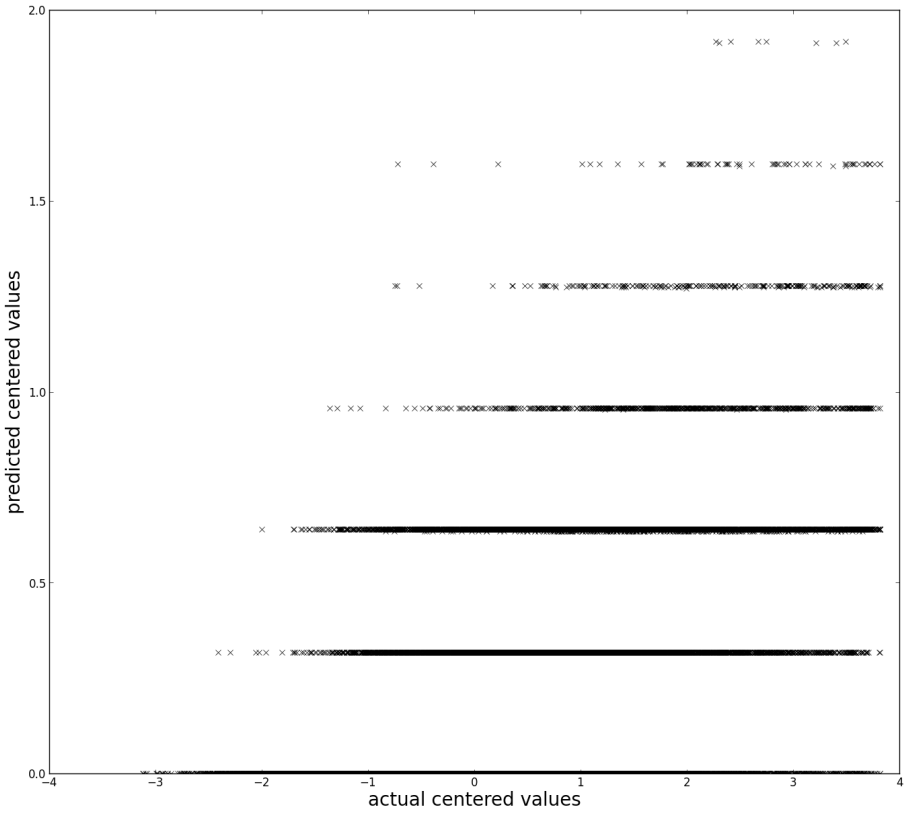


Figure 5.9: Predictions for the Prdm11 protein using a model built on the HK array and predicting probes on the ME array. This model uses only 2 features and achieved and overall correlation of 0.5669. The banding pattern in this plot has an analagous explanation to that given in figure 5.7

Feature	Weight
CGCA	0.3175
TGCG	0.6392

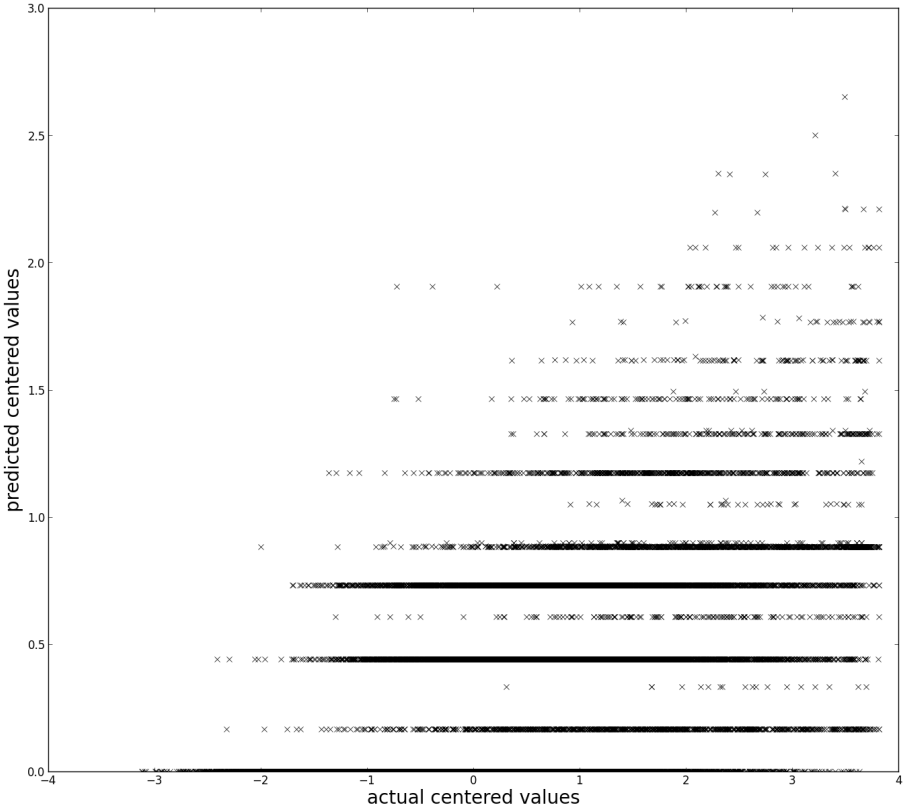


Figure 5.10: Predictions for the Prdm11 protein using a model built on the HK array and predicting probes on the ME array. This model uses only 4 features and achieved an overall correlation of 0.6013. The banding pattern in this plot has an analagous explanation to that given in figure 5.7

Feature	**Weight**
CGCA	0.4427
TAGCG	0.1663
TGCG	0.7320
TTGCG	0.1513

TF	2bp	2bp-9bp	difference	TF	2bp	2bp-9bp	difference
Nr2f6	0.2103	0.7549	0.5446	Sp1	0.5160	0.6529	0.1369
Rorb	0.1835	0.6961	0.5125	Pou3f1	0.5771	0.7072	0.1300
Esrrg	0.1961	0.6880	0.4919	Sox10	0.6121	0.7389	0.1267
Rarg	0.2921	0.7670	0.4748	Srebf1	0.5732	0.6922	0.1189
Nr2f1	0.3177	0.7694	0.4516	Foxj2	0.5838	0.7025	0.1186
Tcf3	0.1969	0.6017	0.4048	Nr5a2	0.5388	0.6559	0.1170
Prdm11	0.4716	0.8676	0.3960	Zfp263	0.4673	0.5839	0.1165
Mybl2	0.4219	0.8169	0.3949	Sdccag8	0.7060	0.8199	0.1138
Nr2e1	0.3606	0.7312	0.3706	Mecp2	0.6688	0.7817	0.1129
Esr1	0.2442	0.5905	0.3463	Tbx4	0.4421	0.5531	0.1110
Snai1	0.4684	0.8005	0.3321	Dbp	0.6016	0.7116	0.1099
Zfx	0.3172	0.6477	0.3304	Sox3	0.6407	0.7485	0.1078
Tbx20	0.3724	0.6934	0.3210	Nfil3	0.5797	0.6807	0.1009
Nkx2-9	0.4983	0.8082	0.3099	Mypop	0.5842	0.6847	0.1004
Rfx7	0.3769	0.6648	0.2878	Zbtb1	0.5743	0.6740	0.0997
Gmeb2	0.6296	0.9116	0.2820	Foxo6	0.6565	0.7538	0.0973
Egr3	0.4667	0.7463	0.2795	Sox14	0.6257	0.7213	0.0955
Egr2	0.4896	0.7690	0.2794	Zscan10	0.5325	0.6279	0.0954
Mlx	0.4777	0.7509	0.2732	Foxc2	0.6794	0.7739	0.0945
Foxo1	0.5044	0.7767	0.2723	Zfp3	0.4800	0.5655	0.0855
Nr2c1	0.2975	0.5674	0.2699	Esrrb	0.5333	0.6185	0.0852
Klf8	0.4307	0.6875	0.2568	Zkscan1	0.6370	0.7207	0.0836
Pou2f1	0.5635	0.8095	0.2459	Atf3	0.5022	0.5828	0.0806
Cebpb	0.5393	0.7794	0.2401	Foxg1	0.6338	0.7118	0.0779
Foxo4	0.5956	0.8337	0.2381	Mzf1	0.5029	0.5693	0.0663
Irf2	0.5733	0.7906	0.2172	Nhlh2	0.5421	0.6084	0.0662
Tbx2	0.4774	0.6891	0.2116	Ar	0.4637	0.5284	0.0647
Rora	0.4686	0.6772	0.2086	Zfp300	0.5837	0.6439	0.0602
Gata4	0.5965	0.7998	0.2032	Dnajc21	0.6063	0.6656	0.0592
Zic5	0.2455	0.4409	0.1954	Zfp637	0.6780	0.7357	0.0576
Zfp740	0.4994	0.6945	0.1951	Sp140	0.6428	0.6907	0.0478
Tcfec	0.5361	0.7214	0.1853	Foxo3	0.6526	0.6927	0.0401
Zfp202	0.5045	0.6847	0.1802	Atf4	0.5689	0.6090	0.0400
Nr4a2	0.5433	0.7140	0.1707	Zkscan5	0.5385	0.5722	0.0337
Tbx3	0.4791	0.6476	0.1685	Klf9	0.5904	0.6219	0.0314
Foxp1	0.5956	0.7543	0.1587	Dmrtc2	0.6877	0.7078	0.0200
Xbp1	0.5600	0.7092	0.1491	Junb	0.5847	0.6033	0.0186
Klf12	0.4386	0.5866	0.1480	Foxp2	0.7032	0.7170	0.0138
Tbx5	0.2972	0.4414	0.1442	Ahctf1	0.5478	0.5585	0.0107
Pou1f1	0.6410	0.7851	0.1441	Zscan20	0.4154	0.4211	0.0057
Sox6	0.5551	0.6938	0.1387	Tbx1	0.6272	0.6178	-0.0093

Table 5.3: Predictability using models with only 2bp DNA words compared to models with all 2bp-9bp words.

5.4.2.4 Almost all protein binding micro-array probes are predictable to some extent

Perhaps one of the most surprising aspects of the analysis of the DREAM5 challenge data is that most PBM probes are distinguishable in their propensity to bind a given protein. A null hypothesis might have been that a subset of probes would bind specifically and the majority would have the same non-specific affinity to the protein, i.e. the latter group would be non-predictable, but this is clearly not the case.

Figures 5.11 and 5.12 show that almost all probes have a systematically variable affinity for the protein of the array experiment, which is predictable.

5.4.2.5 Overall performance of lasso method compares well

The overall performance of the lasso method does well when compared to the results of the other methods (Section 5.2.1).

The lasso has been trained on log intensities throughout. Predictions are made using log intensities. This is the log Pearson's column in the tables below. The 'Pearson's' column is calculated in the obvious way by taking exponentials. The author believes that the log Pearson's statistic and the Spearman's statistic are the most interesting ones. On the exponential scale the correlation coefficient estimator is heavily influenced by a few large data points. On the other hand, an x-y plot is far more interesting than any single number.

The results in tables 5.4 and 5.5 were obtained after applying a B-spline smooth for normalisation, using all 3bp to 8bp words and probe position information in the model.

- The worst predictor performance was probably for the protein Mzf1, it is notable that the lasso made its best predictions using only 9 features in this case. The Spearman's correlation coefficient was particularly low in one case, 0.22.
- The best Spearman's correlation was for Nr2f1 at 0.83.
- The number of weights that have been introduced into the model is several thousand in most cases, many of the weights are very small and an experi-

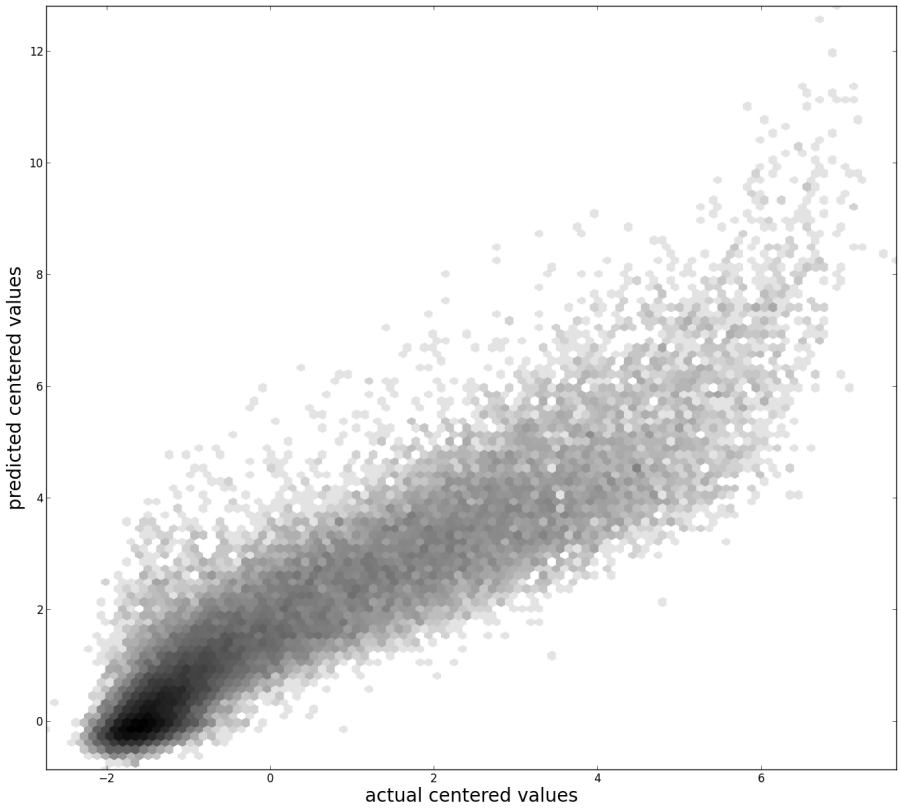


Figure 5.11: Predictions for the Gmeb2 protein using a model built on the HK array and predicting probes on the ME array. The correlation here is 0.9221 and 617 features were selected, see the table below for the features with the largest weights.

Feature	**Weight**
TTACG	0.6401
GCGTAGG	0.6414
CCGTACGG	0.6631
TGCGT	0.7439
AACGT	0.9000
ACGCA	0.9859
TACGT	1.1362
GACGT	1.3101

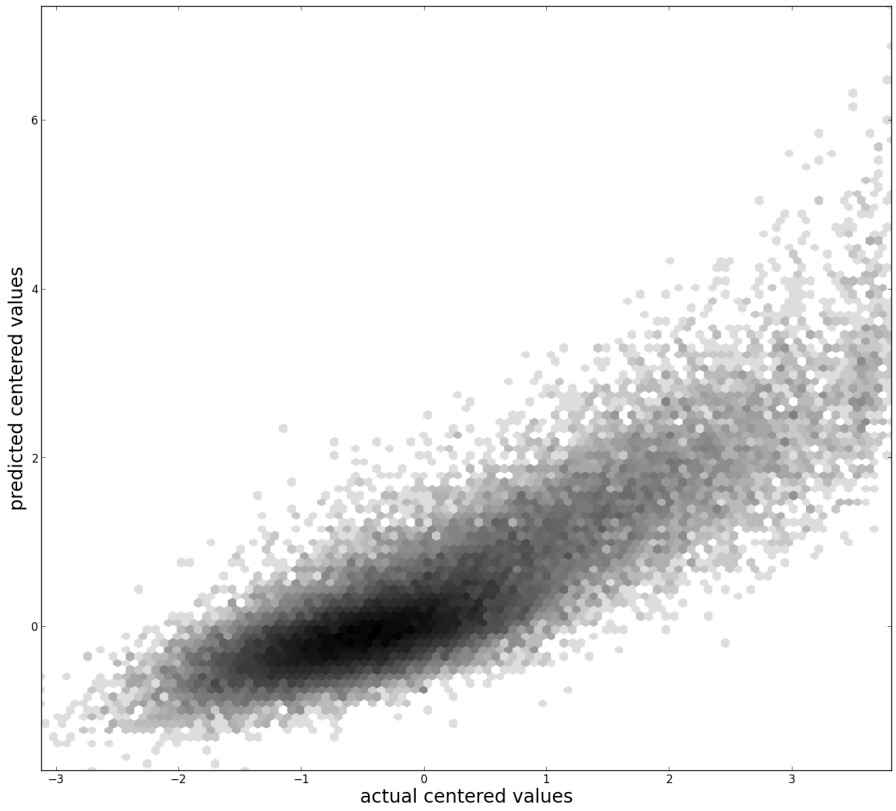


Figure 5.12: Predictions for the Prdm11 protein using a model built on the HK array and predicting probes on the ME array. The correlation here is 0.8504 and 950 features were selected, see the table below for the features with the largest weights.

Feature	**Weight**
ATTGCG	0.5892
CGCTAA	0.6044
CCGCA	0.6875
TTGCG	0.7944
TTTGCG	0.8462
TTAGCG	0.8597

ment where some of these were manually set to zero did not have significant impact on the predictor.

The last point is interpreted as the lasso path algorithm deciding to remove features that had been, at first, introduced. In practice the weights do not return to zero exactly but do get very small. In section 5.4.2.4 it is shown that models with excluded 'small' weights can be more sparse.

Models and predictions have been made for all 172 arrays but for brevity and easy comparison only the subset of 66 from the original DREAM5 challenge are shown here. The aggregate statistics for the 66 arrays are given in table 5.6.

Several arrays benefit from a process of quantile normalisation with their background distributions, as discussed in section 3.3.2. An improvement in aggregate correlation can be achieved in this way. All training and predictions are done on the log-scale and the Pearson's correlation numbers are obtained by simple exponentiation. If we wanted to display improved Pearson's correlation numbers we could scale the data prior to exponentiation to ensure that exponentiated values did not exceed the 2^{16} maximum value. It has been verified that this procedure improves the Pearson's correlation, though it was not used in the production of the results shown here.

5.5 Conclusions

5.5.1 Predictive models have diverse characteristics

The results presented in this section are for a small subset of arrays that are exemplars of particular behaviour, e.g.

- Very good correlation over all probes, e.g. Gmeb2 with a correlation coefficient greater than 0.9 (Figure 5.11).
- Arrays predictable with only a few features (Figures 5.7 and 5.9).
- Arrays predictable with only 2bp words (Table 5.2).

Protein	Pearson's	log Pearson's	Spearman's	# features
Ar	0.5644	0.5895	0.5740	4057
Dbp	0.6868	0.7151	0.7032	5896
Foxo6	0.7435	0.7236	0.6713	6677
Klf12	0.5679	0.6566	0.6672	1714
Klf8	0.7675	0.7922	0.7432	4018
Klf9	0.5110	0.6582	0.6848	4308
Mlx	0.8095	0.7020	0.5964	7406
Mzf1	0.5321	0.3967	0.2247	9
Mzf1	0.5373	0.5154	0.4861	3143
Nfil3	0.7425	0.7328	0.7164	4398
Nr2f6	0.8794	0.8594	0.6472	3592
Nr4a2	0.7781	0.7681	0.7840	3851
Pou2f1	0.7338	0.7743	0.6815	5230
Mypop	0.7052	0.6840	0.6340	7635
Pou1f1	0.6798	0.6777	0.6012	5445
Prdm11	0.8120	0.8708	0.8236	3466
Rorb	0.6945	0.7537	0.6604	6322
Sox10	0.7490	0.7720	0.7398	3265
Sox3	0.6402	0.7133	0.7003	4775
Sox6	0.6305	0.6562	0.5999	4141
Srebf1	0.6842	0.7265	0.7109	3020
Tbx2	0.7791	0.7468	0.6657	8945
Tbx20	0.6998	0.6475	0.5284	6731
Tbx4	0.5435	0.5704	0.5834	6561
Tbx5	0.5375	0.4716	0.4239	4674
Tcfec	0.6849	0.7515	0.7168	4285
Xbp1	0.6691	0.7002	0.6728	5663
Zfp202	0.6916	0.6846	0.6282	3424
Zfp263	0.6504	0.6586	0.6513	4194
Zfp3	0.4730	0.5346	0.5218	4671
Zfx	0.8147	0.8332	0.7786	1490
Zkscan1	0.7489	0.7861	0.7816	3785
Zscan10	0.6888	0.6963	0.6794	4843
Mean	**0.6798**	**0.6916**	**0.6450**	**4595**

Table 5.4: Performance of lasso predictor trained on HK design arrays.

Protein	Pearson's	log Pearson's	Spearman's	# features
Ahctf1	0.6110	0.6228	0.6102	1845
Atf3	0.6093	0.6046	0.5758	3671
Atf4	0.6143	0.6454	0.6378	3534
Dnajc21	0.5007	0.4741	0.5008	9321
Dmrtc2	0.7344	0.7477	0.7453	1993
Egr3	0.7073	0.7803	0.7128	5584
Esrrb	0.5554	0.5697	0.5579	5364
Esrrg	0.6079	0.7007	0.5498	6951
Foxc2	0.7173	0.8127	0.8043	3468
Foxg1	0.7127	0.7402	0.7423	4124
Gata4	0.7537	0.8246	0.7647	6074
Mybl2	0.7458	0.8468	0.7752	6784
Nhlh2	0.6777	0.7184	0.6945	2373
Nkx2-9	0.7646	0.8404	0.8314	9513
Nr2e1	0.7502	0.7578	0.7324	7923
Nr2f1	0.7390	0.8700	0.8285	6468
Nr5a2	0.7640	0.6992	0.6864	4336
Pou1f1	0.7351	0.8119	0.7548	6345
Rarg	0.8283	0.7981	0.7169	6000
Rfx7	0.6519	0.6988	0.6645	5885
Rora	0.7529	0.7362	0.6846	2349
Sdccag8	0.8466	0.8422	0.8141	2334
Snai1	0.7732	0.7777	0.6254	4298
Sp140	0.6746	0.6912	0.6962	2188
Tbx1	0.6310	0.6607	0.6850	559
Zbtb1	0.5997	0.6963	0.6867	5264
Zfp300	0.6482	0.6798	0.6699	3582
Zfp637	0.7582	0.7914	0.7778	1648
Zic5	0.5920	0.5713	0.5125	7035
Zkscan5	0.5622	0.6032	0.5877	4422
Zfp740	0.8348	0.6581	0.5651	2967
Zscan10	0.5608	0.6088	0.5837	3109
Zscan10	0.6202	0.6590	0.6570	4985
Mean	**0.6860**	**0.7134**	**0.6798**	**4615**

Table 5.5: Performance of lasso predictor trained on ME design arrays.

Pearson's	log Pearson's	Spearman's	# features
0.6829	**0.7025**	**0.6624**	**4605**

Table 5.6: Overall performance of the lasso predictor.

There are arrays that have opposing characteristics in each case though. There are some arrays that are not very predictable at all, i.e. the intensity data appears to be noise with no sequence dependence.

Offering a single number, e.g. correlation, to demonstrate the properties of a predictive model is dissatisfying because a correlation plot shows us much more. A set of correlation plots for models of decreasing sparsity is even more informative (Figures 5.7, 5.8, 5.9 and 5.10).

Taking an average of correlations over a set of tens of arrays and reporting that as an output is even worse than doing it for individual arrays. Nevertheless, single numbers are sometimes wanted.

5.5.2 Observations made with the lasso predictor models

Using the lasso method it is possible to predict the intensities of new probes at least as well as with any other method. The lasso method also gives a relatively simple model of probe intensity, i.e. a few hundred to a thousand weights can differentiate between 40000 previously unseen PBM probes with surprising accuracy (Figures 5.11 and 5.12). If we had the null hypothesis that proteins bound specifically to a particular, small subset of 8bp sequences, and non-specifically to all other DNA sequences then these results would evidence against. Some protein binding micro-array probes can be predicted relatively well using only a handful of short DNA words, this appears to depend upon significant additive effects from features on the same probe, sometimes the same feature. This is in contrast to the situation where a probe obtains nearly all of its weight from a single, longer, and more unusual feature.

It is not clear what the actual reasons are for the observed behaviour. A plausible argument for what is happening is the conformation of the DNA is being differentiated between rather than a typical, 'sequence-specific', binding sequence. It is plausible that probes are being parametrised by a small collection of DNA words in some cases because a probe's propensity for *in situ* double stranding is being modelled. If single *vs.* double strandedness determines protein binding then we could imagine that DNA melting temperature is being modelled by a small collection of short DNA words. Another possibility is that parts of

dimeric or tetrameric proteins could be binding in multiplicities to individual probes due to attraction to short DNA words at several positions.

Chapter 6

Genomic Binding Prediction

6.1 Background

In this chapter a comparison is made between predictions of transcription factor binding sites using existing position weight matrix models, (PWM), and the word models developed earlier on. ChIP-seq data from the ENCODE project [107] is used as the 'gold standard' to judge the predictions.

A characterisation of the distribution of scores from PWMs compared to DNA word predictors is shown. The DNA words models are seen to have a better signal to noise ratio than the PWM models when looking at the distribution of scores over the genome. However the prediction of ChIP-seq peaks is not conclusively better or worse.

6.1.1 Scanning the Genome

How we search the genome looking for potential binding sites reflects our model of how a gene regulating, DNA binding, protein performs its function.

We can imagine a DNA binding protein diffusing through the nucleus, 'scanning' the DNA looking for its recognition site. This could be a one dimensional movement along the double helix or three dimensional diffusion throughout the nucleus. The feasibility of the search mechanism was discussed in Berg *et al.* [108] and more recently in Hammar *et al.* [73]. In Bauer *et al.* [109] a complex picture is drawn involving multiple modes of motion and DNA binding affinity.

It seems likely that the protein will, in fact, have a very restricted search space based upon chromatin structure and the steric availability of its preferred binding location. Multi-protein complexes that guide other proteins to their positions, e.g. through bending DNA into loops, could 'place' a DNA binding protein into a precise functional location, (or within a few base pairs), leaving the 'search problem' to be rather different than a 'genome wide scan'. That is to say, perhaps a protein's DNA affinity serves the purpose of keeping it where it is put rather than differentiating between all the places it might end up. Perhaps a protein only has to bind to a region of DNA rather that a precise genomic locus to perform its function. It is possible that a protein could be localised to a region via an affinity to one of its surfaces and it could be 'bound' by a 'sequence specific' binding domain on another surface. This is just speculation but given that, as mentioned above, at least in eukaryotes, the 'blind genome search' seems not tenable, we should not be surprised if the search dynamics are significantly more complicated.

The ChIP-seq data we use in this chapter as the gold standard tells us the location of binding sites that have been determined *in vivo*, reflecting the biological realities described above. In contrast, when we try to predict the binding sites using our simplistic search of the entire genome we do not reflect an actual transcription factor's search process and therefore our expectations are limited. This applies to both the PWM and word models.

6.1.2 Matrix models for prediction

There are many many algorithms that have been developed to find transcription factor binding sites [57, 62], some of which are described in section 1.5. Many methods use position weight matrices as their description of protein binding probability when doing prediction [110].

The position weight matrices used in this chapter were derived from the PPMs available in the JASPAR database [111]. This database has matrices derived from ChIP-seq, SELEX and PBM experiments. Three matrices are used to score individual nucleotides for mouse chromosome 19, each of which was derived from ChIP-seq data. These are described in section 6.2.1. Scans using these existing

PWMs provide a reference computational prediction of binding sites to compare against the word model based predictors developed in this thesis.

The notation for matrix models was described in chapter 2. Similar mathematical notations will be used here. Throughout this chapter the terms 'motif' and 'position weight matrix' will be used interchangeably. This is justified since the position weight matrices discussed here are simply log transformed probabilities,[1] i.e. there is no normalisation for genomic distribution of nucleotides.

6.1.3 Word models for prediction

The word models described in chapter 5 provide descriptions of proteins' affinities to arbitrary DNA sequences of up to a certain length. These models can be used to give a score to every genomic position in the same way as can be done for position weight matrices. Scores are generated for every base pair of mouse chromosome 19. Variations on the application of the word models to prediction were tried including the use of smoothing and the scoring of reverse compliments.

6.1.4 Background Sequence Model

When predicting binding sites with PWMs it is often decided to use a background sequence model that incorporates the distribution of nucleotides in the genome [65]. Models of natural stochastic variation in genetic sequences lead to some simple predictions of the numbers of occurrences of DNA substrings within the genome. e.g. for a 10bp sequence we know that there are 4^{10} possibilities. To a first approximation we might say that each of these will appear in the genome with equal likelihood and therefore in any $4^{10} \approx 1$ million, base pairs we would expect to see each of these sequences once by chance alone. For a more refined approximation we might use a more complex stochastic model of genomic sequence, perhaps taking actual frequencies of nucleotides rather than uniformly distributed frequencies. Is this any better? We know that there are great differences in the distribution of nucleotides as we move along a chromosome and so having a static model for any particular chromosome seems as likely to be

[1] or normalised frequencies

misleading as improving over the unbiased approach of uniform nucleotide probabilities. For example if the greatest binding affinity of a protein model is for a nucleotide sequence that happens to be a likely sample from the genomic frequency distribution, then the number of expected chance occurrences is larger. If the protein binds to regions where the local single nucleotide frequency statistics are 'distant' from those of the genome's mono-nucleotide frequencies we will have added a bias in the opposite direction.

In the absence of any good motivation the principle of least information leads us to use the mono-nucleotide model over a uniform distribution in the rest of this chapter. This applies to both PWM and word models.

6.2 Methods

6.2.1 Available Overlapping Data

The total number of proteins for which the ENCODE project has published mouse ChIP-seq data is, at the time of writing, 55. The total number of word models we have for distinct proteins is 81.

The analysis that follows is restricted to transcription factors that have all three of the following representations available,

1. Matrix model
2. Word model
3. ChIP-seq data

At the time of writing this limits us to the following transcription factors,

- gata4 [112]
- cebpb [113]
- tcf3 [114]

6.2.2 Genomic Sequence Data

We will restrict binding site prediction to a single chromosome. Mouse chromosome 19 was chosen as a typical example of mouse genomic sequence. Whilst results from scanning the entire genome would be more complete we assume this to be a large enough sample of genomic sequence to accurately generate the prediction performance statistics we are interested in.

The mouse chromosome 19 sequence data was as found in the following file obtained from http://www.ensembl.org.

`Mus_musculus.NCBIM37.67.dna.chromosome.19.fa`

6.2.3 ChIP-seq Data

The ChIP-seq peak data was downloaded via the ENCODE project web site, http://www.encodeproject.org [107]. Peaks had been derived from short read data via the MACS algorithm [115], which is based on read depth and reflect regions of high read depth. These peaks are taken to contain the locations of *in vivo* transcription factor binding sites and for this reason they are our gold standard.

Different file formats are provided by different labs; the peak files are in either bigbed, narrowpeak or bed6 format. Different procedures were written to handle each. An introduction to the peaks is shown in the distributions of peak widths in figure 6.1. Later in the methods section the means of these distributions will be referenced when segmenting the genome. It is interesting to note the distribution of gata4 peak widths being quite different to that of cebpb and tcf3. The reason for this is unclear.

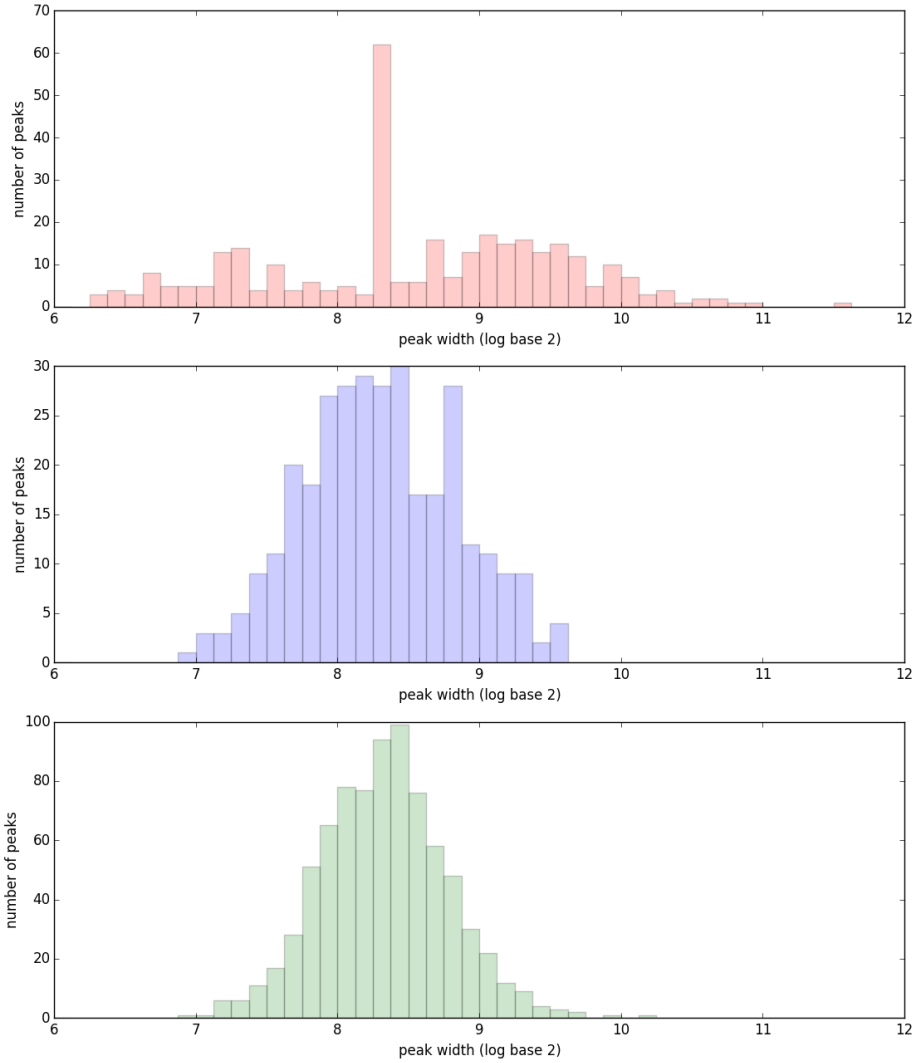


Figure 6.1: This figure shows the distribution of peak widths for gata4 in red, cebpb in blue and tcf3 in green. In the case of gata4 and cebpb the peaks are pooled from the separate data.

The following IDs identify the data used and are from the ENCODE project site. Three datasets are available from experiments on mouse cells and these are described next.

6.2.3.1 gata4

The gata4 experimental data came from the lab of Ross Hardison at Penn State. There are two biological replicates for mouse liver cells:

ENCFF002ADR
ENCFF002ADS

The number of peaks for ENCFF002ADR was 179. The total number of base pairs covered was 125147. The number of peaks for ENCFF002ADS was 152. The total number of base pairs covered was 33819.

6.2.3.2 cebpb

Three different sets of peaks were available for this protein. The data were provided by the lab of Barbera Wold at Caltech. The first two peak sets are biological replicates for mouse myocyte cells:

ENCFF001XUR
ENCFF001XUS

The number of peaks for ENCFF001XUR was 230. The total number of base pairs covered was 68962. The number of peaks for ENCFF001XUS was 220. The total number of base pairs covered was 78428.

A second experiment of myocyte cells differentiated for 60 hours was also available:

ENCFF001XUT

The number of peaks for ENCFF001XUT was 350. The total number of base pairs covered was 122486.

6.2.3.3 tcf3

The tcf3 data was also from the lab of Barbara Wold at Caltech is for mouse C2C12 myocyte cells differentiated for 5 days:

ENCFF001XVM

The number of peaks for ENCFF001XVM was 321. The total number of base pairs covered was 108692.

6.2.4 Matrix Data

The position weight matrices used to score each position of mouse chromosome 19 are given below[1], and were taken from the JASPAR data-set [111]. Each of the matrices were normalised such that each column was a probability distribution, the zero values were made positive and small for computational convenience prior to taking the logarithm of each value.

All of these PWMs have been derived from ChIP-seq experiments.

6.2.4.1 gata4

```
>113MA0482.1 Gata4
10      0       0       0       2707    0       0       547     0       601     386
1039    2165    89      0       39      0       2746    0       1240    1004    929
374     306     0       0       0       0       0       0       940     157     682
1323    275     2657    2746    0       2746    0       2199    566     984     749
```

6.2.4.2 cebpb

```
>97MA0466.1 CEBPB
13006   75198   0       0       4556    0       74715   8654    60151   99494   0
10026   5868    0       0       0       99494   5478    51954   39343   0       36038
33617   18428   0       0       75531   0       10015   0       0       0       2043
42845   0       99494   99494   19407   0       9286    38886   0       0       61413
```

6.2.4.3 tcf3

```
>152MA0522.1 Tcf3
2717    8100    0       17261   0       0       0       0       0       6957    2351
8489    6433    17261   0       4101    16850   0       0       12529   4204    4115
4689    1756    0       0       12403   411     0       17261   3397    96      6625
1366    972     0       0       757     0       17261   0       1335    6004    4170
```

[1]Here, as elsewhere the term position weight matrix is used loosely to denote something that has a canonical transformation into a position weight matrix.

6.2.5 Obtaining scores for a matrix model

Scoring with a matrix model is done in the usual way; each locus i of the genome is given the score that corresponds to lining up the matrix starting at locus i.

As before, let S_i^{i+n-1} be the n bp sequence starting at position i in the genome. Let $s_k \in \{A, C, G, T\}$ be the kth element of S_i^{i+n-1}, let v_k be the score given to locus k in the genome, let m_{ij} be the ijth element of the matrix. Let l be the length of the matrix,

$$v_k = \Sigma_{i=k}^{k+l-1} m_{s_i i} \tag{6.1}$$

where s_i is interpreted as $0, 1, 2, 3$ according to its value A, C, G, T.

Software that scans sequences classifying them as bound or unbound is available, e.g. FIMO was determined to perform well at this task [116]. The simple approach used here is justified since we will vary the classification threshold ourselves in order to generate a RoC analysis.

6.2.6 Obtaining scores for a word model

As described in chapter 4 the word models are lists of features with weights; the models are *trained* by using linear combinations to predict probe sequences with intensity labels.

When predicting the intensity of a probe on a micro-array every feature that appeared in a probe had its weight added to obtain an overall score for that 60bp sequence.

To scan the genome we take each base pair and add together the scores for all features that begin at that position. If we decide that each 60bp sequence of the genome should receive a score in analogy to the protein binding micro-array probes then we can, afterwards, *smooth* the data using a 60bp sliding window. It will be useful to refer to the weight given to a particular sequence, d say, via a function, w say. The weight associated with sequence, (or feature), d, is then $w(d)$.

It should be noted that when scoring protein binding micro-array probes some features would be excluded from contributing to the last n positions of a probe

due to being longer than n bp. For this reason, scoring in the genome *per base pair*, then smoothing with a 60bp window, is not exactly the same as giving the score to each 60bp region of the genome. One could argue that a 60bp probe constraint is encoded into the word model, perhaps via its use of shorter features, and that for this reason we should score each 60bp sequence of the genome as if it were a probe. This argument only really holds weight if we believe that we are predicting affinity to 60bp sequences rather than some subsequence thereof. Whilst it seems likely that measuring affinity to longer nucleotide sequences might be a valid and fruitful endeavour there is no evidence that it makes any systematic difference given the current paucity of data.

Formally, the method that will be used is described as follows. Let $s_k \in \{A, C, G, T\}$ be the nucleotide at position k in the genome. Let S_i^{i+n-1} be the n bp sequence starting at position i and ending at $i + n - 1$. Let v_k be the score given to position k in the genome. Let D_l be the set of all features of length l. For each d of length $l \in D_l$ let $w(d)$ be the weight of d.

$$v_k = \Sigma_l \; \Sigma_{d \in D_l:\; d = S_k^{k+l}} \quad w(d) \tag{6.2}$$

where l ranges over values, in our case, between 2 and 8. The weight function returns values that have a mean of zero, this means that the natural thing to do when a value is not available for a genomic sequence, either because it is an n, or because the sequence or no part thereof appears as a word in the word model, is to assign the value of zero.

6.2.7 Comparison between matrix and word models

In this section a review of the ROC curve method for signal detection analysis is given in the context of this chapter's data. There are some obvious limitations to this approach in the context of our data and these will be discussed. A way to make the ROC analysis more relevant to our data will be presented.

In order to assess the matrix and word models we look for overlap with ChIP-seq peaks. The genomic positions under the peaks will be referred to as condition positives and the genomic positions not under ChIP-seq peaks will be referred to as condition negatives.

	condition positive	condition negative
test positive	true positive	false positive
test negative	false negative	true negative

Table 6.1: ROC definitions

	condition positive	condition negative	
test positive	40	60	100
test negative	10	890	900
	50	950	

Table 6.2: Example ROC calculation

The ChIP-seq peaks that are available are a few hundred base pairs long. Each base pair inside these peaks are thought of as 'bound' and those outside as 'unbound'. Presumably, given the width of the binding protein, these regions are not, in fact, entirely bound.

Our approach is to define each base pair in the genome as a matrix or word model positive or negative.

This is done by choosing a cut-off and labelling the positions with scores above the cut-off as 'test positive' and those below as 'test negative'.

We can then describe a matrix or word model as having good sensitivity if it correctly labels 'bound' positions as matrix or word 'test positives' and good specificity if it labels 'unbound' positions as 'test negatives'.

Taking these meanings of 'test positive', 'test negative', 'condition positive' and 'condition negative' we can, for a given cut-off, produce a contingency table, (sometimes also called a confusion matrix).

For example, let us suppose a genome of 1000bp and ChIP-seq peaks that cover 50bp of the genome. Also suppose that a matrix or word model, at some cut-off, selects 100bp of the genome as positive, (and the rest as negative). We might have the following numbers.
The *false positive rate* is defined to be,

$$\frac{\text{false positives}}{\text{true negative} + \text{false positives}} \tag{6.3}$$

The *true positive rate* is defined to be,

$$\frac{\text{true positives}}{\text{true positive} + \text{false negatives}} \tag{6.4}$$

In this example the *false positive rate* is, 60/950, the *true positive rate* is, 40/50.

6.2.8 An improved ROC calculation procedure

The obvious limitation with the straight forward approach described above is that we are labelling *base pairs* as positive or negative when we are using ChIP-seq peak regions *hundreds of base pairs* long as the 'bound' input. We do not believe that entire peak regions are actually bound and therefore we create spurious false negatives in cases where a peak does have a significant word or matrix 'test positive' label.

An approach to mitigating this is to segment the genome in a way that makes a ChIP-seq peak a single segment and then segment the remaining genome in a consistent way.

The method used is as follows, take each ChIP-seq peak as a single segment and divide the remainder of the genome into segments with the same size as the mean of the distribution of ChIP-seq peak sizes. Each of the latter, 'condition negative', segments will, similarly, be called as test negatives or test positives. The ROC procedure is then performed as before but counting segments rather than base pairs.

There are possible variations on whether to call a segment as a test positive or test negative. We can do this based upon whether there is a single contained base pair with a score above a cut-off or whether the mean of the contained base pairs' scores is above a cut-off.

6.2.9 Smoothing

Smoothing of the scores for matrices or word models is done using a simple moving window. i.e. for a window of length n starting at position i the scores $v_i, v_{i+1}, \ldots, v_{i+n-1}$ are added and divided by n.

6.2.9.1 Motivation for Smoothing

Our aim is to annotate each base pair of the genome with a score that expresses our belief as to whether a protein will bind at that position. An obvious objection to this is that a protein's interacting surface would, typically, be larger than the 'length' of a single base pair in the genome; what does it mean to give a score to a single position in the genome? Perhaps we have in our minds that it would be the 'center', or the 'beginning', of the interaction between protein and DNA.

If a matrix model has been obtained in a manner similar to that described in the second chapter of this thesis then the correspondence between the position of the protein and matrix is somewhat clearer, i.e. if we score a sequence starting at locus i then the score represents the affinity for that sequence binding to a protein 'starting' at that position.

If the matrix has been obtained from a statistical inference on collections of sequences hundreds of base pairs long then this view is less compelling, though not necessarily wrong.

In the case of the word models we learnt a collection of 'features' that predict the binding of proteins to 60bp sequences attached to arrays. On one extreme we may have a set of nearly identical 8bp features in a word model that best predicted protein binding micro-array intensities. In this case we might assume that these features, when found in the genome, predict the position of the protein on the DNA in a way similar to the first of the matrix models described above.

6.3 Results

Figures 6.2, 6.3 and 6.4 show scores for the matrix and word models plotted in the dalliance genome browser [56].

Something that can be immediately seen is the difference in appearance between the profiles of the word model scores and those of the matrices; the word models look like they are giving a better signal to noise ratio (Figures 6.2, 6.3 and 6.4).

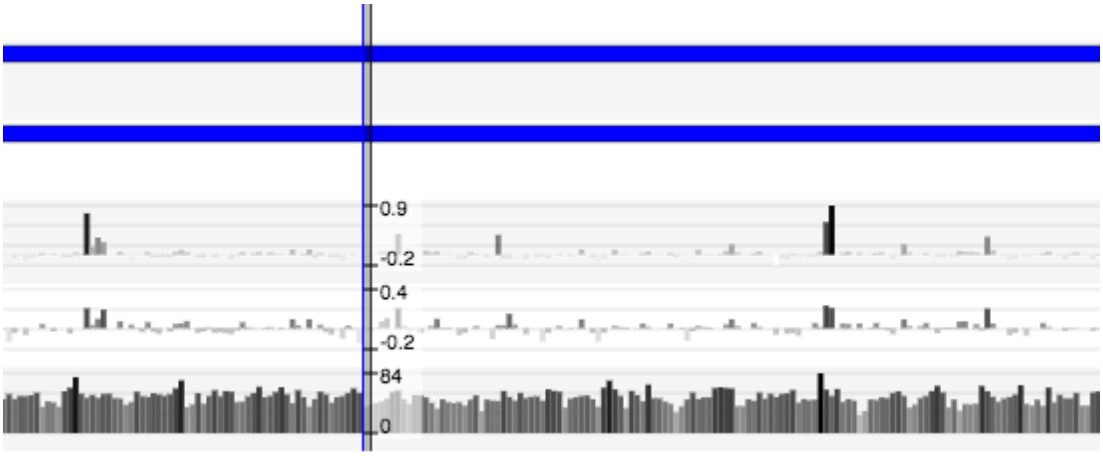


Figure 6.2: Scores for two word models and a matrix model shown together with ChIP-seq peaks for cebpb. The top track with the blue bars show ChIP-seq peaks, the next two tracks are for word models. The bottom track is for the PWM model. Note that PWM scores have been attributed to the first nucleotide position of each sequence being scored.

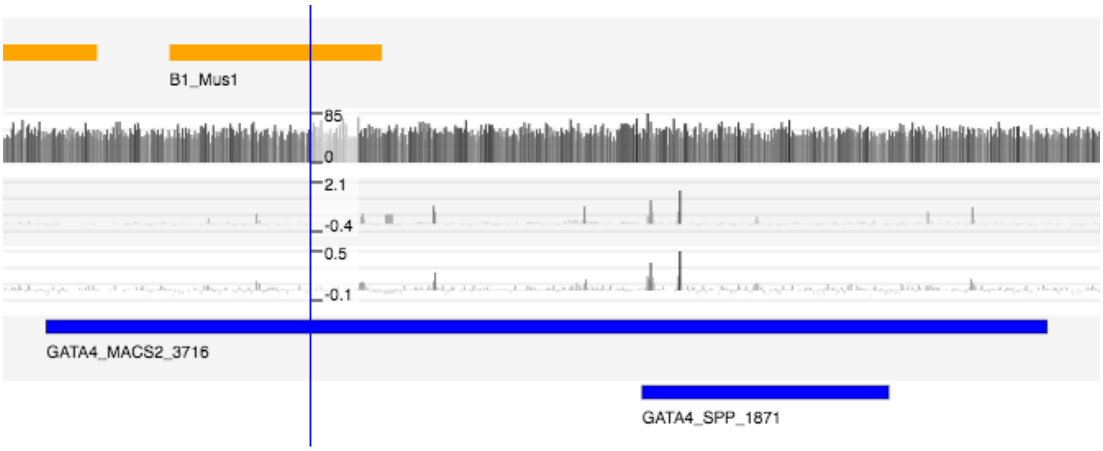


Figure 6.3: The top track in this browser view shows a gene annotation, the next track shows the scores derived from the PWM model and the next two tracks show scores for two word models. The blue bars for the bottom track are ChIP-seq peaks. Note that PWM scores have been attributed to the first nucleotide position of each sequence being scored.

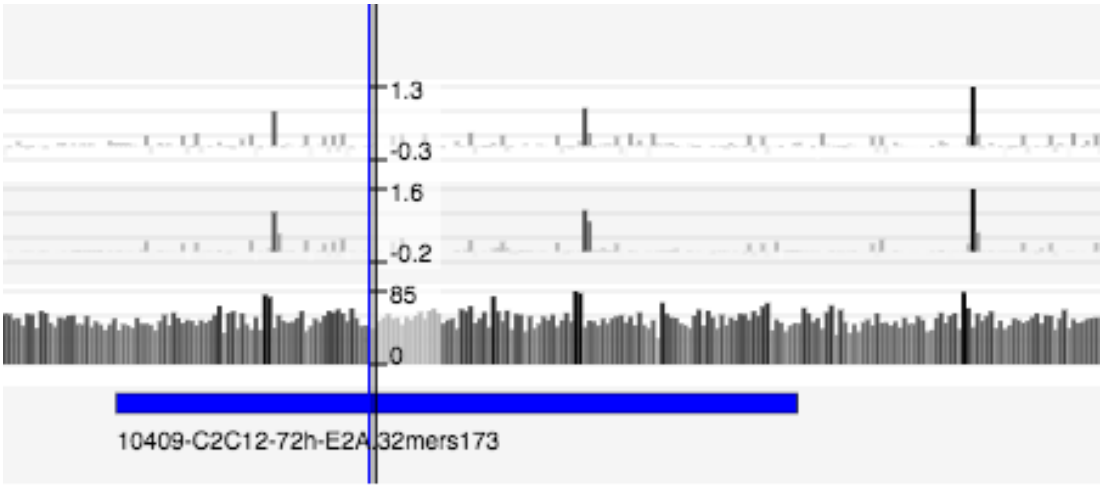


Figure 6.4: Scores for two word models and a matrix model shown together with ChIP-seq peaks for tcf3. The top two tracks are the word models, followed by the PWM model scores with the blue bar of a ChIP-seq peak at the bottom. Note that PWM scores have been attributed to the first nucleotide position of each sequence being scored.

Figures 6.5, 6.6 and 6.7 show the distributions of scores for the different models. The bold lines show the distribution of scores from scanning the genomic sequence and the dotted lines show the distributions of possible scores, i.e. scores from all possible 10mers.

The PWM models produce scores that follow a Gaussian distribution. The Gaussian distribution is characterised by tails that fall off quickly. In contrast, the distributions of scores for the word models have long tails to the right hand side that separate a wide range of the higher scores. This is the reason that we see the better 'signal to noise ratio' in the genome browser (Figures 6.2, 6.3 and 6.4). The long tails of the word models separates the values of high scoring sequences, for a Gaussian distribution this separation is less. The ranks of sequences will be less robust in the Gaussian case, being more susceptible to modelling error and measurement error.

The distributions of sequences in the genome is not a uniform distribution of all possible 10mers but we know this is an approximation. It is interesting to compare the histograms of scores obtained from this actual uniform distribution over 10mers to the actual genomic distribution.

The PWM distribution show a clear over-representation of higher scores in the scanned data. This can be probably be explained by their being trained on ChIP-seq data which has a background of genomic sequence statistics.

The word models do not have a clear over or under-representation of higher scores and this could, similarly, be attributed to their being trained on un-biased sequence data.

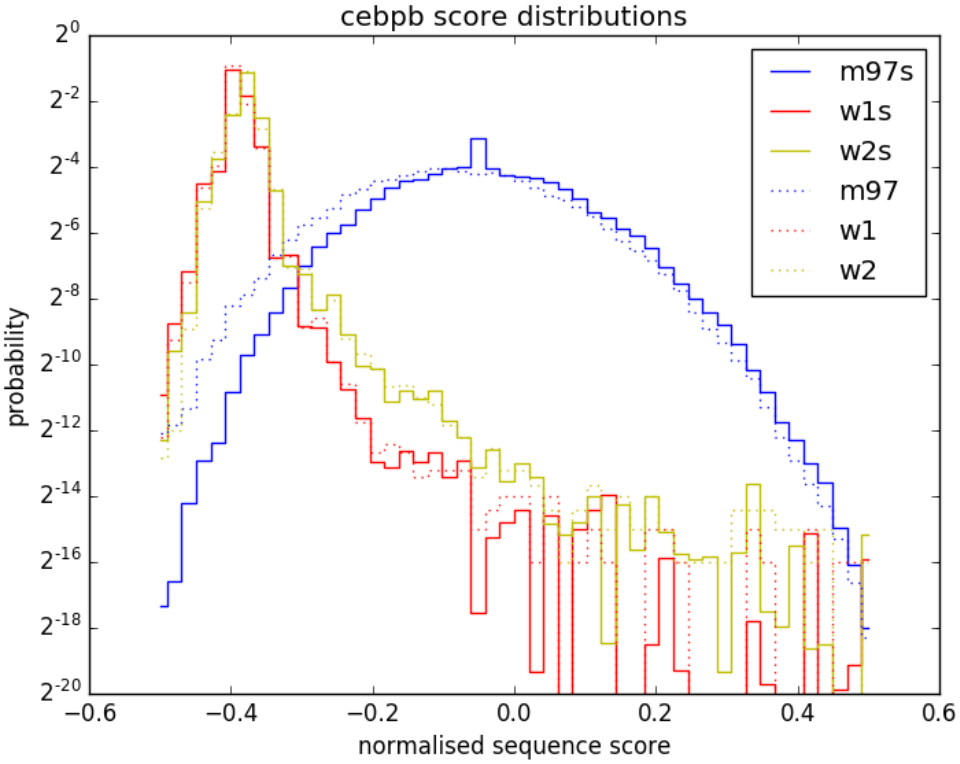


Figure 6.5: The x-axis gives normalised scores. The y-axis is the probability of sequences having a given score.

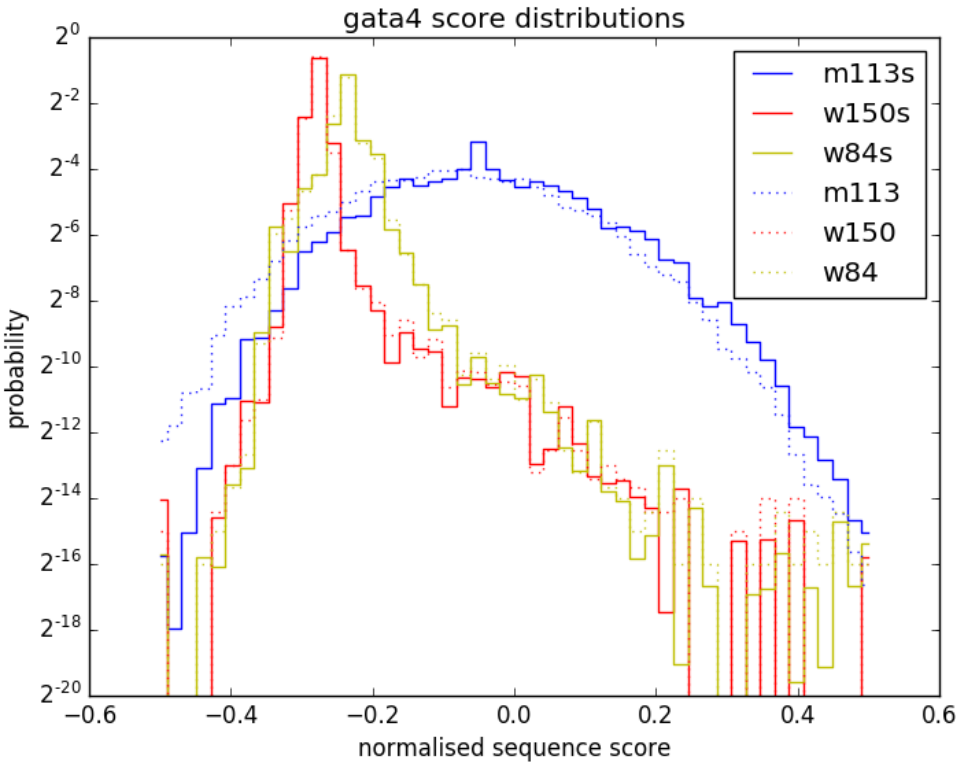


Figure 6.6: The x-axis gives normalised scores. The y-axis is the probability of sequences having a given score.

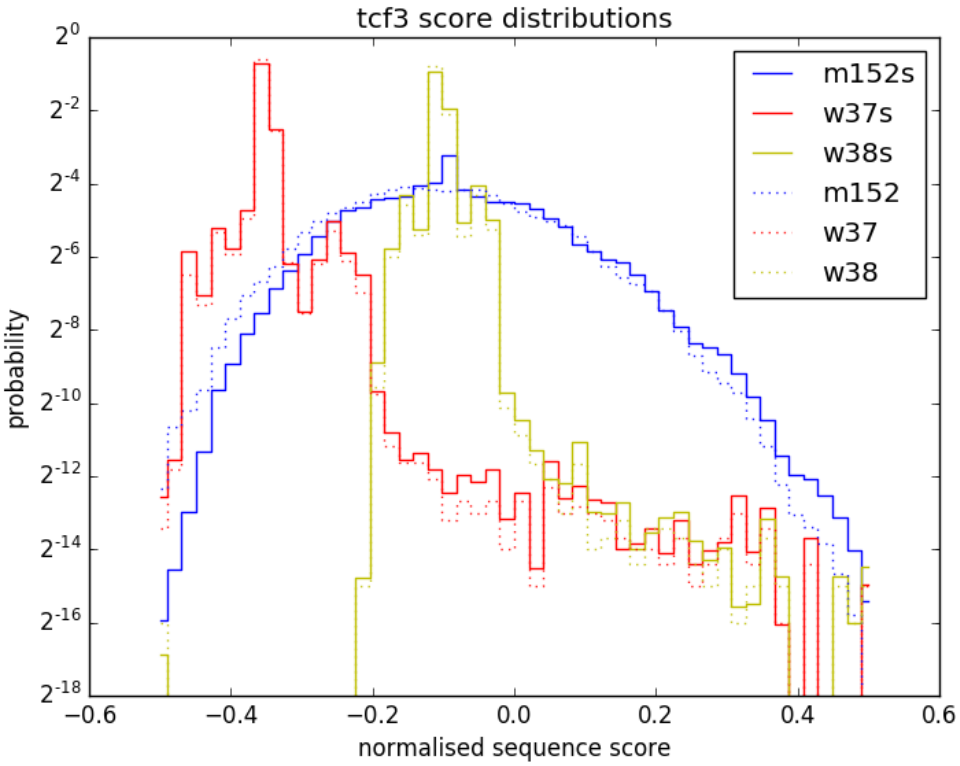


Figure 6.7: The x-axis gives normalised scores. The y-axis is the probability of sequences having a given score.

Figures 6.8, 6.9 and 6.10 show the ROC curves for the segmented chromosome (Section 6.2.8). In all of the figures the word models are the red and yellow lines, (those starting with a 'w' prefix in the legend), the matrix models have blue lines and these are labelled with an 'm' prefix. For the tcf3 transcription factor the matrix model appears to dominate the word models. But for the other transcription factors, cebpb and gata4 it is less clear. With such a small number of overlapping data-sets we should probably refrain from making any conclusive remarks. The word models are appealing for their sparsity and significantly different distribution of possible scores when compared to PWMs, e.g. their ability to create the better signal to noise ratio observed in the genome browser. However, we have been unable to produce a better predictor of ChIP-seq peaks using the word models as defined by the ROC analysis.

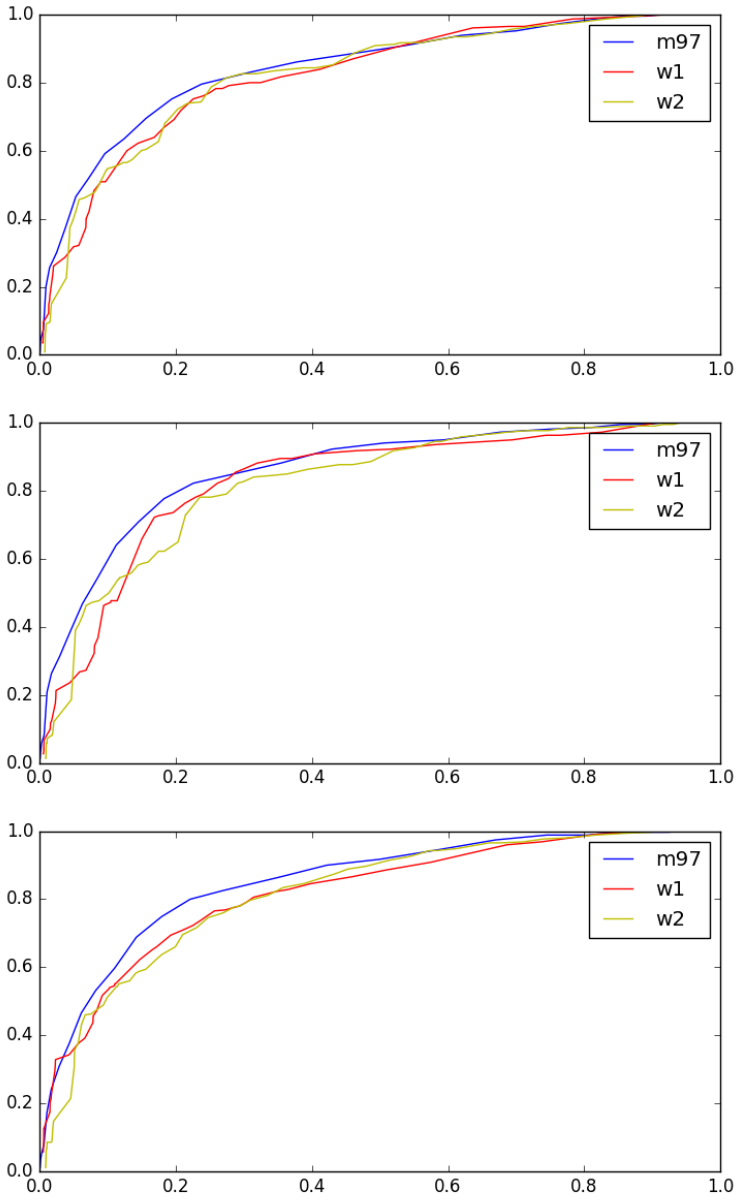


Figure 6.8: This plot shows ROC curves for each of the three sets of peaks available for the cebpb transcription factor. From top to bottom these are ENCFF001XUR, ENCFF001XUS and ENCFF001XUT.

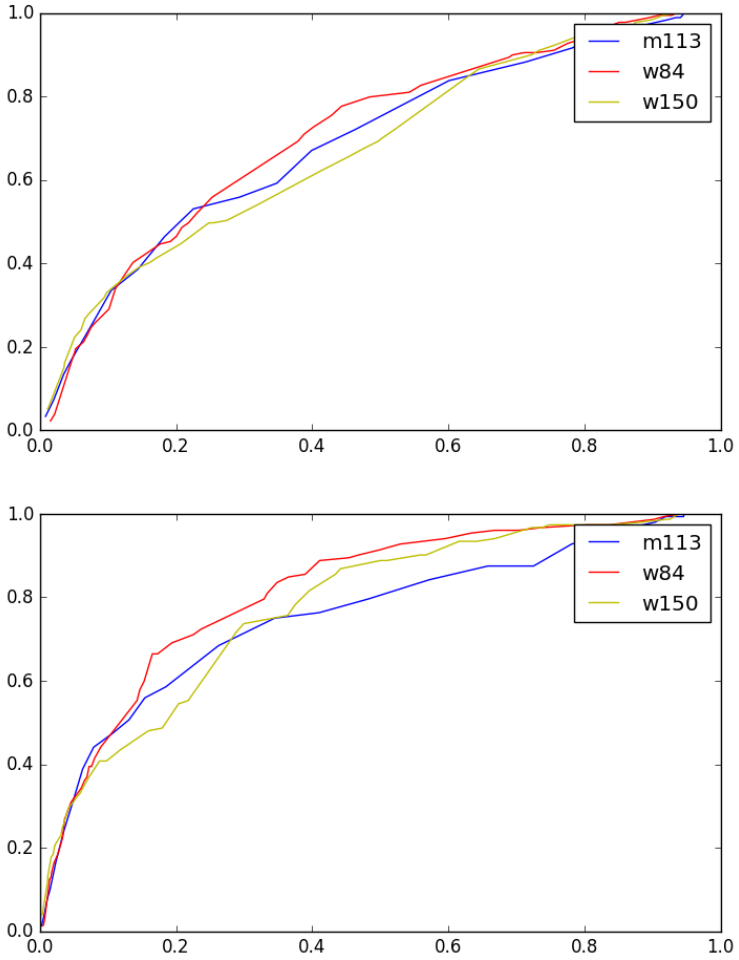


Figure 6.9: This plot shows ROC curves for each of the two sets of peaks available for the gata4 transcription factor. From top to bottom these are ENCFF002ADR, ENCFF002ADS.

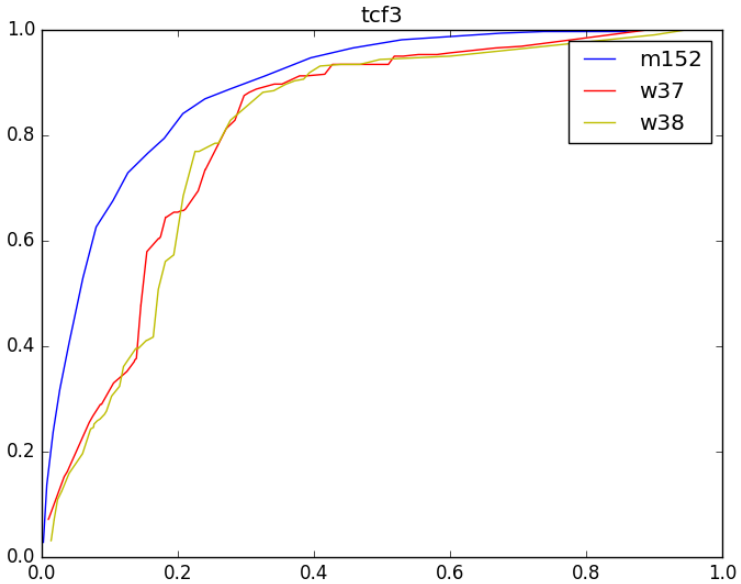


Figure 6.10: This plot shows ROC curves for the peaks set available for the tcf3 transcription factor, ie. ENCFF001XUT.

6.4 Conclusion

Our approach in this chapter has been to scan an entire chromosome looking for the loci that give the highest score for binding affinity according to our models. Our objective was to compare the performance of PWM models as predictors with the word models trained on protein binding micro-arrays. There weren't enough overlapping data-sets (Section 6.2.1) to allow us to make any conclusive remarks on the relative performance of the matrix and word models. However, for the three examples where there was overlapping data, it is impressive that the word models, trained on PBM data, are able to compete with matrix models, trained on ChIP-seq data, i.e. the data source of the gold standard. It is also interesting to observe the substantial difference in the distributions of scores between the matrix and word models.

We have observed word models where a set of 16 weighted 2bp features predict protein bind micro-array intensities very well (Section 5.4.2.3). If we were to score

genomic sequences using such a model we might ask the question whether we are predicting a well defined, static, position for a protein attached to DNA or the probability that a protein will be in a certain 'neighbourhood' of the genome. Unfortunately the lack of data prevents us from doing this experiment, i.e. we do not have ChIP-seq data for the transcription factors in question.

Chapter 7

Conclusions

In this final chapter we review the previous chapters and make some suggestions for further investigation.

The first chapter of this thesis is an introduction to transcription factors, including historical context.

In the second chapter an improved energy matrix model is fitted to some 'physical' measurements of dissociation constants made using a micro-fluidic platform. An extra parameter allows a better fit than that originally offered in the paper that accompanied the data's publication.

In the third chapter, efforts to visualise and correct spatial artefacts observed in protein binding micro-array data are presented. The utility of this work is limited when measured *via* its ability to improve correlations of 'all probe' predictions. Nonetheless the tools that were developed allowed spatial artefacts to be removed in a visually verifiable way. The graphical representation of the data using a 3D point cloud, the surface modelling with B-splines and the visual cutting plane method for data-exclusion offered a clear improvement in handling this noisy data-set.

In the fourth chapter are descriptions and comparisons of the make up of probe sequences on the protein binding micro arrays and the presentation of the model with which transcription factor binding to protein binding micro-array probes is based. A careful description of the mathematics behind the sequences is given since it is important to understand the statistical and compositional properties of the sequences if we are to sample from them. The model and computational

approach described is efficient and can be scaled to 'larger' problems than thought possible by previous experimenters.

In the fifth chapter is the presentation of predictors built using the protein binding micro-array data and the probe models described in the previous chapter. Nested models of decreasing sparsity show interesting features. The protein binding micro-array platform offers many, unbiased measurements of the same protein's interactions but further data could improve its scope in this author's opinion.

In the sixth chapter the predictors learnt in the fifth chapter are applied to scanning genomic sequences. Genomic sequences are scored using the predictors and a comparison is made with ChIP-seq peaks, as a gold standard. An appropriate ROC calculation is devised and the performance of the predictors is compared to PWMs via ROC curves.

In this thesis, we set out to investigate whether it is possible to develop an improved method to predict genomic transcription factor binding sites using a new data type, i.e. protein binding micro-arrays, and a new machine learning method, which produces sparse models over a set of DNA words. While we were able to present novel and interesting models, our predictions against the gold standard available were not significantly different.

Further Directions

'Tis much better to do a little with certainty, and leave the rest for others that come after you, than to explain all things by conjecture without making sure of any thing.' Newton, 1704

Prediction of transcription factor binding sites is an important problem however the ROC plots (Figures 6.8, 6.9 and 6.10) show the limited accuracy of existing methods. We now consider future work that might improve prediction methods.

The quality of the data with which we train models can ultimately limit what can be achieved by improving the models themselves. Given small data-sets we can use constrained models, such as PWMs, but are then faced with the associated

biases. Adding extra parameters can remove biases but high-dimensional models require vast amounts of data to be fitted.

The protein binding micro-arrays offer large amounts of data, though, at present this data appears to be noisy and possibly biased to the *in vitro* conditions where it is measured. One opportunity is to improve the quantity and fidelity of the protein binding micro-arrays. Another opportunity is to incorporate information from other data-sets, such as epigenetics.

Data to allow modelling of epigenetic state

We would like data to describe the entire epigenetic state of the cell. This would include chromatin modifications in addition to methylation information. This information could give us a proxy to the steric availibility of binding sites and the effect of other, cooperative, regulatory mechanisms (Section 6.1.1).

To fully assess the impact of epigenetic modification on transcription factor binding requires ChIP-seq experiments and entire genome DNase and methylation data-sets for the same transcription factors and cell type [117].

There are no complete epigenetic datasets for any of the three transcription factors with PBM datasets discussed in the previous chapter, however a single DNase I hypersensitivity data-set was available for the gata4 protein which was briefly investigated.

The processed ENCODE DNase I dataset ENCFF001YRK contained 7,010 hypersensitive regions on mouse chromosome 19, containing 3,061,224 bases and corresponding to 4.9% of the chromosome. DNase I hypersensitive regions of a chromosome identify DNA that is more accessible to binding proteins and as expected most of the ChIP-seq peaks identifying gata4 binding sites were contained within these regions (78%). To test whether the PWB or PBM based prediction methods behaved differently in these regions, the intersection of ChIP-seq peaks with hypersensitive regions provided a new gold standard with which the ROC procedure was repeated, see the dotted lines in figure **??**. We can see that the ROC curves are not obviously improved. The low false positive rate region is highlighted in figure **??**. It might be tempting to suggest that the DNase data has improved our predictions in the latter case, but more data is required to have any confidence.

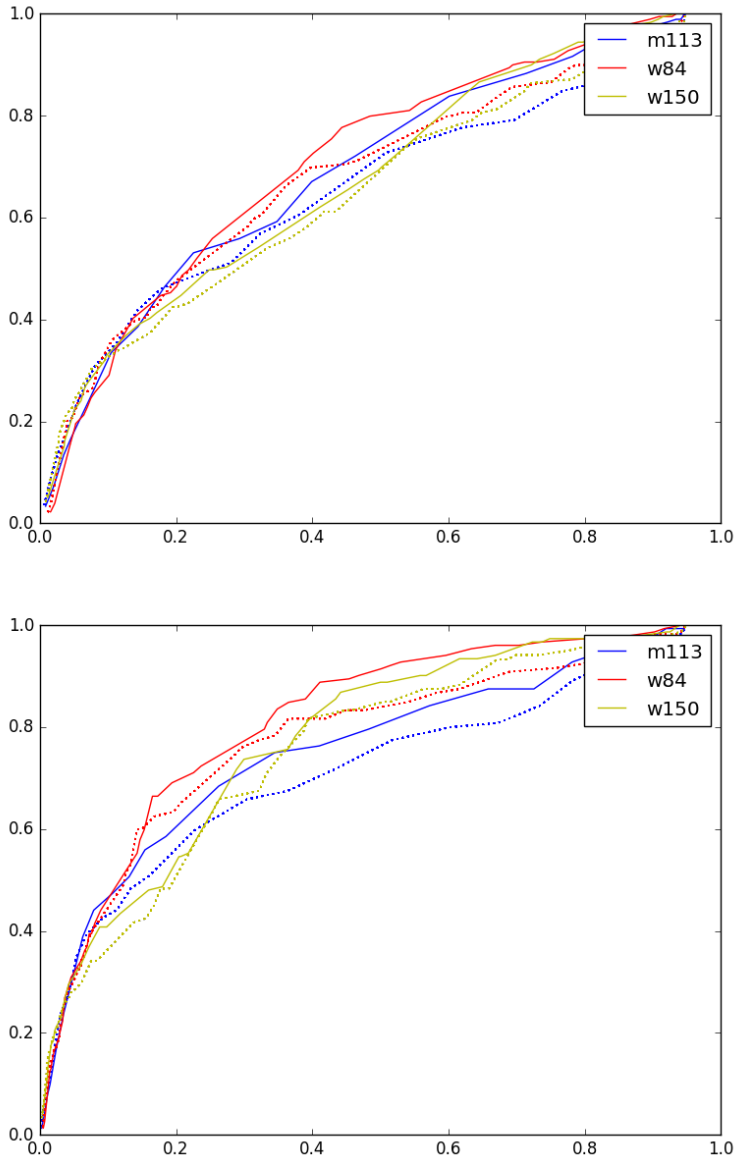


Figure 7.1: Dotted lines have been added to show the ROC curves using a gold standard created from the intersection of the ChIP-seq peaks, described in the previous chapter, and a DNase I hypersensitivity data-set ENCSR000CNJ (ENCODE). The top curve is for the ENCFF002ADR peaks, and the bottom curve is for the ENCFF002ADS peaks.

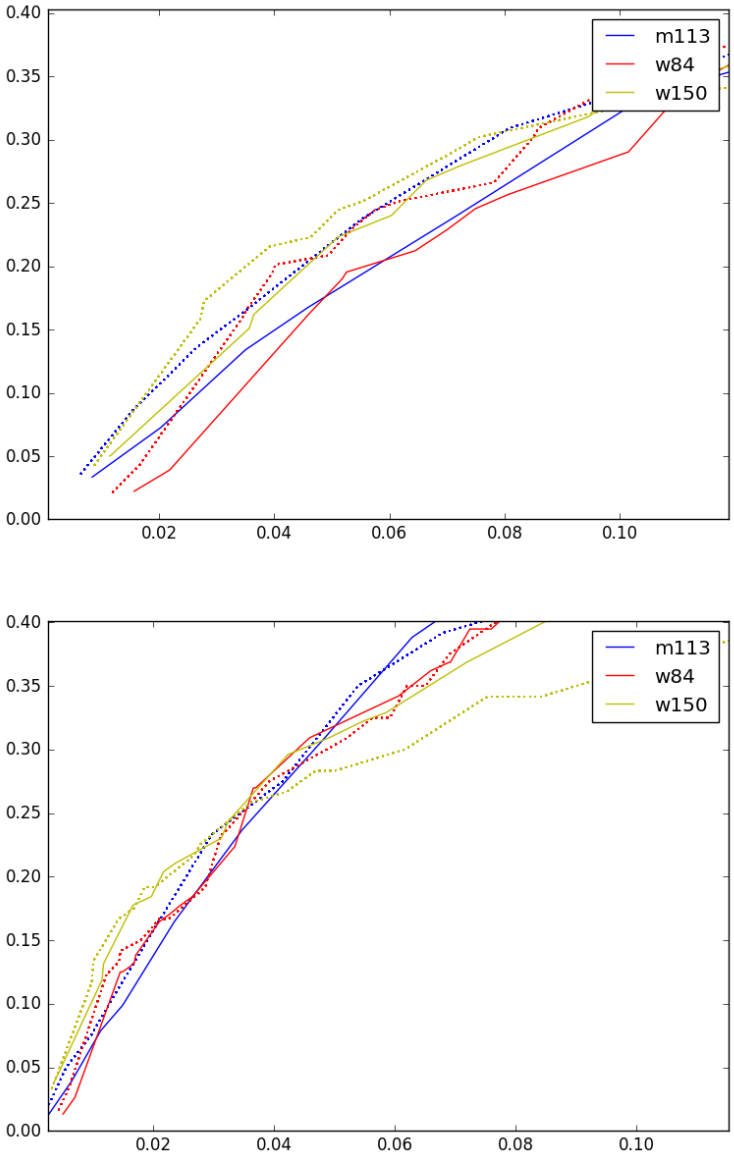


Figure 7.2: The same to the previous figure but showing only the part of the curve related to false positive rates below 0.1.

One avenue of future work would be to collect complete data-sets for cell types where transcription factors of interest are active. This should help us understand whether the addition of epigenetic data can be used to improve predictive models.

However based on the single experiment described above, it seems unlikely that any improvement will be large. This suggests that taking steric availability into account, by incorporating epigenetic data, is unlikely to explain most of the poor performance of transcript factor binding site prediction algorithms. A better strategy may, therefore, be to improve transcription factor binding models.

Protein binding micro-array improvements

The protein binding micro-arrays are in many cases noisy and different experiments have diverging characteristics. Reporting average correlation statistics for a set of 172 of these arrays, over 82 different proteins, is not necessarily a useful exercise. It seems that the experiments are potentially very informative, providing a rich characterisation of the properties of both the array bound oligonucleotides and the protein constructs measured with them. Though the lack of repetition of experiments, the precision of measurements, and availability of all possible data is frustrating. Improving on these issues seems like the obvious way to improve investigations in this area.

One protein[1], Zscan10, was measured on 6 arrays. Data for three experiments on the HK design array and three experiments on the ME array are available. See figure 7.4 for the intensity distributions and figure 7.3 through for correlation plots for the probes on each pair of Zscan10 arrays of the same design. These plots suggest a lack of reproducibility that would certainly guide our attempts at prediction.

After finding a probe's intensity an obvious next question is, what is the measurement's error distribution? Inspecting figure 7.4 we see that the distributions appear to have different locations and scales. It would be interesting to try and parameterise these distributions with location and scale parameters. We could then normalise them and estimate error distributions for the intensity measure-

[1]Another protein, Mzf1, was measured on 4 arrays, twice on an HK array and twice on an ME array. All other proteins were measured only once on an HK array and once on an ME array.

ments. Relevant questions are,

- what is this distribution of signal intensities?
- how does it arise?
- why do these distributions differ between replicate experiments?
- why do these distributions differ between proteins?

To answer these questions more data is required.

Considering figure 7.3 shows us that, even with a perfect model of probe sequence content, we may still find prediction difficult.

In the absence of economic constraints we might like to see significant amounts of data on the properties of dsDNA micro-arrays alone. This data should be produced in the absence of confounding factors. For example, we might want to see 100s of protein binding micro-arrays produced and measured for cy3 incorporation. e.g. 10 arrays done on 10 different days in several different labs. One could then repeat this under various regimes of temperature, salt concentrations and equilibrating times.

Once a 'noise model' of the arrays was well understood protein binding experiments could be done yielding a potentially much greater amount of useful information.

Several replicates per protein allow us to use the central limit theorem effectively, i.e. the variance of the estimator of the mean tends towards a normal distribution with variance $\frac{\sigma}{\sqrt{n}}$ where n is the number of replicates and σ is the standard error of the data. This convergence depends upon the distribution of the underlying data. We do know that the data are non-Gaussian but cannot make any general statements about these distributions.

With replicate experiments an understanding of error distributions could be obtained, then we could determine how many replicates are needed for a particular purpose.

For the same protein, parameters such as

- protein representation, (domain or whole protein)
- alternative protein orthologues/paralogues

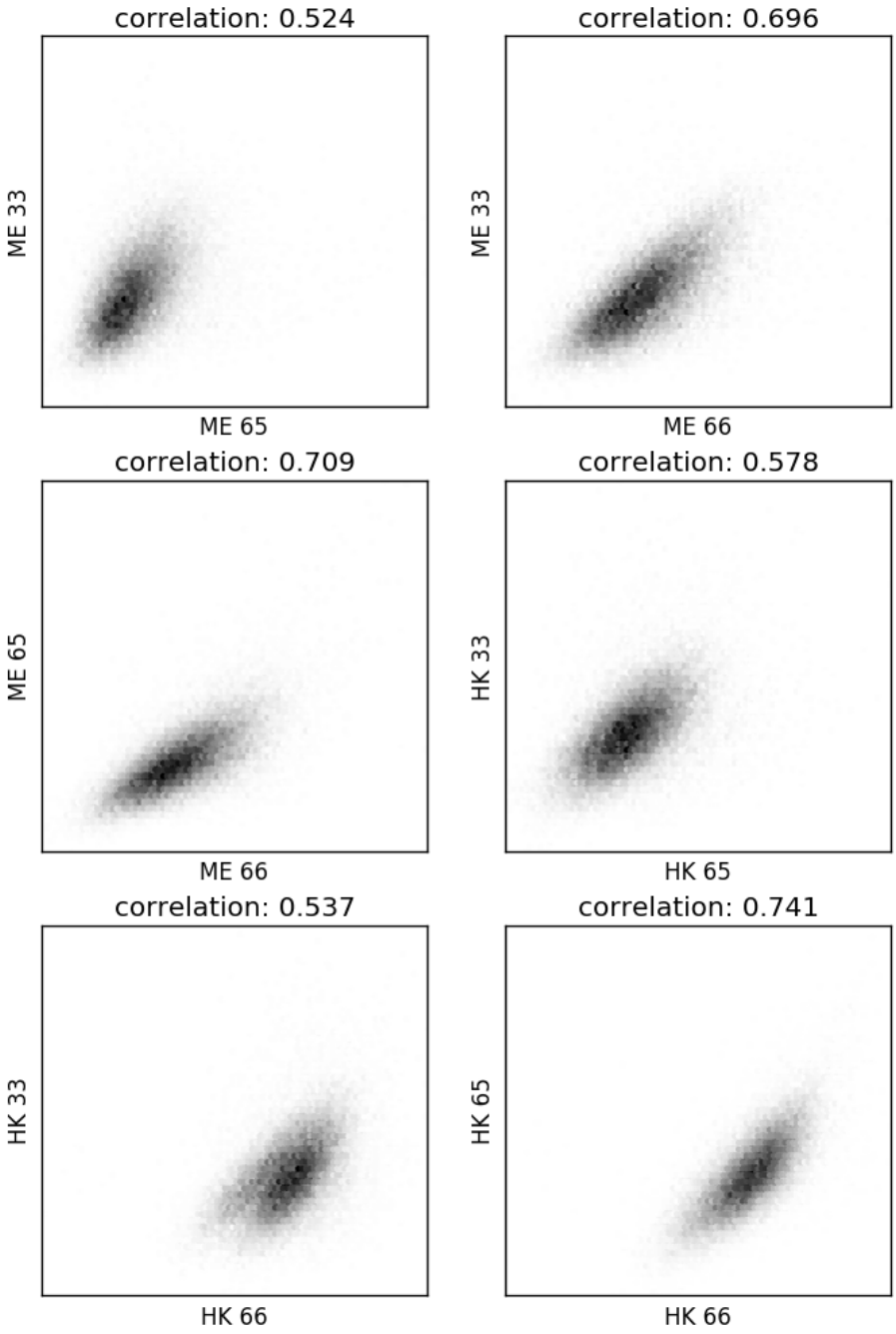


Figure 7.3: Correlations between probes for each array type and for each Zscan10 array

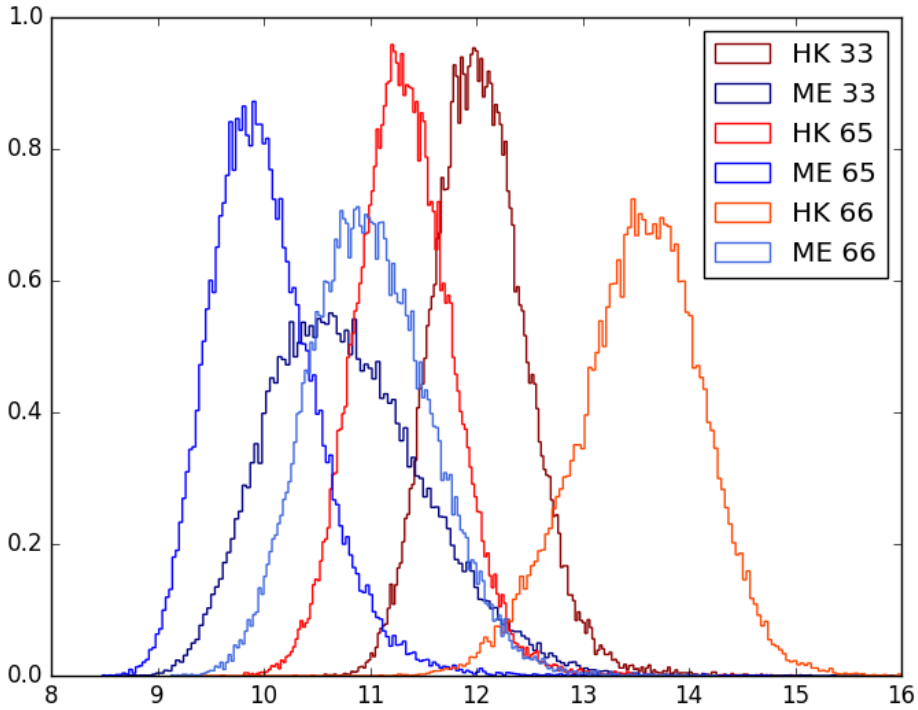


Figure 7.4: Intensity distributions for the protein Zscan10. These 6 histograms show the signal intensities for the protein Zscan10 as measured on 6 arrays of each design.

- a range of protein concentrations
- a range of salt concentrations
- a range of temperatures
- experiments with and without cofactors
- experiments using arrays generated different de Bruijn sequences, (this mitigates a number of experimental biases)
- several replicates of each experiment to compensate for the significant noise for each set of parameters

could all be varied in order to obtain greater depth of understanding.

An example of the utility of having replicate experiments with different protein concentrations is described in chapter 2, namely the potential to derive a disassociation constant for non-specific binding.

To begin to capture effects of cooperative binding, data-sets could be generated for all of the above conditions measuring multiple proteins in tandem. This would help answer questions about the validity of predicting the binding of individual factors. Obviously the number of experiments required is very large, and a high degree of automation would likely be needed in order to face such a challenge.

Final words

As a general summary, finding transcription factor binding sites in genomic sequences remains a difficult problem, nevertheless it is also an important one. It should be possible to find these sites computationally using sequence information but more work is required. PWMs are much used and can be useful but they have very poor specificity. These models are built from incomplete binding affinity data, e.g. chemical activities are not considered. Proteins *in vivo* will bind based upon conditions that we may not have measured and that perhaps go beyond what is currently, reliably, measurable, e.g. all epigenetic states.

Appendix A

As part of chapter 3 a surface fitting method was described. The B-spline algorithms, parameter fitting and drawing of resulting surfaces were implemented as a tool from basic, high performance components.

- A custom C++ B-spline implementation
- Numerical optimisation with the Eigen linear algebra library
- Native OpenGL for rendering of surfaces
- The Lightweight, ODB, object relational mapping layer over database
- Sqlite3 database for data storage and organisation

Some screen grabs of the GUI that was made are shown in this appendix.

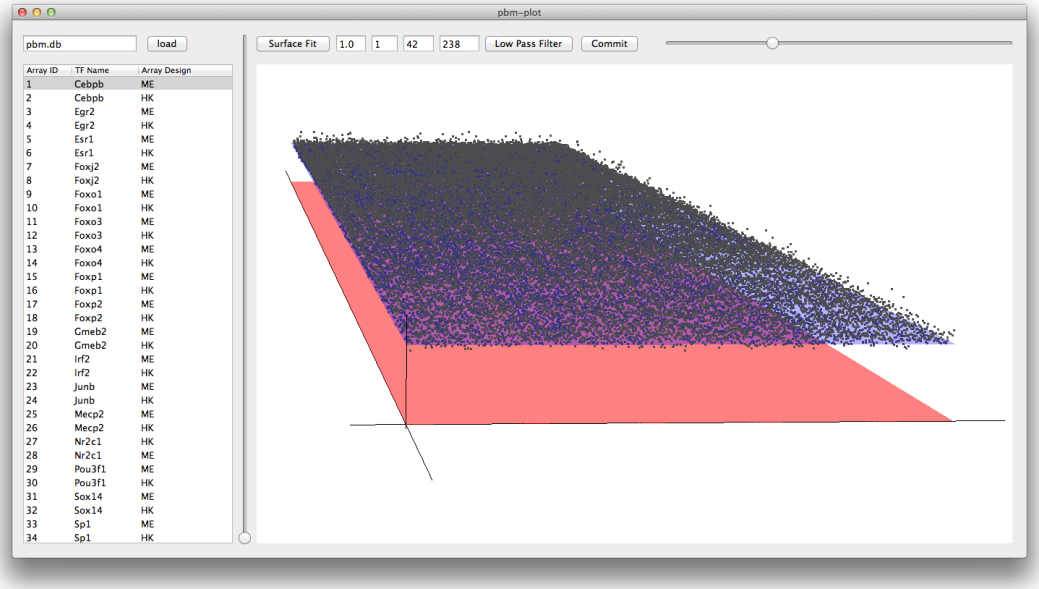


Figure A.1: On the left hand side is a list of experiments that can be ordered by index, name or array type. The table view is populated from an Sqlite3 database that is specified by its filename. Grid references and intensity values for every probe are pulled from the database and drawn in less than 1 second. This allows easy and rapid comparisons between experiments.

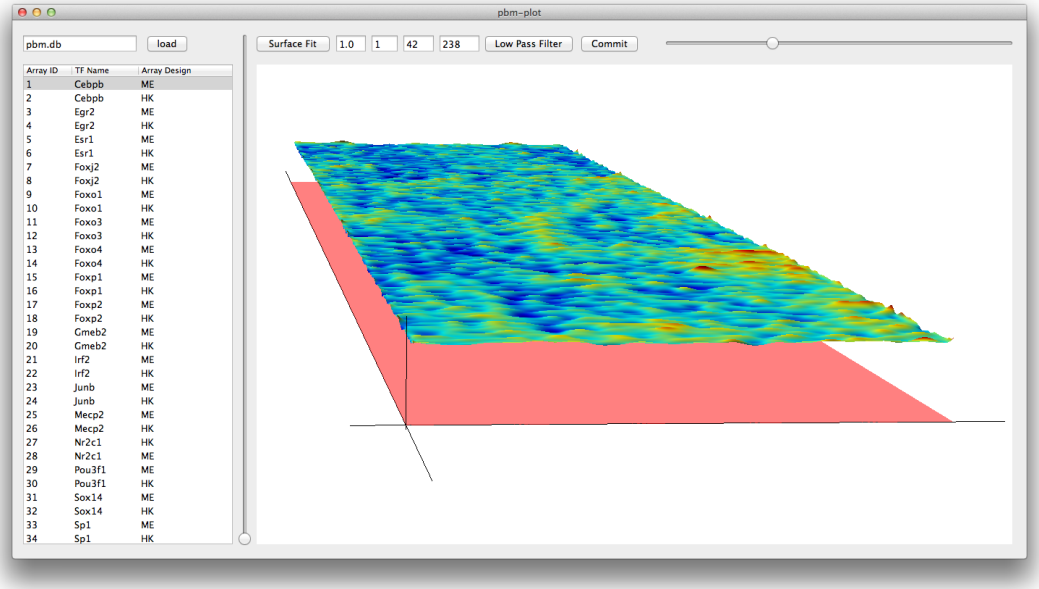


Figure A.2: Using the 'Surface Fit' a B-spline surface is drawn using the four parameters to the right of the button. The first is the smoothing penalty and the second and third are the number of equally spaced knots in the x and y directions. After fitting the surface the difference between the surface and the global mean can be subtracted from the data, these normalised values can be added to a column in the 'Intensity' table of the database by clicking the 'Commit' button.

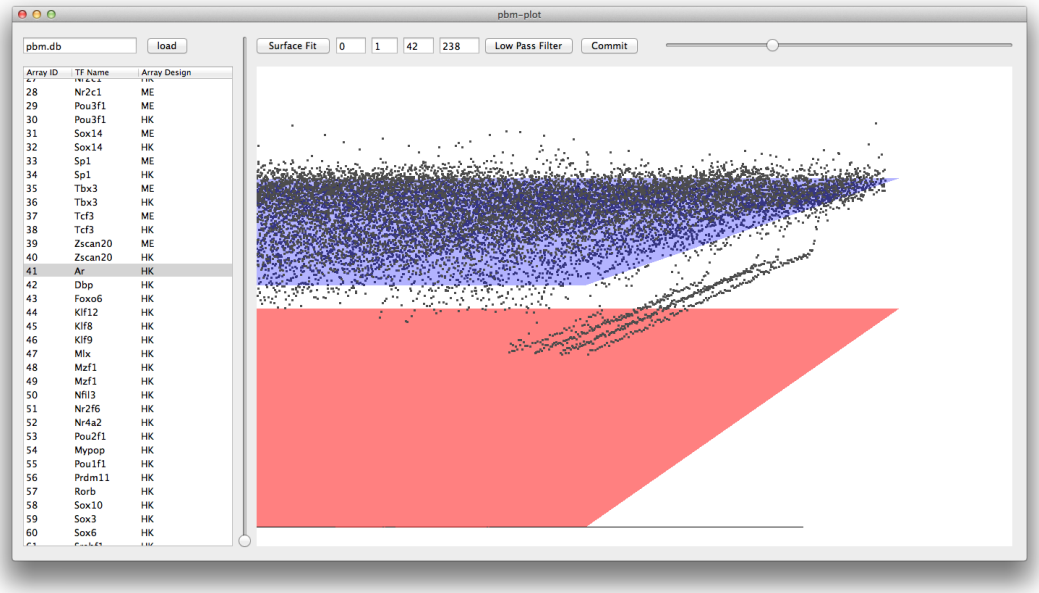


Figure A.3: In this screen grab we can see some outlying data that we wish to exclude. The ability to move around the data in 3D allows us to see the spatial artefact in detail.

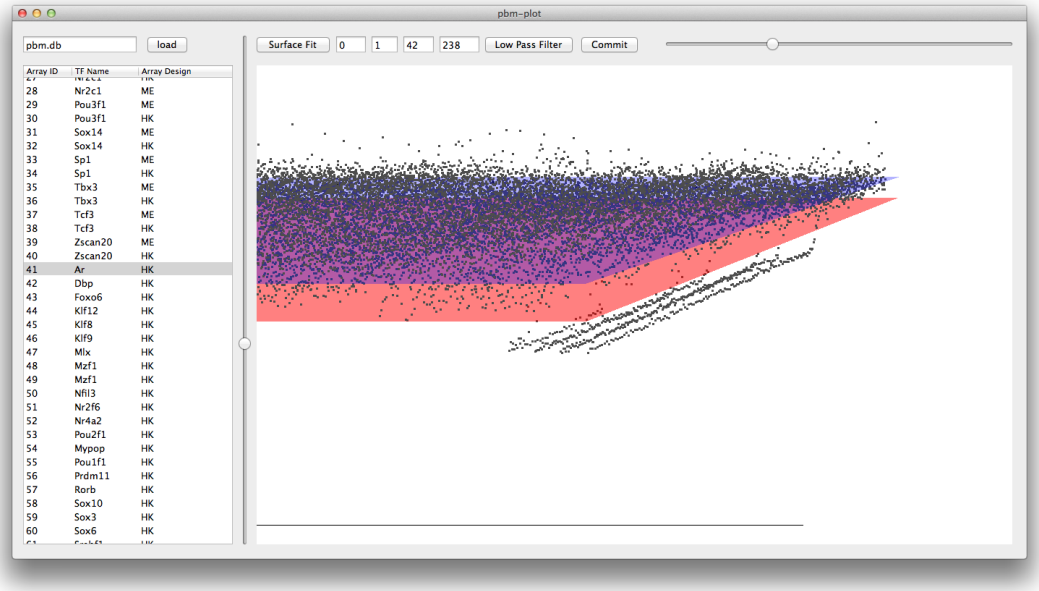


Figure A.4: By using the slider between the table-view and the graphics window the red plane can be raised to a desired height. By using the 'Low Pass Filter' button, any data below the red plane is flagged in the database and will be excluded from further analysis.

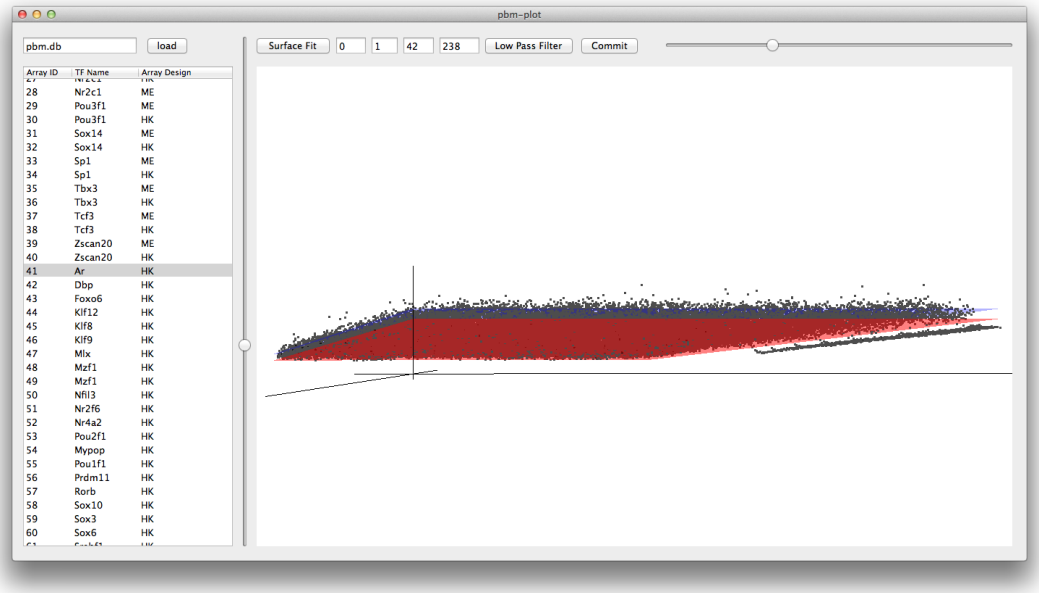


Figure A.5: It is possible to inadvertently clip data when making a low pass filter. Being able to move around the data enables us to spot potential problems such as this.

References

1. Jost, D., Zubair, A. & Everaers, R. Bubble statistics and positioning in superhelically stressed DNA. *Physical Review E* **84,** 031912 (2011) (cit. on p. 2).

2. Lipps, H. J. & Rhodes, D. G-quadruplex structures: *in vivo* evidence and function. *Trends in Cell Biology* **19,** 414–422 (2009) (cit. on p. 2).

3. Maune, H. T. *et al.* Self-assembly of carbon nanotubes into two-dimensional geometries using DNA origami templates. *Nature Nanotechnology* **5,** 61–66 (2010) (cit. on p. 2).

4. Gothelf, K. V. & Tørring, T. DNA nanotechnology: a curiosity or a promising technology? *F1000Prime Reports* **5** (2013) (cit. on p. 2).

5. Griswold, A. Genome packaging in prokaryotes: The circular chromosome of *E. coli*. *Nature Education* **1,** 57 (2008) (cit. on p. 2).

6. Segal, E. & Widom, J. From DNA sequence to transcriptional behaviour: a quantitative approach. *Nature Reviews Genetics* **10,** 443–456 (2009) (cit. on pp. 2, 3).

7. Denker, A. & de Laat, W. The second decade of 3C technologies: detailed insights into nuclear organization. *Genes & Development* **30,** 1357–1382 (2016) (cit. on p. 2).

8. De Wit, E. & de Laat, W. A decade of 3C technologies: insights into nuclear organization. *Genes & development* **26,** 11–24 (2012) (cit. on p. 2).

9. Wang, J. C. DNA topoisomerases: why so many? *Journal of Biological Chemistry* **266,** 6659–6662 (1991) (cit. on p. 3).

10. Yang, W. Nucleases: diversity of structure, function and mechanism. *Quarterly Reviews of Biophysics* **44,** 1–93 (2011) (cit. on p. 3).

11. Suzuki, M. M. & Bird, A. DNA methylation landscapes: Provocative insights from epigenomics. **9,** 465–476 (2008) (cit. on pp. 3, 25).

12. Von Hippel, P. H. & McGhee, J. D. DNA-protein interactions. *Annual review of biochemistry* **41,** 231–300 (1972) (cit. on p. 3).

13. Jacob, F. & Monod, J. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology* **3,** 318–356 (1961) (cit. on p. 5).

14. Ptashne, M. Isolation of the λ phage repressor. *Proceedings of the National Academy of Sciences* **57,** 306–313 (1967) (cit. on p. 5).

15. Gilbert, W. & Muller-Hill, B. ISOLATION OF THE LAC REPRESSOR. *Proceedings of the National Academy of Sciences* **56,** 1891–1898 (1966) (cit. on p. 5).

16. Riggs, A. D., Bourgeois, S., Newby, R. F. & Cohn, M. DNA binding of the lac repressor. *Journal of molecular biology* **34,** 365–368 (1968) (cit. on pp. 5, 32).

17. Gilbert, W. & Maxam, A. The Nucleotide Sequence of the lac Operator. *Proceedings of the National Academy of Sciences* **70,** 3581–3584 (1973) (cit. on p. 5).

18. Schmitz, A. & Galas, D. J. Sequence-specific interactions of the tight-binding I12-X86 lac repressor with non-operator DNA. *Nucleic Acids Research* **8,** 487–506 (1980) (cit. on p. 5).

19. Hawley, D. & McClure, W. Compilation and analysis of *Escherichia coli* promoter DNA sequences. *Nucleic Acids Research* **11,** 2237–2255 (1983) (cit. on p. 5).

20. Stormo, G. D. DNA binding sites: representation and discovery. *Bioinformatics* **16,** 16–23 (2000) (cit. on p. 5).

21. Schneider, T. D., Stormo, G. D., Gold, L. & Ehrenfeucht, A. Information content of binding sites on nucleotide sequences. *Journal of Molecular Biology* **188,** 415–431 (1986) (cit. on p. 5).

22. Stormo, G. D., Schneider, T. D. & Gold, L. Quantitative analysis of the relationship between nucleotide sequence and functional activity. *Nucleic Acids Research* **14,** 6661–6679 (1986) (cit. on pp. 5, 21).

23. Berg, O. G. & von Hippel, P. H. Selection of DNA binding sites by regulatory proteins. *Journal of Molecular Biology* **193,** 723–743 (1987) (cit. on p. 5).

24. Pabo, C. O. & Sauer, R. T. Protein-DNA Recognition. *Annual Review of Biochemistry* **53,** 293–321 (1984) (cit. on pp. 6, 7).

25. Finn, R. D. *et al.* Pfam: The protein families database. **42** (2014) (cit. on pp. 7, 10).

26. Babu, M. M., Luscombe, N. M., Aravind, L., Gerstein, M. & Teichmann, S. A. Structure and evolution of transcriptional regulatory networks. *Current Opinion in Structural Biology* **14,** 283–291 (2004) (cit. on p. 7).

27. Hubbard, T. J. P., Ailey, B., Brenner, S. E., Murzin, A. G. & Chothia, C. SCOP: a Structural Classification of Proteins database. *Nucleic Acids Research* **27,** 254–256 (1999) (cit. on p. 7).

28. Rose, P. W. *et al.* The RCSB Protein Data Bank: new resources for research and education. *Nucleic Acids Research* **41,** D475–D482 (2012) (cit. on p. 7).

29. Jones, S. An overview of the basic helix-loop-helix proteins. *Genome Biology* **5,** 226 (2004) (cit. on p. 10).

30. Murre, C., McCaw, P. S. & Baltimore, D. A new DNA binding and dimerization motif in immunoglobulin enhancer binding, daughterless, MyoD, and myc proteins. *Cell* **56,** 777–783 (1989) (cit. on p. 10).

31. Landschulz, W., Johnson, P. & McKnight, S. The leucine zipper: a hypothetical structure common to a new class of DNA binding proteins. *Science* **240,** 1759–1764 (1988) (cit. on p. 10).

32. Weigel, D., Jürgens, G., Küttner, F., Seifert, E. & Jäckle, H. The homeotic gene fork head encodes a nuclear protein and is expressed in the terminal regions of the *Drosophila* embryo. *Cell* **57,** 645–658 (1989) (cit. on p. 10).

33. Lam, E. W., Brosens, J. J., Gomes, A. R. & Koo, C.-Y. Forkhead box proteins: tuning forks for transcriptional harmony. *Nature Reviews Cancer* **13,** 482–495 (2013) (cit. on p. 10).

34. Štros, M., Launholt, D. & Grasser, K. D. The HMG-box: a versatile protein domain occurring in a wide variety of DNA-binding proteins. *Cellular and Molecular Life Sciences* **64,** 2590–2606 (2007) (cit. on p. 10).

35. Grosschedl, R., Giese, K. & Pagel, J. HMG domain proteins: architectural elements in the assembly of nucleoprotein structures. **10,** 94–100 (1994) (cit. on p. 10).

36. Klug, A. The Discovery of Zinc Fingers and Their Applications in Gene Regulation and Genome Manipulation. *Annual Review of Biochemistry* **79,** 213–231 (2010) (cit. on p. 10).

37. Miller, J., McLachlan, A. D. & Klug, A. Repetitive zinc-binding domains in the protein transcription factor IIIA from *Xenopus* oocytes. *The EMBO journal* **4,** 1609–14 (1985) (cit. on p. 10).

38. Urnov, F. D., Rebar, E. J., Holmes, M. C., Zhang, H. S. & Gregory, P. D. Genome editing with engineered zinc finger nucleases. *Nature Reviews Genetics* **11,** 636–646 (2010) (cit. on p. 15).

39. Tsai, S. Q. & Joung, J. K. Defining and improving the genome-wide specificities of CRISPR-Cas9 nucleases. *Nature Reviews Genetics* **17,** 300–312 (2016) (cit. on p. 15).

40. Oakes, B. L. *et al.* Multi-reporter selection for the design of active and more specific zinc-finger nucleases for genome editing. *Nature Communications* **7** (2016) (cit. on p. 15).

41. Bulyk, M. L. DNA microarray technologies for measuring protein-DNA interactions. *Current Opinion in Biotechnology* **17,** 422–430 (2006) (cit. on pp. 15, 16, 18).

42. Johnson, D. S., Mortazavi, A., Myers, R. M. & Wold, B. Genome-Wide Mapping of *in vivo* Protein-DNA Interactions. *Science* **316,** 1497–1502 (2007) (cit. on p. 16).

43. Bailey, T. L. DREME: motif discovery in transcription factor ChIP-seq data. *Bioinformatics* **27,** 1653–1659 (2011) (cit. on p. 16).

44. Galas, D. J. & Schmitz, A. DNAase footprinting a simple method for the detection of protein-DNA binding specificity. *Nucleic Acids Research* **5,** 3157–3170 (1978) (cit. on p. 16).

45. Boyle, A. P. *et al.* High-resolution genome-wide *in vivo* footprinting of diverse transcription factors in human cells. *Genome Research* **21,** 456–464 (2011) (cit. on p. 16).

46. Tuerk, C. & Gold, L. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science* **249,** 505–510 (1990) (cit. on p. 17).

47. Ogawa, N. & Biggin, M. D. High-throughput SELEX determination of DNA sequences bound by transcription factors *in vitro*. *Methods in Molecular Biology* **786,** 51–63 (2012) (cit. on p. 17).

48. Garner, M. M. & Revzin, A. A gel electrophoresis method for quantifying the binding of proteins to specific DNA regions: application to components of the *Escherichia coli* lactose operon regulatory system. *Nucleic Acids Research* **9,** 3047–60 (1981) (cit. on p. 17).

49. Brand, L. H. *et al.* Screening for Protein-DNA Interactions by Automatable DNA-Protein Interaction ELISA. *PLoS ONE* **8** (2013) (cit. on p. 17).

50. Boozer, C., Kim, G., Cong, S., Guan, H. & Londergan, T. Looking towards label-free biomolecular interaction analysis in a high-throughput format: a review of new surface plasmon resonance technologies. **17,** 400–405 (2006) (cit. on p. 17).

51. Maerkl, S. J. & Quake, S. R. A Systems Approach to Measuring the Binding Energy Landscapes of Transcription Factors. *Science* **315,** 233–237 (2007) (cit. on pp. 18, 26, 32, 33, 35, 37).

52. Mukherjee, S. *et al.* Rapid analysis of the DNA-binding specificities of transcription factors with DNA microarrays. *Nature Genetics* **36,** 1331–1339 (2004) (cit. on p. 18).

53. Berger, M. F. & Bulyk, M. L. Universal protein-binding microarrays for the comprehensive characterization of the DNA-binding specificities of transcription factors. *Nature Protocols* **4,** 393–411 (2009) (cit. on pp. 18, 45, 48, 55, 57).

54. Bulyk, M. L., Gentalen, E., Lockhart, D. J. & Church, G. M. Quantifying DNA-protein interactions by double-stranded DNA arrays. *Nature Biotechnology* **17,** 573–577 (1999) (cit. on pp. 18, 45).

55. Bailey, T. L., Williams, N., Misleh, C. & Li, W. W. MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Research* **34,** W369–W373 (2006) (cit. on pp. 22, 24).

56. Down, T. A., Piipari, M. & Hubbard, T. J. Dalliance: Interactive genome viewing on the web. *Bioinformatics* **27,** 889–890 (2011) (cit. on pp. 22, 160).

57. Das, M. K. & Dai, H.-K. A survey of DNA motif finding algorithms. *Bioinformatics* **8,** S21 (2007) (cit. on pp. 22, 24, 149).

58. Gershenzon, N. I. Computational technique for improvement of the position-weight matrices for the DNA/protein binding sites. *Nucleic Acids Research* **33,** 2290–2301 (2005) (cit. on p. 23).

59. Maynou, J. *et al.* Transcription factor binding site detection through position cross-mutual information variability analysis. **2009,** 7087–7090 (2009) (cit. on p. 23).

60. Marsan, L. & Sagot, M. F. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *Journal of Computational Biology* **7,** 345–362 (2000) (cit. on p. 23).

61. Vilo, J., Brazma, A., Jonassen, I., Robinson, A. & Ukkonen, E. Mining for putative regulatory elements in the yeast genome using gene expression data. *Proceedings. International Conference on Intelligent Systems for Molecular Biology* **8,** 384–94 (2000) (cit. on p. 23).

62. Tompa, M. *et al.* Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology* **23,** 137–144 (2005) (cit. on pp. 24, 149).

63. Schmidt, D. *et al.* Five-vertebrate ChIP-seq reveals the evolutionary dynamics of transcription factor binding. *Science* **328,** 1036–40 (2010) (cit. on p. 24).

64. Foat, B. C., Morozov, A. V. & Bussemaker, H. J. Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE. **22** (2006) (cit. on p. 25).

65. Down, T. A. & Hubbard, T. J. P. NestedMICA: sensitive inference of over-represented motifs in nucleic acid sequence. *Nucleic Acids Research* **33,** 1445–53 (2005) (cit. on pp. 25, 121, 150).

66. Kasowski, M. *et al.* Variation in transcription factor binding among humans. *Science* **328,** 232–235 (2010) (cit. on p. 26).

67. Zhao, Y., Granas, D. & Stormo, G. D. Inferring Binding Energies from Selected Binding Sites. *PLOS Computational Biology* **5,** 1–8 (2009) (cit. on p. 26).

68. Mintseris, J. & Eisen, M. B. Design of a combinatorial DNA microarray for protein-DNA interaction studies. *Bioinformatics* **7,** 429 (2006) (cit. on pp. 28, 96, 112, 113).

69. Orenstein, Y. & Shamir, R. Design of shortest double-stranded DNA sequences covering all k-mers with applications to protein-binding microar-

rays and synthetic enhancers. *Bioinformatics* **29,** i71–i79 (2013) (cit. on pp. 28, 88, 113, 118).

70. Annala, M., Laurila, K., Lähdesmäki, H. & Nykter, M. A Linear Model for Transcription Factor Binding Affinity Prediction in Protein Binding Microarrays. *PLoS ONE* **6** (ed Isalan, M.) e20059 (2011) (cit. on pp. 28, 55–58, 70, 71, 79, 85, 121, 122).

71. Weirauch, M. T. *et al.* Evaluation of methods for modeling transcription factor sequence specificity. *Nature Biotechnology* **31,** 126 (2013) (cit. on pp. 29, 44, 121).

72. Elf, J., Li, G.-W. & Xie, X. S. Probing Transcription Factor Dynamics at the Single-Molecule Level in a Living Cell. *Science* **316,** 1191–1194 (2007) (cit. on p. 31).

73. Hammar, P. *et al.* The lac repressor displays facilitated diffusion in living cells. *Science* **336,** 1595–1598 (2012) (cit. on pp. 31, 32, 148).

74. Revzin, A. & Von Hippel, P. H. Direct measurement of association constants for the binding of *Escherichia coli* lac repressor to non-operator DNA. *Biochemistry* **16,** 4769–4776 (1977) (cit. on p. 32).

75. Wieland, G. *et al.* Determination of the binding constants of the centromere protein Cbf1 to all 16 centromere DNAs of *Saccharomyces cerevisiae*. *Nucleic Acids Research* **29,** 1054–60 (2001) (cit. on p. 34).

76. Lourakis, M. I. a. A Brief Description of the Levenberg-Marquardt Algorithm Implemened by levmar. *Matrix* **3,** 2 (2005) (cit. on p. 38).

77. LeProust, E. Agilent's microarray platform: How high-fidelity DNA synthesis maximizes the dynamic range of gene expression measurements. *Agilent Technologies, Santa Clara, CA* (2008) (cit. on pp. 45, 47, 110).

78. Mandelkern, M., Elias, J. G., Eden, D. & Crothers, D. M. The dimensions of DNA in solution. *Journal of Molecular Biology* **152,** 153–161 (1981) (cit. on p. 45).

79. Dudley, A. M., Aach, J., Steffen, M. A. & Church, G. M. Measuring absolute expression with microarrays with a calibrated reference sample and an extended signal intensity range. *Proceedings of the National Academy of Sciences* **99,** 7554–7559 (2002) (cit. on pp. 46, 50).

80. Berger, M. F. *et al.* Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. *Nature Biotechnology* **24,** 1429–1435 (2006) (cit. on pp. 46, 107).

81. Guennebaud, G., Jacob, B., *et al. Eigen v3* http://eigen.tuxfamily.org. 2010 (cit. on pp. 57, 70, 106).

82. Smyth, G. K. & Speed, T. Normalization of cDNA microarray data. *Methods* **31,** 265–273 (2003) (cit. on p. 59).

83. Cleveland, W. S. Lowess: A program for smoothing scatterplots by robust locally weighted regression. *American Statistician* **35,** 54–55 (1981) (cit. on p. 59).

84. De Boor, C. *A practical guide to splines* (Springer-Verlag New York, 1978) (cit. on p. 60).

85. Workman, C. *et al.* A new non-linear normalization method for reducing variability in DNA microarray experiments. *Genome Biology* **3,** research0048 (2002) (cit. on p. 60).

86. Mecham, B. H., Nelson, P. S. & Storey, J. D. Supervised normalization of microarrays. *Bioinformatics* **26,** 1308–1315 (2010) (cit. on p. 60).

87. Runge, C. Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten.(German). *Zeitschrift für Mathematik und Physik* (1901) (cit. on p. 61).

88. Hastie, T., Tibshirani, R. & Friedman, J. *The Elements of Statistical Learning* 282 (Springer New York, New York, NY, 2009) (cit. on pp. 62, 67, 124).

89. Friedman, J., Hastie, T. & Tibshirani, R. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* **33** (2010) (cit. on pp. 70, 124, 125).

90. Bolstad, B., Irizarry, R., Astrand, M. & Speed, T. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **19,** 185–193 (2003) (cit. on p. 70).

91. Arazi, B. Position recovery using binary sequences. *Electronics Letters* **20,** 61–62 (1984) (cit. on p. 90).

92. Aguirre, G. K., Mattar, M. G. & Magis-Weinberg, L. de Bruijn cycles for neural decoding. *NeuroImage* **56,** 1293–1300 (2011) (cit. on p. 90).

93. Pevzner, P. A., Tang, H. & Waterman, M. S. An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences* **98,** 9748–9753 (2001) (cit. on p. 90).

94. Lin, Y.-L., Ward, C., Jain, B. & Skiena, S. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 595–606 (2011) (cit. on p. 90).

95. Van Aardenne-Ehrenfest, T. & de Bruijn, N. G. Circuits and trees in oriented linear graphs. *Simon Stevin* **28,** 203–217 (1951) (cit. on p. 95).

96. Artin, M. *Algebra* (Prentice Hall, 1991) (cit. on p. 95).

97. Philippakis, A. A., Qureshi, A. M., Berger, M. F. & Bulyk, M. L. Design of Compact, Universal DNA Microarrays for Protein Binding Microarray Experiments. *Journal of Computational Biology* **15,** 655–665 (2008) (cit. on pp. 95, 100–102).

98. Paterson, K. G. *On sequences and arrays with specific window properties* PhD thesis (University of London, 1993) (cit. on pp. 110, 112, 118).

99. Berlekamp, E. *Algorithmic Coding Theory* 1968 (cit. on p. 112).

100. Davis, T. A. *User's Guide for SuiteSparseQR, a multifrontal multithreaded sparse QR factorization package* 2012 (cit. on p. 116).

101. Down, T. A. Computational Detection and Location of Transcription Start Sites in Mammalian Genomic DNA. *Genome Research* **12,** 458–461 (2002) (cit. on p. 121).

102. Doğruel, M., Down, T. A. & Hubbard, T. J. NestedMICA as an ab initio protein motif discovery tool. *Bioinformatics* **9,** 19 (2008) (cit. on p. 121).

103. Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological),* 267–288 (1996) (cit. on p. 123).

104. Zou, H. & Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67,** 301–320 (2005) (cit. on p. 123).

105. Badis, G. *et al.* Diversity and Complexity in DNA Recognition by Transcription Factors. *Science* **324,** 1720–1723 (2009) (cit. on pp. 126, 128).

106. Newburger, D. E. & Bulyk, M. L. UniPROBE: An online database of protein binding microarray data on protein-DNA interactions. *Nucleic Acids Research* **37** (2009) (cit. on p. 128).

107. Yue, F. *et al.* A comparative encyclopedia of DNA elements in the mouse genome. *Nature* **515,** 355–364 (2014) (cit. on pp. 148, 152).

108. Berg, O. G., Winter, R. B. & Von Hippel, P. H. Diffusion-driven mechanisms of protein translocation on nucleic acids. 1. Models and theory. *Biochemistry* **20,** 6929–6948 (1981) (cit. on p. 148).

109. Bauer, M., Rasmussen, E. S., Lomholt, M. A. & Metzler, R. Real sequence effects on the search dynamics of transcription factors on DNA. *Scientific Reports* **5** (2015) (cit. on p. 148).

110. D'haeseleer, P. What are DNA sequence motifs? *Nature Biotechnology* **24,** 423–425 (2006) (cit. on p. 149).

111. Mathelier, A. *et al.* JASPAR 2014: An extensively expanded and updated open-access database of transcription factor binding profiles. *Nucleic Acids Research* **42** (2014) (cit. on pp. 149, 155).

112. Arceci, R. J., King, A., Simon, M. C., Orkin, S. & Wilson, D. Mouse GATA-4: a retinoic acid-inducible GATA-binding transcription factor expressed in endodermally derived tissues and heart. *Molecular and Cellular Biology* **13,** 2235–2246 (1993) (cit. on p. 151).

113. Kyo, S. *et al.* NF-IL6 represses early gene expression of human *papillomavirus* type 16 through binding to the noncoding region. *Journal of virology* **67,** 1058–66 (1993) (cit. on p. 151).

114. Kophengnavong, T., Michnowicz, J. E. & Blackwell, T. K. Establishment of Distinct MyoD, E2A, and Twist DNA Binding Specificities by Different Basic Region-DNA Conformations. *Molecular and Cellular Biology* **20,** 261–272 (2000) (cit. on p. 151).

115. Zhang, Y. *et al.* Model-based analysis of ChIP-Seq (MACS). *Genome Biology* **9** (2008) (cit. on p. 152).

116. Jayaram, N., Usvyat, D. & Martin, A. C. Evaluating tools for transcription factor binding site prediction. *Bioinformatics,* 1 (2016) (cit. on p. 156).

117. Pique-Regi, R. *et al.* Accurate inference of transcription factor binding from DNA sequence and chromatin accessibility data. *Genome Research* **21,** 447–455 (2011) (cit. on p. 172).