Chapter 1

Introduction

The genetic information stored in our DNA is transcribed into RNA by large molecular holoenzymes called RNA polymerases. In eukaryotic organisms there are three types of RNA polymerases, out of which RNA polymerase II (Pol II) is the one responsible for transcribing protein-coding genes and many noncoding RNAs such as micro-RNAs (Megraw et al., 2009; Saltzman and Weinmann, 1989). Pol II activity is highly regulated at the level of the individual transcript, and this regulation is essential for both cellular homeostasis and development of multicellular organisms (Fuda et al., 2009). The most central and best understood mechanisms of gene regulation is mediated by the interaction of sequence specific transcription factors (TFs) with DNA target sequences, each other and with other members of the Pol II complex (Mitchell and Tjian, 1989). Transcription factors orchestrate the transcription cycle because their activities are in turn controlled by cellular signals, for instance on the level of post-transcriptional modifications and protein-protein interactions. Each factor has a preference towards a specific set of DNA words which dictates the positions at which it is recruited to the genome. As this mechanism of DNA site recognition acts in part to choose the target genes of the transcription factors, the DNA patterns are commonly known as 'regulatory motifs'.

In this introduction I firstly outline the known regulatory mechanisms acting on the level of transcription to highlight the importance of and challenges in the study of transcriptional regulatory mechanisms (Section 1.1). I then briefly review the previous literature on computational regulatory motif inference (Section 1.2),

before introducing the specific computational methodology used in the project (Section 1.3). I then discuss the biological resources which were applied in this work (Section 1.4), and finally introduce the specific contributions in this work to the inference and classification of regulatory motifs (Section 1.5).

1.1 Gene regulation by control of transcription

Transcription factors act by promoting or inhibiting the recruitment of Pol II to the gene's promoter, to initiate RNA transcription at the transcription start site (TSS) of the gene, eventually leading to the generation of a full-length RNA transcript. This classical understanding of eukaryotic transcriptional regulation – involving only proximally located transcription factor binding sites (TFBS) – has had to give way to a more complex view of regulatory interactions. Firstly, factors which interact with Pol II not only act to recruit it to the complex, but can also affect its post-initiation clearance from the promoter, elongation of the transcript, and its termination, all of which are found to be rate-limiting and therefore highly likely regulated steps in the case of some genes (Venters and Pugh, 2008). Secondly, regulatory regions are found not only proximal to the TSS, but also kilobases further upstream, or even downstream, of their target genes in an orientation independent manner (Banerji et al., 1981).

The more distal regulatory regions are known as "enhancer" regions when they have an activatory role, and "silencer" regions when they inhibit recruitment of the transcriptional machinery (Visel et al., 2009). Several large studies have been conducted and are currently underway to systematically discover and catalogue tissue specific enhancers acting in mammalian and fish genomes (Ellingsen et al., 2005; Pennacchio et al., 2006; Visel et al., 2008). Enhancer- and silencer-like regions, as well as insulators which set the 'borders' of the chromatin domains regulated by enhancers and silencers, have also been described in yeasts (Bi and Broach, 2001; Buchman et al., 1988). The chromatin packaging of the genome sets limits to the regions that are available for transcription factor binding, and regulatory interactions that control this process can both activate and repress expression (Li et al., 2007; Steinfeld et al., 2007; Venters and Pugh, 2008). Figure 1.1A depicts these various factors and interactions involved in transcriptional

regulation.

General transcription factors (GTFs) bind to specific target sequences close to the transcription start site (TSS) at defined locations (Venters and Pugh, 2008), as shown in Figure 1.1A. Names and approximate positions are shown for the GTF target sequences. Regulatory transcription factors bind either to activate or repress the transcription of the target gene by binding to their target DNA sequences either near the core promoter or more distally (enhancers). Interactions between the TFs, GTFs and the Pol II are also important for regulation. Co-regulators which do not themselves bind DNA in a sequence-specific manner also interact with GTFs, TFs and nucleosomes (via modified histone tails). Both activation and repression can occur via each of these interactions.

Trans-acting enhancer regions are thought to contribute to eukaryotic gene regulation by looping DNA to promote the recruitment of the transcription machinery at a TSS (Figure 1.1B). Many genes are known to achieve their observed expression patterns through the combination of weak promoters and enhancer regions, which supplement them. In this example the expression pattern of a gene is modulated by both a promoter, as well as brain and limb specific enhancer elements. Silencer elements, which were not depicted here, can also act from a large distance to the TSS.

Enhancers and silencers rely on the organisation of genes into chromosomal domains that can in part be co-regulated. However, it has also been suggested that TF target genes are organised non-randomly for the majority of TFs, even in *S. cerevisiae* with its compact non-coding genome (Janga et al., 2008), short promoter sequences and relatively few examples of long-distance enhancer or silencers. The organisation of targets of a TF along chromosomes, possibly through their association in shared three-dimensional 'chromosomal territories' (Cremer and Cremer, 2001; Gasser, 2002; Lieberman-Aiden et al., 2009), could pose yet another largely uncharacterised level of regulatory information. The effect of neighbouring genes sharing similar promoter motifs has also been shown in *D. melanogaster* (Zhu and Halfon, 2009).

Another mechanism of transcriptional regulation not depicted above is the tissue or time specific use of alternative TSSs. The majority of human and mouse Pol II promoters have clusters of close TSSs instead of a single one (Frith et al.,

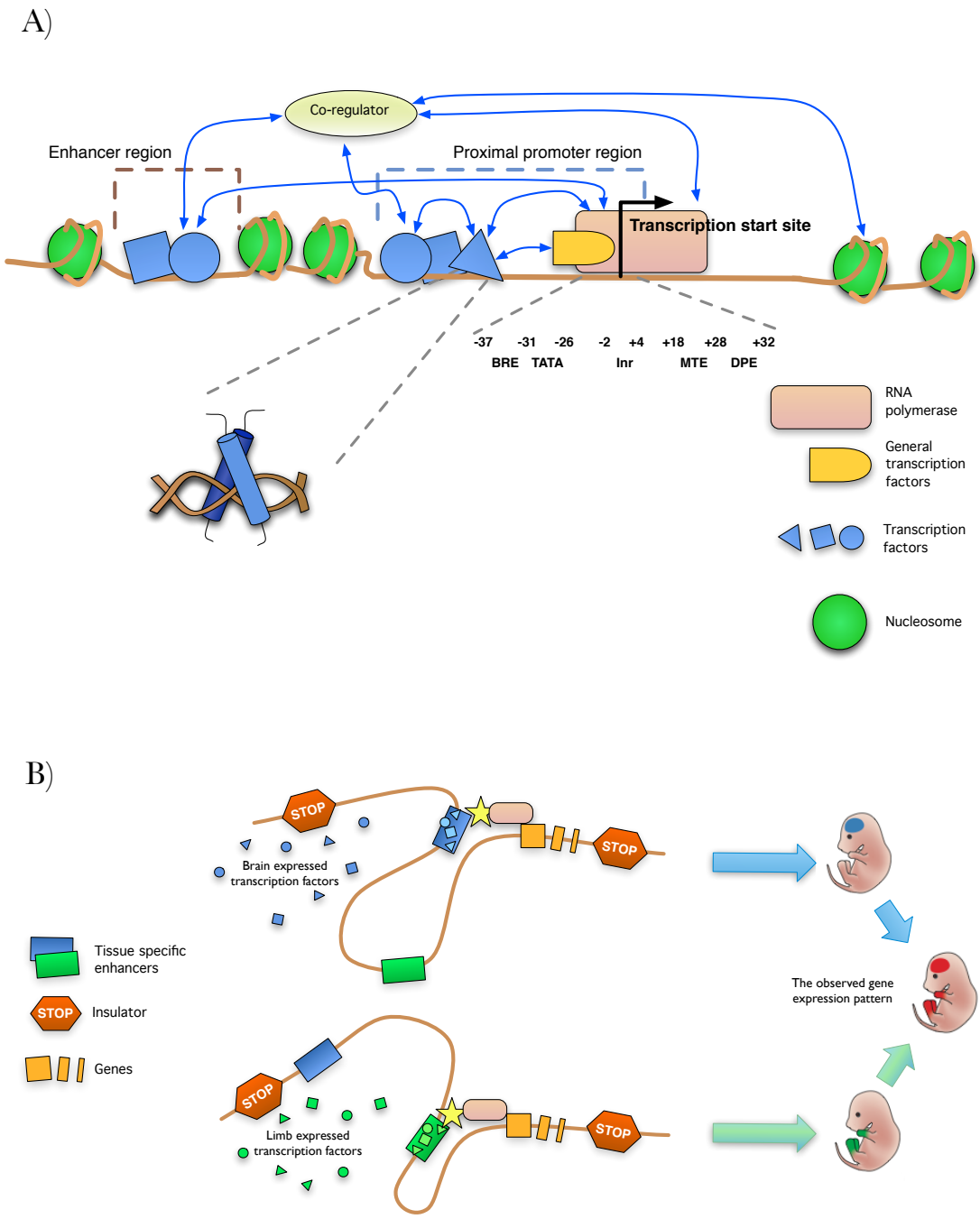

Figure 1.1: Key regulatory interactions which modulate transcription initiation. A) A promoter centric view on transcriptional regulation. Transcription factors interact with DNA and other regulatory factors to modulate the action of the RNA polymerase. B) An enhancer centric view on transcriptional regulation. Figure adapted from Visel et al. (2009) and Fuda et al. (2009).

2008). Larger scale TSS usage variation also occurs. Alternative promoter usage can in fact act as a mechanism for creating variant isoforms of gene products (Carninci et al., 2006), and changes in alternative TSS use are found associated to tissue and developmental stage specific dynamics of transcription (Consortium et al., 2009; Valen et al., 2008).

The identification and study of gene regulatory sequence is more difficult than protein-coding sequence because of several factors. Perhaps most importantly, the conservation pattern of regulatory sequence does not resemble that of protein coding sequence. Purifying selective constraint in regulatory sequences is often seen between closely related species (Hardison, 2000; Loots et al., 2000; Ludwig, 2002), but genomic TF binding studies suggest that turnover of regulatory elements occurs at remarkably high rate even when expression pattern (i.e. the connectivity of the TF network) shows little change (Schmidt et al., 2010). Indeed, changes in regulatory interactions have been hypothesised to be a cause of species divergence both in fungi (Borneman et al., 2007) and in animals (Carroll et al., 2000; Galant and Carroll, 2002). Furthermore, regulatory elements are often not constrained in the ordering, orientation or number of functional sites (Ludwig, 2002; Markstein and Levine, 2002). Consequently, alignment based comparative methods, which have been largely developed for the study of protein coding DNA, suffer from misalignments. For instance only 59% agreement is found between methods in the case of the 12 whole-genome *Drosophila* genomes aligned in the study by Stark et al. (2007). Detecting selective constraint acting on short blocks – often less than 20bp long (Bergman and Kreitman, 2001) – is not easy. Indeed, alignment based comparative analyses can only identify a small fraction of functional elements (Siggia, 2005). Alignment free *cis*-regulatory motif discovery methods which can consider recurring signals between related species to be conserved regardless of alignment or orientation are only beginning to appear (Gordan et al., 2010; Kim et al., 2010; Xie et al., 2009).

TF binding sites frequently occur in clusters – homotypic or heterotypic (Gotea et al., 2010). Site proximity of different TFs can modulate both cooperative and repressive interactions between different TFs (Kulkarni and Arnosti, 2005; Lebrecht et al., 2005), and competition of TFs for overlapping TFBSs is known to contribute for instance to *Drosophila* embryo segmentation (Walter

et al., 1994). Repetitive (homotypic) clustering of sites for the same TF is also well documented and can act to ensure stable binding (Cunningham and Cooper, 1993) or modulate a graded transcriptional response (Donahue et al., 1983). Interestingly, it has been suggested that even proximal or overlapping spacing of sites might be produced by selection mechanisms acting to maintain the overall composition of TFBSs in *cis*-regulatory elements instead of a constraint acting to maintain binding site position or orientation (Lusk and Eisen, 2010).

1.1.1 Sequence specific transcription factors

Understanding properties of *cis*-regulatory sequences is an ongoing challenge faced by the field of regulatory genomics. Another challenge which similarly continues to require extensive experimental and computational work is the annotation of transcription factors in genomes. High coverage annotations of TF genes are available for some well studied organisms in manually curated databases, ranging from RegulonDB for *Escherichia coli* (Huerta et al., 1998; Salgado et al., 2006), DBTBS for *Bacillus subtilis* (Ishii et al., 2001; Sierro et al., 2008), FlyBase (Wilson et al., 2008b) and FlyTF (Adryan and Teichmann, 2006; Pfreundt et al., 2010) for *Drosophila*, TFdb (Kanamori et al., 2004) and TFCat (Fulton et al., 2009) for human and mouse.

Advanced comparative sequence analysis techniques based on the use of protein domain profile Hidden Markov models have been helpful in systematically predicting large numbers of transcription factors for many sequenced genomes, both eukaryotic and prokaryotic (Kummerfeld and Teichmann, 2006; Wilson et al., 2008a). To illustrate the insight that TF annotation gives about transcription regulation, a comparison is shown below between the number of predicted sequence specific transcription factor genes out of the total number of protein coding genes for four eukaryotic species, as well as the *E. coli* (K12). The data presented is from the DBD database (Wilson et al., 2008a) (Release 2.0, downloaded 12/6/2010) which predicts TFs based on statistically significant matches to protein domain models from either the PFAM (Finn et al., 2010; Sonnhammer et al., 1997) or the SUPERFAMILY (Gough et al., 2001; Wilson et al., 2009) databases.

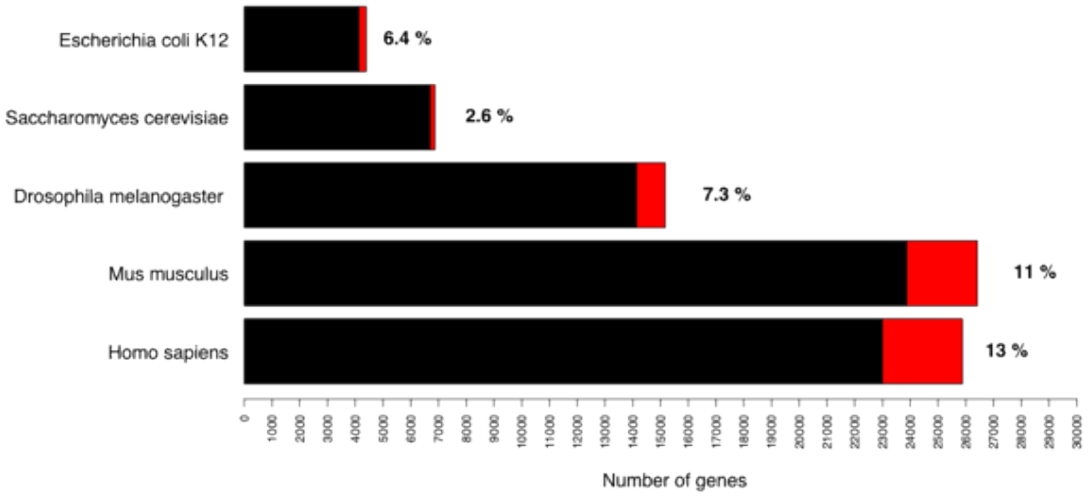

Figure 1.2: TF counts versus gene counts. The data presented is from the DBD database (Wilson et al., 2008a).

The TF number comparison shown in Figure 1.2 highlights several properties of transcriptional regulation. Firstly, the number, and more interestingly the proportion of TFs, increases for large genomes. For example in the case of the human genome, 13% of its approximately 23,000 genes are predicted to be TFs, whereas only 2.6% out of the 6,700 *Saccharomyces cerevisiae* genes are annotated as TFs. The increase in fraction of regulatory factors from the total number of genes amongst eukaryotes is thought to be a manifestation of the increased need to specifically regulate genes in larger, more complex organisms.

Single-cellular eukaryotic genomes contain a smaller fraction of TFs from total gene number when compared to bacteria (*S. cerevisiae* at 2.6%, *E. coli* at 6.4%). This is a well documented observation and thought to be a result of tissue and condition specific combinatorial regulation of genes in eukaryotes (van Nimwegen, 2003), epigenetic regulation (Choi and Kim, 2008), as well as the additional post-transcriptional control mechanisms such as microRNAs that are abundant in higher eukaryotes but absent in some fungi such as *S. cerevisiae* (Grimson et al., 2008). A power-law relationship has been described between the genome size and the number of TFs present in a genome, both in eukaryotes and prokaryotic organisms, with a lower exponent in eukaryotes (van Nimwegen, 2003).

Known binding site motifs of eukaryotic TFs tend to be less constrained than bacterial motifs (Wunderlich and Mirny, 2009). This together with the much larger genome sizes of eukaryotes also points at the requirement for additional levels of regulation. To put it simply, the DNA motif of a eukaryotic TF does not contain enough information to help it distinguish its cognate sites from non-functional sites that could occur as often as every $10^3 - 10^4$ nucleotides (assuming a simple genomic background model parameterised by average GC content). This view is supported by *in vivo* ChIP-seq binding studies of genomic binding sites of several eukaryotic TFs: assumably non-functional binding far from genes is found to be abundant in several studies (Robertson et al., 2007; yong Li et al., 2008). Abundant non-functional binding of TFs was in fact observed already in a much more laborious UV-crosslinking and Southern blot study by Walter et al. (1994).

Clustering of TFBSs can provide additional regulatory information by allowing combinatorial binding of TFs (Georges et al., 2010; Makeev et al., 2003; Papatsenko, 2009). More recently, a large scale analysis of human and mouse TF protein-protein interactions and expression measurements of the factors strongly suggests the combined action of sequence specific TF complexes, most importantly homeobox factors, in cell fate specific regulation of target genes (Ravasi et al., 2010). Homeobox factors are interesting in this context because they are especially common in mammals (Wilson et al., 2008a), they have short five or six nucleotide long motifs (Affolter et al., 2008) and they often bind with an additional, specific co-factor in a manner specific to cell-type (Ravasi et al., 2010). In conclusion, in higher eukaryotes it is important to consider gene regulation as a combination of multiple mechanisms including for instance increased combinatorial interactions of TFs, multiple classes of noncoding RNAs (Jacquier, 2009), epigenetic mechanisms (Jaenisch and Bird, 2003) and alternative transcripts (Carninci et al., 2006).

When the TFs of each organism are grouped by the content of their DNA binding families (Figure 1.3), it becomes apparent that TFs of all the organisms shown here fall into a much smaller number of DNA binding domains (e.g. 155 domains in 2886 human TFs, or 46 domains in 177 *S. cerevisiae* TFs). The low overlap between TF domain content of different genomes highlights that many of

the TF families have expanded within specific lineages (Babu et al., 2004). For example, the overlap between domains annotated in *H. sapiens* and *E. coli* is only four domains (HTH_3, HTH_{11}, CSD and PAS domains) whereas the mammals *H. sapiens* and *M. musculus* share 151 domains. The reader is referred to Wilson et al. (2008a) for a more thorough discussion of the kingdom specific expansion of DNA binding domains.

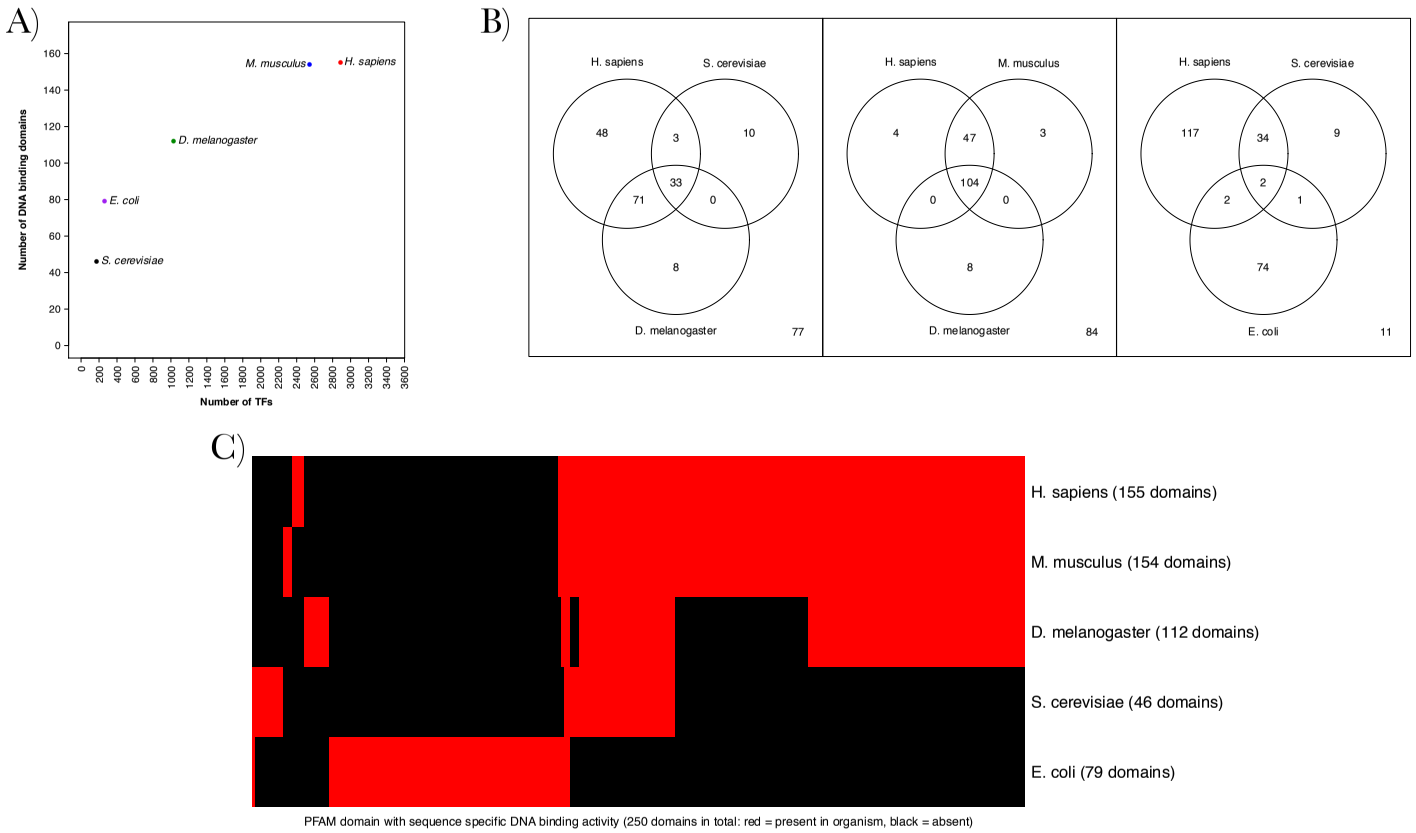

Figure 1.3: The TF domain coverage of genomes. A) Number of TF domain families vs TF gene counts in a genome. B) Overlap of TF domains between different organisms, represented in a Venn diagram. C) Overlap of TF domains between different organisms, represented as a heatmap (red = domain present, black = domain absent in organism). Data presented here originates from the DBD database (Wilson et al., 2008a).

1.1.2 Binding specificity of transcription factors

Determining the genomic binding sites and modelling sequence specificity patterns of TFs has proven a formidable task. Currently the only eukaryotic organism for which binding specificity of the large majority of its transcription factors has been determined based on DNA–protein interaction assays is *Saccharomyces cerevisiae*, which has a small genome by eukaryotic standards (12 Mbases with 6,532 protein coding genes according to the Ensembl release 58.1j (Hubbard et al., 2009)). I will take special interest here in discussing *S. cerevisiae* because what is already known of its transcriptional regulation is the closest that we currently have to a 'regulatory code' of any eukaryotic genome, and because computational genome scale regulatory motif inference in *S. cerevisiae* is the focus of the work described in Chapter 5.

The first large scale effort towards the *in vivo* profiling of TF binding on a genome scale was the study by Harbison et al. (2004), where ChIP-chip assays were conducted with 203 sequence specific TFs, each factor's binding profile being measured in one or more of 12 different growth conditions. The original analysis of the paper detailed a high confidence motif for 63 of the 203 TFs studied. MacIsaac et al. (2006) then provided a re-analysis of the large dataset with two phylogenetic foot printing based inference algorithms. PhyloCon (Wang and Stormo, 2003) and Converge (MacIsaac et al., 2006) yield motifs for an additional 36 TFs. The resolution of the ChIP-chip assay however does not reach beyond 500nt due to the limitations set by the use of randomly sheared genomic DNA fragments and tiling arrays (Sikder and Kodadek, 2005). ChIP-chip in other words is not ideal for determining accurate binding site profiles for TFs. ChIP followed by sequencing (ChIP-seq) offers a partial solution to the resolution problem, and allows more accurate and quantifiable *in vivo* study of protein-DNA binding. ChIP-seq assays with TFs have been to date conducted with TFs of larger, higher eukaryote genomes [1], with the exception of Lefrançois et al. (2009) who assayed a series of budding yeast TFs as a proof of concept of a multiplexed ChIP-seq experiments (a single sequencing experiment contains samples for multiple TFs). *In vitro*

[1] Large scale efforts to profile sequence specific TF specificity in human and several model organisms *in vivo* with ChIP-seq have begun as part of the ENCODE and modENCODE projects. See `http://www.genome.gov/10005107` for more information.

measurements of TF DNA specificity however already provide a close-to-complete, high resolution dataset for the *S. cerevisiae*: a protein binding microarray (PBM) Mukherjee et al. (2004) based study by Zhu et al. (2009), and a study by Badis et al. (2008) using a combination of PBMs, cognate site identifier microarrays (Warren et al., 2006), and DIP-chip (Liu et al., 2005).

Our knowledge of sequence specific protein–DNA interactions is far less complete in the case of larger eukaryotic genomes than it is in the budding yeast. The JASPAR database (Portales-Casamar et al., 2010), which contains a high quality non-redundant resource of TFBS motifs for different kingdoms of life, contains only 75 TFBS motifs for the 2886 TFs in human. For mouse there are only 40 TFs present in JASPAR (out of 2548 TFs). Furthermore, most high throughput studies to date have concentrated on a small number of highly expanded TF domain families, such as homeodomains (Noyes et al., 2008a) and basic helix-loop-helix factors (Grove et al., 2009; Maerkl and Quake, 2009), with the exception of Badis et al. (2009) whose 104 TFBS motifs cover 22 different families of TFs. New high-throughput methods for studying DNA–protein interactions are becoming available in addition to universal PBMs which currently provide majority of the publicly available high-throughput TF–DNA specificity data. These new promising methodologies include ChIP-seq (Robertson et al., 2007), bacterial one-hybrids (Meng and Wolfe, 2006; Noyes et al., 2008a,b), multiplexed massively parallel SELEX (Jolma et al., 2010) and a microfluidic molecular interaction assay platform by Maerkl and Quake (2007a).

Although new protein–DNA interaction probing technologies have the potential to transform our knowledge of eukaryotic transcriptional regulation, it is also clear that efficient computational methods for motif inference and classification continue to be of key importance. My aim in Chapter 2 is to present a new class of motif family models that can be learned using experimentally determined PWM motifs, such as those derived from new HT technologies. In Chapters 3 and 4 I present applications of motif family models for sensitively inferring motifs from genomic sequence, and for classifying computationally inferred motifs by their DNA binding domain type, respectively. In both of these lines of work use experimentally determined motif data to provide a comparison for evaluating computational predictions. Experimentally determined regulatory motifs are also

central to the *S. cerevisiae* motif inference performance benchmark in Chapter 5, where *de novo* predictions are compared to experimental motifs.

1.2 Computational inference of transcription factor binding site motifs

Computational inference of TFBSs by applying short motif inference algorithms to pieces of genomic DNA sequence is a long-standing research problem. It has motivated computational biologists to propose literally hundreds of algorithms over the course of more than 30 years. Many of these algorithms are introduced in previous reviews (Das and Dai, 2007; MacIsaac and Fraenkel, 2006; Nguyen and Androulakis, 2009; Sandve and Drabløs, 2006), and therefore only essentials of different approaches are covered here.

The first motif inference algorithm was published in the landmark paper by Korn et al. (1977) where pairwise comparisons of aligned sequence immediately close to prokaryotic transcription start sites (TSS) and terminator sequences were used to infer recurring motifs. The Korn et al. (1977) approach, which simply lists recurring sequence words found by pairwise comparisons of noncoding DNA sequence, is the earliest precursor to oligonucleotide word enumeration based motif inference algorithms. Such algorithms aim to exhaustively list possible k-mers that satisfy an objective function such as a conservation or significance score, commonly allowing a certain maximum number of mismatches. This approach is still taken in several recently published algorithms, ranging from reporting ranked k-mers of a specified length (Helden et al., 1998; van Dongen et al., 2008) to IUPAC consensus strings that allow for describing degeneracy in positions (Marschall and Rahmann, 2009; Xie et al., 2005, 2007). In fact the Tompa et al. (2005) *ab initio* motif inference method benchmark showed the word-enumeration based Weeder (Pavesi et al., 2001) as one of the best performing inference method of the 13 methods that were tested. The Tompa et al. (2005) benchmark is discussed in more detail in Section 5.1.2. Enumeration based methods can be made computationally very fast through the use of modern computers with access to a large volume of runtime memory together with highly optimised look-up data

structures, such as suffix trees which were originally introduced in computational biology detection of repeat elements (Sagot, 1998).

Word enumeration methods however have certain inherent limitations. Firstly, the reliance on lookup based data structures make them incapable of modelling very long TFBS patterns – 8-mers or 10-mers are typically studied – which are known to be present amongst eukaryotic TFBS motifs of many TF families. Cys_2His_2 zinc finger motifs for instance can be as long as 15 or 20 nucleotides due to the common architecture of their protein–DNA interaction which involves several zinc finger domains binding in tandem (LeClerc et al., 1991; Wolfe et al., 2000). Motifs with a large number of weakly constrained positions are also problematic for word enumeration methods which generally require sequence word clustering based on edit distance to group individual related sequence words to motif models to describe degeneracy. The great majority of TFs do not bind to a unique DNA 'word', but instead they show a distribution of binding affinity across a number of possible sites (known as 'degeneracy'). Degenerate positions are well known to occur in TFBS motifs (examples with degenerate motifs are shown in Figure 1.4), and the information content of a position has been shown to correlate with its conservation (Moses et al., 2003) and the number of contacts the base makes with amino acid residues (Gelfand and Mirny, 2002). Genome scale *in vivo* profiling of transcriptional control is rapidly forming an image of transcriptional control where not only is a large spectrum of possible binding sequences observed (Badis et al., 2009), but also that even weak binding sites can exert a regulatory response (Gertz et al., 2009) and therefore are biologically meaningful. Therefore, models of sequence motifs should ideally represent the sequence specificity distribution as completely as possible, whilst being able to weight strongly binding sequences above weakly binding sequences, neither of which is possible with k-mer enumeration based models.

The above-mentioned limitations of word enumeration methods in describing transcription regulatory motifs resulted in development of probabilistic motif inference methods, which most commonly use the position weight matrix (PWM) as the motif model. The PWM is described in more detail in Section 1.2.1, and examples of PWMs are shown in Figure 1.4 as sequence logos (Schneider and Stephens, 1990).

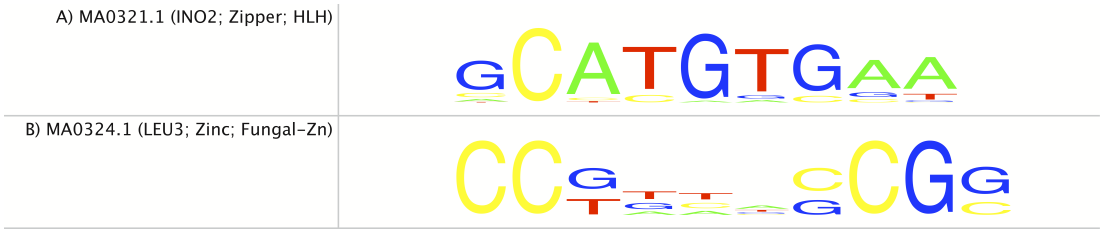

Figure 1.4: Strongly constrained PWM motif, and one with degenerate positions in the middle.

Some motif inference algorithms consider other data in addition to genomic sequence to support the inference task. Especially gene expression data has been made use of in various approaches. The earliest method for this is gene expression clustering based division of genes to sets followed by motif inference from the individual sets (Roth et al., 1998). More sophisticated methods have been developed for gene expression, beginning from the pioneering multivariate regression based word enumeration framework REDUCE by Bussemaker et al. (2001) that sparked publication of several related methods; c-REDUCE considers conservation as well as gene expression evidence (Kechris and Li, 2008), and MatrixREDUCE (Foat et al., 2005) models binding sites as position-specific affinity matrices (essentially PWMs where nucleotide weights correspond to the binding affinity of the nucleotide relative to the nucleotide with the optimal affinity). Other multivariate regression based approaches for using expression data in motif inference have also been published (Conlon et al., 2003; Keleş et al., 2002).

Ranking or weighting sequences based on position specific properties such as TF or nucleosome binding (based on ChIP-chip) has also been applied in motif discovery (Conlon et al., 2003; Liu et al., 2002). This approach has been expanded in the PRIORITY algorithm (Narlikar et al., 2006) to a general nucleotide position specific prior function suited for instance for specifying information about the sought after TF domain family specificities (Narlikar et al., 2006), nucleosome positioning (Narlikar et al., 2007), DNA duplex stability (Gordân and Hartemink, 2008), or sequence conservation information (Gordan et al., 2010).

Phylogenetic foot printing methods that apply sequence conservation, alignment of orthologous sequences and phylogenetic models of regulatory regions to

improve sensitivity to detect regulatory motif and *cis*-regulatory modules have also been developed (Siddharthan, 2008; Sinha et al., 2004; Wang and Stormo, 2003).

In conclusion, a multitude of different approaches have been applied to regulatory motif inference. Finding a suitable algorithm for a biological problem at hand can be a daunting task for a researcher, and indeed one might expect that standard benchmarking methods would have surfaced in the literature of motif inference algorithms. However, the great majority of the above mentioned publications describing motif inference algorithms are either:

1. applied to a specific biological problem without an explicit performance assessment with other algorithms.

2. compared with a publication specific biological dataset with one, two or a handful of different common tools such as MEME (Bailey and Elkan, 1995).

3. compared with a synthetic sequence set with one, two or a handful of different common tools.

Performance comparison of motif inference tools is itself a non-trivial problem. Very few comprehensive attempts have been made to date to systematically assess different tools (Li and Tompa, 2006; Pevzner and Sze, 2000; Sinha and Tompa, 2003a; Tompa et al., 2005). The assessment by Tompa et al. (2005) is perhaps the most comprehensive to date, covering 13 different algorithms. In Chapter 5 I discuss the challenges of measuring motif inference performance with synthetic and real promoter sequence (Section 5.1.3), and describe a new, large scale motif inference benchmark challenge (Section 5.3.2).

1.2.1 The position weight matrix

The PWM, also known as a position specific scoring matrix (PSSM) or a gapless profile, is a commonly used probabilistic model used in motif inference algorithms. It has been found to preserve more of the information of individual motif positions (columns) than consensus string motifs, and to systematically perform better in

describing regulatory binding site patterns (Osada et al., 2004). It is also the motif model of choice in my work.

PWMs are probabilistic sequence motif models that can be scanned along sequence to assign a score for a sequence window to contain a motif match. Commonly a threshold is determined for the sequence window scores, such that windows where the threshold is exceeded are called motif matches (potential binding sites). A large part of my work has revolved around analysing properties of inferred PWM motifs and their connection to previously known motifs (Chapters 2, 3, 4) with the use of motif family models. In addition, in Chapter 5 I present an assessment of the prediction performance of several *de novo* motif discovery algorithms. A formal definition of the PWM is therefore in place, and provided below (adapted from Rahmann et al. (2003)).

Let $\mathbb{A}$ be a finite alphabet with cardinality $|\mathbb{A}|$ ($|\mathbb{A}| = 4$ for DNA and RNA). If $\mathbb{A}^k$ represents the space of all string of k symbols from $\mathbb{A}$, a PWM $\mathbf{M}$ is a probability distribution over all of the sequence positions i of $\mathbb{A}^k$. More specifically, $\mathbf{M}$ is an $|\mathbb{A}| \times k$ matrix where each column vector $\mathbf{M}_i$ represents the weights $m_{i,j}$ (nucleotide j at sequence position i) for a multinomial distribution, i.e. $\mathbf{M}_{i,j}$ are nonnegative such that $(\Sigma_{ie\mathbb{A}}\mathbb{A}_i = 1)$.

$\mathbf{M}$ is thought of as a generative model for sequences from $\mathbb{A}^k$ such that symbol s at each position i is generated independently according to the multinomial distribution parametrised by $\mathbf{M}_i$. The probability $\mathbb{P}_{\mathbf{M}}(S)$ of a sequence S from A^k being generated by $\mathbf{M}$ is $\mathbb{P}_{\mathbf{M}}(S) = \prod_{i=1}^{k} M_{i,S_i}$. $\mathbf{M}$ is in other words a product multinomial distribution over $\mathbb{A}^k$. The probability $\mathbb{P}_{\mathbf{M}}(S)$ score is often used as the match score. The NestedMICA suite motif scanning algorithm which I have used, provided in the program `nmscan` (Down and Hubbard, 2005), transforms the scores to bit scores and transforms them such that maximum score reported is 0 (Function 1.1).

$$W(S,p) = \prod_{i=1}^{|W|} W_i(S_{p+i-1}) \tag{1.1}$$

In brief, the PWM is a model for gapless position-specific probability distributions of nucleotides which assumes independence of nucleotide positions (Rahmann et al., 2003). Departures of the position independence assumption have

been reported in the form of variable length linkers, interdependencies between nucleotides at different binding site positions (Badis et al., 2009; Benos et al., 2002a; Bulyk et al., 2002), and compensatory mutations that maintain the binding energy and function of binding sites (Mustonen et al., 2008). More complex probabilistic motif models based on for instance Bayesian (Barash et al., 2003; Ben-Gal et al., 2005) and Markov networks (Sharon et al., 2008) have been developed to fit these observations. With the exception of the newest DNA–protein interaction assays which provide direct binding energy measurements of a protein with a large spectrum of different DNA binding sites (Berger et al., 2006; Maerkl and Quake, 2007b), parameter estimation of motif models more complex than the PWM is hard with often scarce biological data. The PWM therefore remains the model of choice for most large scale motif inference tools; it is intuitive to interpret as a sequence logo (Schneider and Stephens, 1990) and retains more of the information contained in binding site patterns than sequence word based models (Osada et al., 2004).

1.3 Computational methodology

Several lines of the work I describe in the later chapters builds on previously described computational frameworks, the most important of which I will summarise below. Firstly, Hidden Markov models are used for modelling sequential data (described in Section 1.3.1, applied in Chapter 2 for inferring motif family models). Secondly, the nested sampling Monte Carlo method used for drawing samples from complex probability distributions that are not analytically tractable (described in Section 1.3.2, applied in Chapters 2 and 3). Thirdly, random forest classification is applied in Chapters 4 and 5 for the supervised machine learning task of predicting TF domain labels for regulatory motifs (Section 1.3.4).

1.3.1 Hidden Markov Models in motif inference

A Hidden Markov Model (HMM) is a model for sequential signals. It is a stochastic finite automaton consisting of finite number of states. Each state has an associated probability distribution, and the distribution is typically multidimensional

(Dogruel, 2008). The HMM was originally developed and described in a series of papers by Baum *et al.* (Baum, 1972; Baum and Petrie, 1966; Baum et al., 1970; Baum and Eagon, 1967; Baum and Sell, 1968), and it quickly developed into a popular model in speech recognition (Baker, 1975). Applications to biological pattern recognition problems from data such as protein and DNA sequence arrived much later, sparked by several widely circulated papers from Haussler and others (Brown et al., 1993; Krogh et al., 1994). In these papers HMMs were described as a superset of the profile multiple alignment methods which were already commonly used in modelling protein sequence. Indeed, HMM profile based protein domain families computed with tools such as HMMER (Eddy, 1998) and stored in databases such as Pfam (Finn et al., 2010; Sonnhammer et al., 1997) and SUPERFAMILY (Wilson et al., 2009) are perhaps the most ubiquitous biological application of HMMs in computational biology, in addition to other common uses such as gene finding (Stanke and Waack, 2003). The HMM is also a commonly used formalism in regulatory motif inference problems. Firstly however let us arrive at a formal definition of an HMM and some of the common terminology used in connection to them.

For an observable sequence $O = O_1 O_2 \dots O_T$ emitted by HMM λ, each of its observables (symbols) is said to be emitted by a sequence of T hidden states from a finite set of N hidden states $S = S_1, S_2, \dots, S_N$. As described by Rabiner (1989), the model is parameterised by three types of parameters:

1) The transition probability distribution A_{ij} (Equation 1.2)

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], 1 \leq i, j \leq N \tag{1.2}$$

HMMs are often depicted as a diagram with directed, weighted edges showing transitions a_{ij} between nodes representing states. The missing edges between states correspond to transitions with probability 0 (see Figures 1.5 and 1.6).

2) The observable emission probability distribution $B = b_j(k)$ (Equation 1.3)

$$b_j(k) = P[v_k \ \ at \ \ t | q_t = S_j] 1 \leq j \leq N \ \ \cap \ \ 1 \leq k \leq M. \tag{1.3}$$

3) The initial state distribution $\pi = \pi_i$ (Equation 1.4)

$$\pi_i = P[q_1 = S_i], 1 \leq j \leq N and 1 \leq k \leq M. \tag{1.4}$$

A HMM can be used to solve several types of problems in relation to the observable sequence and the hidden state path, the three most common of which are:

1. Given a sequence of observations $O = O_1 O_2 \ldots O_T$ and a HMM $\lambda = (A, B, \pi)$, compute the probability of the observation sequence, given the model λ, that is, $P(O|\lambda)$. Computing $P(O|\lambda)$ involves integrating the possible state paths through the model with their likelihood (also known as the forward algorithm).

2. Given a sequence of observations $O = O_1 O_2 \ldots O_T$ and λ, how do we find the most likely hidden state path $Q = q_1 q_2 \ldots q_T$ (the 'Viterbi path') that generates ('explains') a sequence of observables. The algorithm that solves this problem is known as Viterbi decoding.

3. Adjusting λ parameters (A, B, π) such as to maximise $P(O|\lambda)$.

My work with the motif family model estimation problem has involved working on the first of the three above problems: defining a likelihood function over the sequence of nucleotide sequence motif columns and expressing it as an HMM forward algorithm. This work is described in more detail in Chapter 2, and its applications into motif inference and motif classification are described in Chapters 3 and 4.

Motif inference algorithms are also often expressed with an HMM model. The most common such sequence model, used for example in MEME (Bailey and Elkan, 1994), is the zero-or-one occurrences per sequence model, or ZOOPS (Figure 1.5). The common feature of the sequence models used in probabilistic motif inference algorithms is that they express biological sequence (e.g. DNA) as a string of symbols emitted by a series of emissions from a background model and a sequence motif. The background state generates the 'un-interesting' symbols

in the analysed sequence (the non-motif containing positions, which in most promoter analysis problems constitute the bulk of the sequence). The 'interesting' states are the overrepresented motifs, which are parameterised most commonly as a position weight matrix (PWMs described in Section 1.2.1).

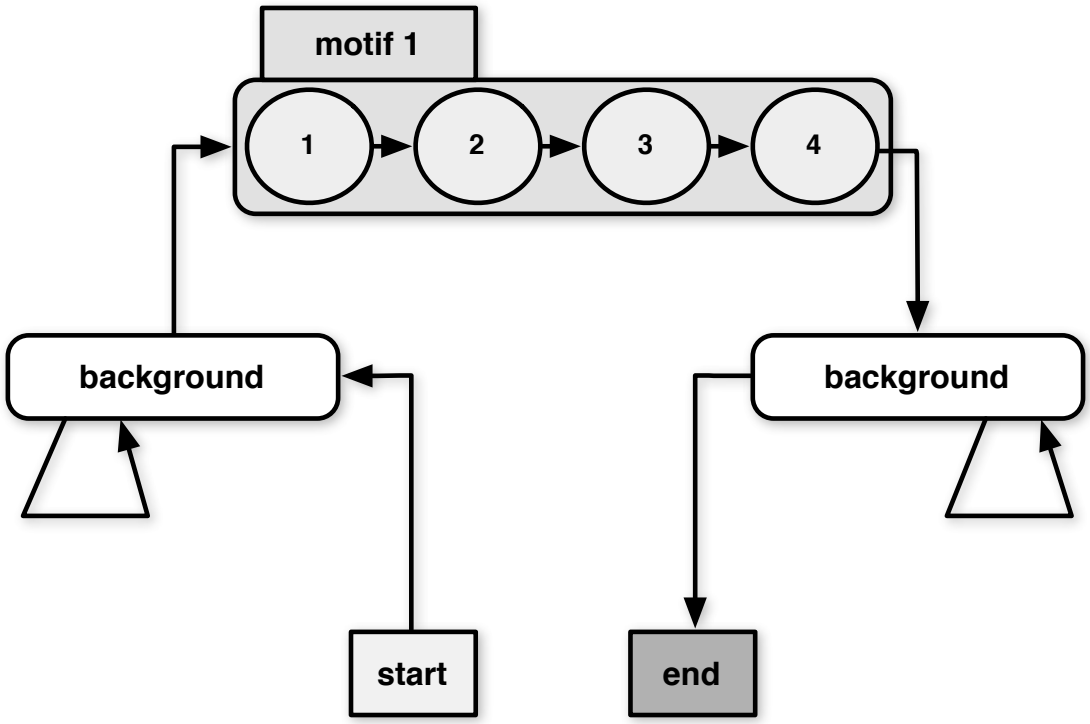

Figure 1.5: The zero-or-one occurrences per sequence–motif model (ZOOPS).

The sequence HMM used in the NestedMICA motif inference algorithm (Down and Hubbard, 2005) which I have also expanded as part of my project is slightly more complex, allowing multiple motifs to be modelled simultaneously. An example of these 'multiple-uncounted sequence-motif mixture models' (MUSMM) are shown in Figure 1.6.

The important improvement of the MUSMM model over the ZOOPS model is that it allows simultaneous motif learning from sequence data. In other words parameter estimation of each of the motifs is not done in iterations of learning a motif, masking its putative hit positions from the sequence, before repeating

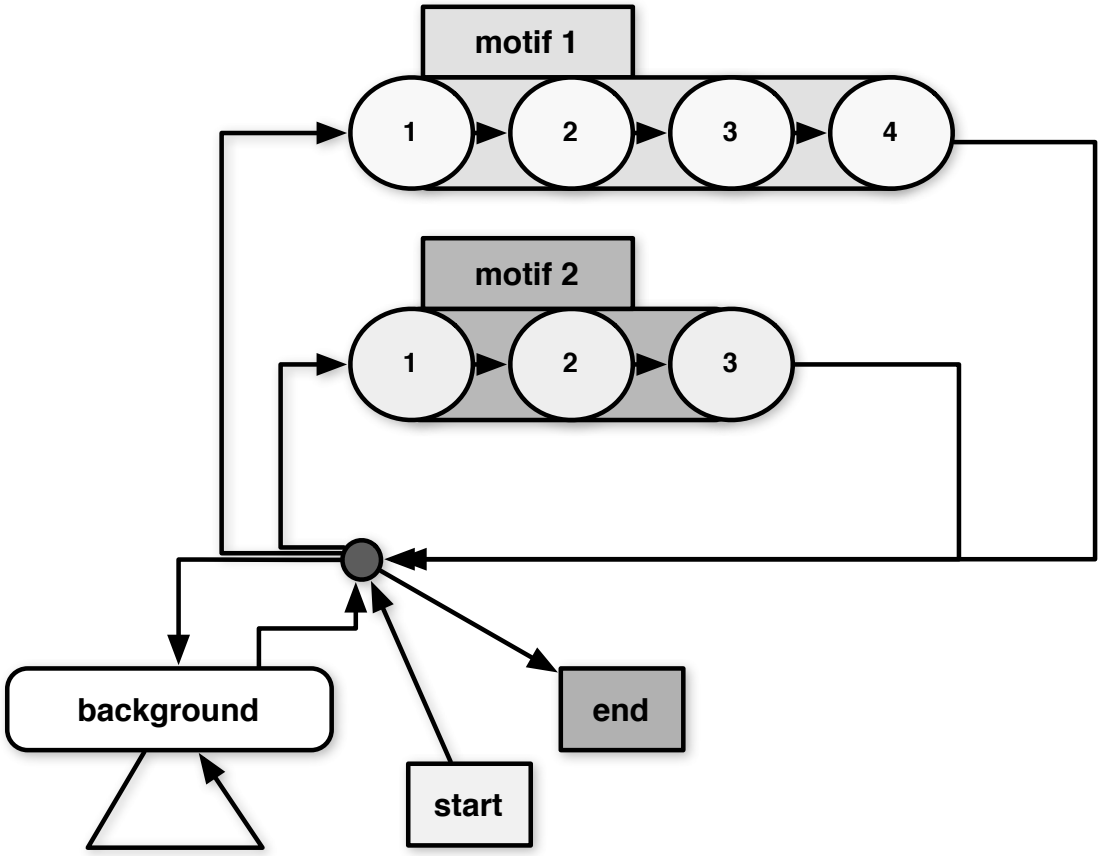

Figure 1.6: The multiple-uncounted sequence-motif mixture model (MUSMM).

the parameter estimation for the next, less strong motif. A greedy motif learning that requires repeated masking of sequence will incur an unpredictable sensitivity drop when multiple motifs are inferred: less and less sequence which is masked based on previously predicted is available for subsequent iterations. As I will show in Chapter 2, the metamotif inference framework I have developed also uses an analogous design to the NestedMICA algorithm to allow multiple metamotifs to be inferred simultaneously, with what I call the multiple-uncounted motif–metamotif mixture model, or MUMMM.

1.3.2 Nested sampling

Inference of parameters for Bayesian probabilistic models is often difficult, particularly for high dimensional models that are common in biology. Analytical solutions are almost always intractable. Most commonly approximate solutions are estimated using different Monte Carlo (MC) sampling techniques. I will below describe a state-of-the-art MC method, called nested sampling. Nested sampling is an MC technique originally introduced by Skilling (2004), and it is used in the metamotif inference algorithm I discuss in Chapter 2, as well as the NestedMICA motif inference algorithm which I expand in Chapter 3, and use for a large motif inference problem in Chapter 5.

As described by Dogruel et al. (2008), nested sampling is a MC method applied to an ensemble of e solutions (e typically ranges in hundreds to thousands). A nested sampler is firstly initialised with samples drawn from the prior distribution of states. After sampling, states are sorted by their likelihood and the member with lowest likelihood is removed from ensemble and replaced with a new sample, with the constraint that the new state has a higher likelihood than the removed state (Figure 1.7).

Samples are drawn from the prior distribution subject to the constraint that the likelihood of the new state must exceed that of the discarded state. This is done initially with rejection sampling (von Neumann, 1951), but after a certain number of iterations (the number of which is decided dynamically by measuring the rejection rate of the proposals), new samples begin to be generated with MCMC moves from other members of the ensemble because simple rejection sam-

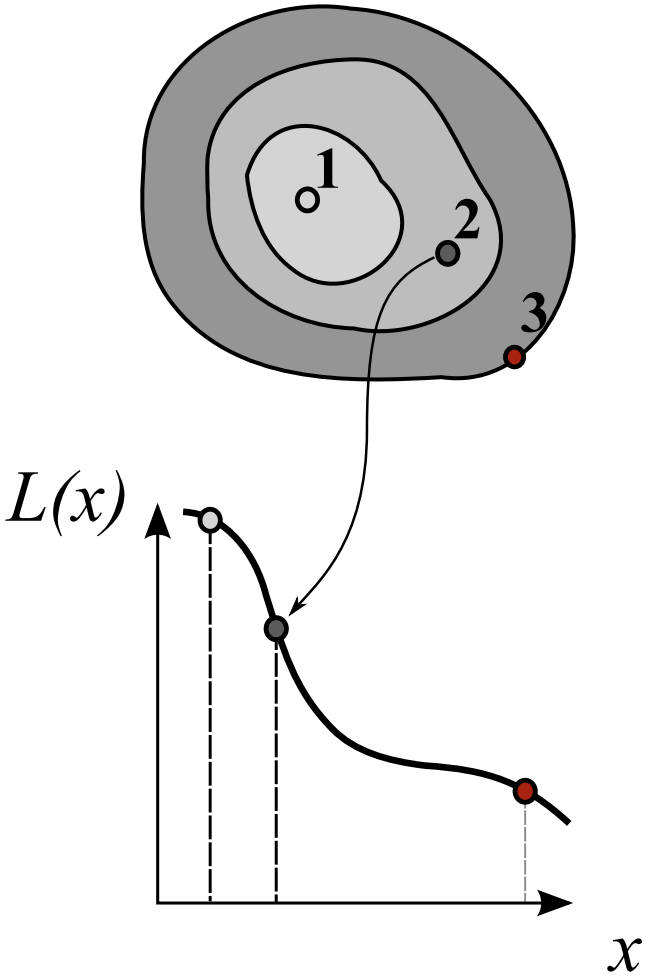

Figure 1.7: The likelihood contour. Lowest likelihood state is removed and a new state sampled on every iteration.

pling from prior with this increasing constraint becomes progressively 'harder' as the minimum likelihood threshold increases. As the sampling progresses through repeated iterations (typically in the range in tens to hundreds of thousands of iterations), more and more prior mass is excluded and the sampler reaches higher likelihood regions of the space. This is in a way analogous to simulated annealing, except progress occurs automatically without applying a temperature gradient to 'heat' or 'cool' the process (assuming that there are no complete plateaus in the space). Notably, nested sampling has demonstrated good performance in avoiding strictly local optima (Mukherjee and Parkinson, 2006; Shaw et al., 2007; Vegetti and Koopmans, 2009), unlike for instance Gibbs sampling which is a common MC strategy in motif inference. The fraction of prior mass removed from consideration at step t tends towards W_t (Equation 1.5).

$$W_t = \frac{1}{e}(\frac{e}{e+1})^t \tag{1.5}$$

A particular strength of the nested sampling technique is that it allows direct estimation of the Bayesian evidence of the model, something which Monte Carlo methods do not traditionally do. Assuming that the likelihood of states removed at step t is approximately equal at L_t, the Bayesian evidence Z of the model can be estimated as described in Equation 1.6.

$$Z = \Sigma_{t=1}^{\infty} W_t L_t \tag{1.6}$$

The estimate of Z becomes progressively more accurate as sampling progresses, and indeed Z can be used for comparing models (motif set models derived with different input parameters for instance can be assessed by their Bayesian evidence). Furthermore, change in the evidence estimate Z_t (evidence at step t) is the criterion used for terminating the sampling (Equation 1.7). This same criterion is used with the DNA, protein and metamotif samplers in the NestedMICA suite (Dogruel et al., 2008; Down and Hubbard, 2005; Piipari et al., 2010a).

$$\frac{1}{Z_t} L_t (\frac{e}{e+1})^t < 0.01 \tag{1.7}$$

1.3.3 The NestedMICA algorithm

NestedMICA applies nested sampling to motif inference, using an independent component analysis (Comon, 1994) like formulation of the motif inference problem: input sequences are modelled as a mixture of a number of independent motif signals and random noise (the background model). As described by Down and Hubbard (2005), in linear ICA, a matrix of observations X is approximated as a linear mixture A of some sources s and a noise matrix ν:

$$x = As + \nu \tag{1.8}$$

The noise matrix ν represents errors in the linear approximation. A commonly described example application of ICA is the "cocktail party problem": a set of M microphones record different mixtures of the voices of N speakers. Given samples from these microphones at t time points, ICA methods attempt to factorize the $M \times t$ observation matrix into an $N \times t$ source matrix and an $M \times N$ mixing matrix. One can map the motif inference problem to an independent component analysis like formulation where the observations are a series of nucleotide strings, the sources are short sequence motifs, and a sequence background model represents the random noise. The mixing operation in motif ICA however is not simply a matrix multiplication.

The simplest mixing operation, and the one used by default, is simply a binary weighting: a motif has either a zero or 'full' weight in contributing to the likelihood of a sequence. That means that the mixing matrix (depicted in Figure 1.8) informs for each motif and sequence pair if a motif is expected to be a match in the sequence, according to a MUSMM-like sequence mixture model (Figure 1.6, where there are two motifs in the sequence with a nonzero weight). More complex mixing matrices, such as logistic function based weighting, are also included in the NestedMICA suite.

The model parameters – the motifs and the mixing matrix which describes pairing of motifs to sequences – are estimated with the nested sampling strategy (Section 1.3.2). Nested sampling allows inference to be made without heuristics to provide local starting points for motif search. Similarly, repeated runs of the algorithm are unnecessary, unlike with the commonly used Gibbs sampling Smith

(1987) based motif inference algorithms pioneered by Lawrence et al. (1993), or greedy expectation maximization (Dempster et al., 1977) based algorithms such as MEME (Bailey and Elkan, 1995; Bailey et al., 2006). A schematic of the motif ICA and nested sampling, is provided in Figure 1.8.

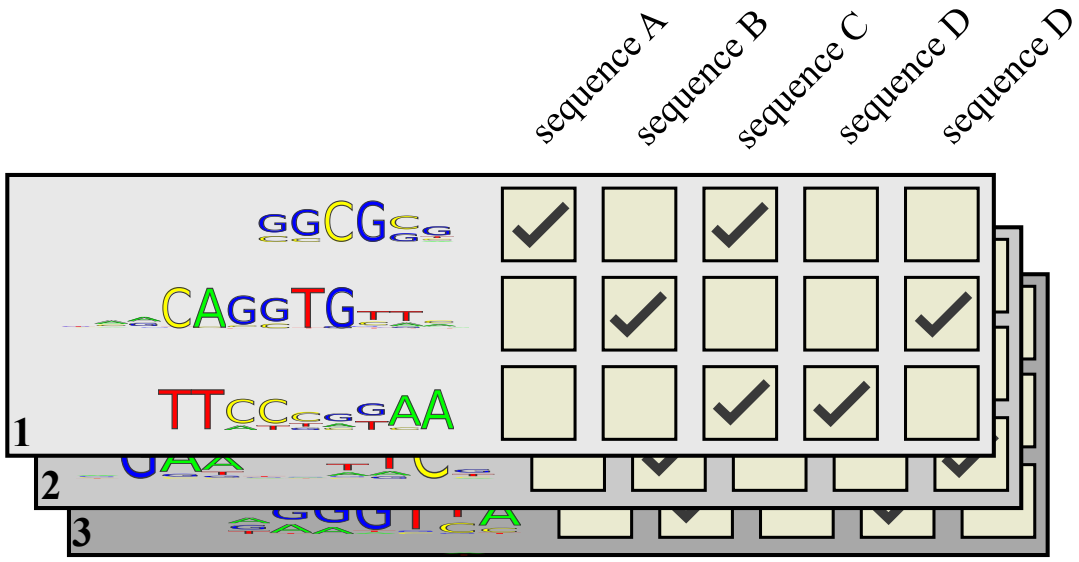

Figure 1.8: The NestedMICA model components: the motif set and the mixing matrix. An ensemble of three states is shown (states labelled 1,2,3).

A realistic model of the genomic sequence is also a key consideration with motif inference algorithms. The sequence background model in these algorithms is commonly modelled with a stationary Markov chain, and therefore depending on the order of the Markov chain it is parameterised simply by the nucleotide, dinucleotide, ... frequencies of the sequence. Real promoter sequence however is not uniform, and instead contains, for instance, discrete regions of GC-richness and AT-richness. NestedMICA uses a sequence background model that allows for compositionally distinct regions, for example the variation in GC content that is known to occur on multiple scales (FitzGerald et al., 2006; Thompson and Rouchka, 2003). In addition to simply varying GC content, dinucleotide content can also be used to subdivide promoters according to their CpG content to two groups: those with exceptionally high frequency of CpG dinucleotide content,

and those with average genomic CpG content (Saxonov et al., 2006). Other regional biases in di-, tri- and tetranucleotides have also been described (Burge et al., 1992). The NestedMICA background model is referred to as mosaic to highlight its capability to describe sequence as a mixture of multiple generative processes (Markov chains). Use of multiple Markov chains, or 'classes', that are weighted per sequence position, improves the capacity of the background to describe compositional biases and is a considerably less complex model than higher order Markov chain backgrounds which are commonly used in motif inference algorithms.

A recently published motif inference algorithm BayesMD, which similarly as NestedMICA applies a Monte Carlo sampling method that is resilient to local maxima (parallel tempering, Gregory (2005)), and a sequence background model related to NestedMICA but trained from a larger selection of noncoding sequences, improves sensitivity over MEME, Align-ACE, MDScan, and also against NestedMICA in most benchmarks (Tang et al., 2008).

NestedMICA has been implemented in the Java programming language in a modular fashion where the definition of the model and the nested sampling framework are separate. As I will show in Chapter 2, this has made it possible to replace the NestedMICA motif model (the PWM) with a different space of models, and to therefore allow applying the nested sampling algorithm in the space of motif family models I have termed 'metamotifs'. Furthermore, using existing nested sampling framework has also had the benefit of high runtime performance and scalability: the original NestedMICA algorithm and my variants of it make use of multiple CPUs when available, and the computational load can be distributed over multiple computers, scaling to up to 40 CPU cores (unpublished data).

1.3.4 Random forest classification

Supervised machine learning techniques aim to build a function based on input training data to predict the state of a response variable (the output). The response can be either continuous, at which case the procedure is called regression, or discrete, at which case the procedure is called classification. The function

should fit the training closely, but it should also generalise to other unseen data (Bhaskar et al., 2006). A compromise therefore needs to be made between a function which memorises the feature value combinations from training data but is incapable of generalising it to new input (an effect often referred to as 'over-fitting'), and one which generalises but is not necessarily able to fit all the training examples (training error). In Chapter 4 I use a supervised machine learning technique called random forest classification to learn the mapping from a motif (PWM) to the likely DNA binding domain which binds it.

A random forest is an ensemble machine learning technique, meaning that the classification function itself is a function of a number of independent classifier functions. The technique can be applied to either regression or classification, but we will concentrate on random forest classification, as regression techniques were not used in this work. According to Breiman (2001b), random forests follow in the line of three types of ensemble classification techniques noted below, all acting on ensembles of classification trees. Any of the three methods noted below are also sometimes confusingly referred to as a type of random forest.

1. "Random subspace" methods, where randomness is applied to subsets of features to use to grow trees (Ho, 1998).

2. Bagging methods, where randomness is applied to the choice of training data examples used to grow classification trees (Breiman, 1996).

3. A method where the splits made at tree nodes are made randomly according to voting (Dietterich, 1998).

The common factor between all of the above methods is that for the k^{th} classification tree, a random vector θ_k is generated independent of past vectors $\theta_1, \ldots, \theta_{k-1}$ but with the same distribution (i.i.d.); A tree is grown using the training set (or its subset) and θ_k, resulting in a classifier $h(\mathbf{x}, \theta_k)$ where $\mathbf{x}$ is an input vector. The nature of θ varies between the different tree construction methods. For instance, in bagging it can thought to be generated as the counts in N boxes resulting from N darts thrown at random at the boxes, where N is number of examples in the training set.

In Breiman's random forest, each θ_k is trained from random selection of features from a subset x_k of bootstrapped examples in x (Equation 1.9) (Breiman, 2001b). Each x_k are taken from roughly two thirds of the examples, and the rest are used for the so-called out-of-bag error estimates (see below).

$$\{h(\mathbf{x}, \theta_k), k = 1, ...\} \tag{1.9}$$

The set of i.i.d. random vectors noted above are noted as θ_k. In a classification problem, a random forest is a collection of decision tree predictors, and the response value is simply chosen by popular vote for the most popular label from the ensemble of k trees (the ensemble is referred to as a 'forest'). The relative frequency at which the winning vote was made in the ensemble gives a confidence estimate for the decision. In regression the response value is the average of the response values in the forest.

A random forest classification has a number of attractive properties as a generic supervised machine learning framework:

1. An unbiased generalisation error estimate is made without the need for separate cross validation. This is achieved by leaving approximately one third of the training data x out from the bootstrapped examples x_k and they are labelled with the k^{th} classification tree. The error rate of this classification is the out-of-bag (oob) prediction error rate.

2. Its generalisation error tends to perform comparably to SVMs (Meyer et al., 2003) and favourably to related ensemble methods such as Adaboost (Freund and Schapire, 1996) or bagging.

3. It is naturally suited for multiclass problems (such as the motif domain labelling problem in Chapter 4), and provides a confidence estimate for the classification decisions regardless of the dimensionality of the class variable.

4. It is simple to understand, and provides insight into the importance of different classifier features (several kinds of proximity measures of training examples can also be computed). This is in contrast with kernel methods whereby variable importances are not straightforward to derive or interpret when one needs to resort to nonlinear kernel functions (usually for

improved classification performance), or multiclass classification. The variable importance measure in Breiman's `randomForest` package (Liaw and Wiener, 2002) which I use in my project is based on permutation testing: for each tree, all values of the m^{th} predictor variable are permuted, classification is made, and internal error rate computed as normally. The difference between correct (unpermuted) and incorrect (permuted) classifications is then computed and averaged over all trees, and normalised by the standard error. The margin is defined as the proportion of votes for true class minus maximum proportion of each of the other classes.

5. Although several adjustable input parameters are made available, only one is generally adjusted (`mtry`, which denotes the number of variables randomly sampled as candidates at each split), values of which the classification is also often robust to (Breiman, 2001a; Liaw and Wiener, 2002). This is in contrast with kernel method based classification, where a grid search of kernel function parameters is always necessary.

1.4 Biological datasets and resources

The most important biological datasets and resources which I have made use of during the course of my project are introduced below. Additional resources used in individual analyses are described in later chapters.

1.4.1 Ensembl

Ensembl is an open access database which provides access to eukaryotic genome sequence and annotation (Birney et al., 2004; Hubbard et al., 2009). Originally developed for analysis of the human genome, the current release 58 now contains 49 annotated eukaryotic genomes. The genome annotations provided by Ensembl are a key resource for large scale regulatory motif inference studies. For instance, all promoter sequences used for predicting motifs in my project have been chosen dependant on the transcription start site predictions provided by Ensembl. The resulting promoter regions are therefore a result of a combination of computational predictions and manual curation. Similarly, masking protein coding

sequences and sequence repeats is made possible by annotations retrieved from Ensembl.

In addition to the web site at http://www.ensembl.org, Ensembl offers programmatic access with a publicly supported Perl API (Stabenau et al., 2004). Several other language specific APIs unsupported by the Ensembl project have also surfaced, including Ensembl Core for Ruby [1] and biojava-ensembl [2]. Both of the above unsupported libraries proved useful in my work, and in the course of my project I in fact developed simple Ensembl database backed tools on top of biojava-ensembl for regulatory motif inference oriented tasks, which in turn were used in all of the peer-reviewed, published work which I have taken part in (Lewis et al., 2009; Piipari et al., 2011, 2010a,b), and Murray *et al.* (in press). See Section 5.2.1 and Appendices A, B for more detail.

1.4.2 Regulatory motif databases

Experimentally determined regulatory motifs have been another key resource in my work, both with motif family classification (Chapter 4) and validation of *de novo* inferred motifs (Chapter 5). The different TFBS motif databases I have resorted to in my work, and the rationale for choosing the individual datasets for analyses, are summarised below.

The regulatory genomics community suffers at the moment from the absence of a single authoritative database, data format, or minimal publishable requirements for distributing experimentally validated regulatory motifs or associated metadata (e.g. species information, experimental method). This is in notable contrast to for instance sequence, protein structure, or gene expression microarray data, each data type of which is generally required to be made available in a public database upon publication in a peer reviewed journal. TFBS motif data is scattered between individual publications, several databases in different partially overlapping subsets, and the standard of data and curation quality varies.

[1] http://www.github.com/jandot/ensembl
[2] http://www.derkholm.net/svn/repos/biojava-ensembl

1.4.2.1 TRANSFAC

Currently the largest single dataset of eukaryotic TFBS motifs is contained in the TRANSFAC database, which is a commercial, curated database of eukaryotic gene regulation maintained by BIOBASE Ltd (Matys et al., 2006; Wingender et al., 2001). TRANSFAC contains a curated set of TFs, known TF–target gene regulatory relationships, and TFBS motifs as position frequency matrices (PFM). Most of the TFBS data stored in TRANSFAC originates from individual small-scale studies, including electrophoretic mobility shift assays (Fried and Crothers, 1981b; Garner and Revzin, 1981), DNase I foot-printing (Brenowitz et al., 1986), immunoprecipitation (Hecht and Grunstein, 1999) and some from higher throughput approaches such as *in vitro* selection (SELEX) (Oliphant et al., 1989). The more recently released TRANSFAC versions have begun expanding the database with ChIP-seq and various other HT methodologies discussed in Section 1.1.2.

TRANSFAC also defines its own structural taxonomy for classifying TF motifs by the structural class and family of binding TF. The structural taxonomy is largely similar on the level of TF domain families to the taxonomy used in the JASPAR database (Section 1.4.2.2), but the coarser level of the hierarchy ('superfamilies' in the TRANSFAC terminology, 'structural classes' in the JASPAR terminology) differs both in the divisions of TF domains and the terminology used.

The species covered by TRANSFAC are primarily vertebrates. Other animal TFs, as well as some plant and fungal TFs are included but in smaller scale. For instance, the database release 12.2, which my analysis in Chapter 4 is based on, contains a mere 38 motifs annotated with the species *S. cerevisiae*, and the same number of motifs for *Arabidopsis thaliana*, 68 for *D. melanogaster*, but 409 for mouse and 455 annotated with *H. sapiens*.

Due to the license fee associated with TRANSFAC, and its closed nature, an open access alternative to TRANSFAC could be beneficial for the research community. Attempts have been made to create alternatives, the most interesting being perhaps ORegAnno (Griffith et al., 2008), which is a community curation based database of transcriptional regulation. The ORegAnno dataset however has unfortunately not progressed to a form that is usable for most researchers.

The JASPAR database (Section 1.4.2.2), which similarly to TRANSFAC relies on a dedicated team of curators, has perhaps the best potential in providing an alternative to TRANSFAC's collection of TFBS motifs.

I made use of TRANSFAC motifs for the motif domain family classification analysis conducted in Chapter 4 primarily because it allowed a direct comparison to previous motif classification methods MotifPrototyper (Xing and Karp, 2004) and SMLR (Narlikar and Hartemink, 2006), and because at the time it contained a considerably larger training and cross-validation dataset than the open-access alternative JASPAR: TRANSFAC 12.2 contained 848 structurally classified motifs (Wingender, 2008) versus a total of 138 in JASPAR 2008 (Bryne et al., 2008). In Chapter 5 I however describe more recent work where I built a motif family classifier based on the most recent JASPAR release, which has been expanded to include for instance many of the high-throughput datasets noted in Section 1.1.2.

1.4.2.2 JASPAR

JASPAR is another commonly used database of TFBS motifs (Bryne et al., 2008; Portales-Casamar et al., 2010; Sandelin et al., 2004). JASPAR distinguishes itself from TRANSFAC in several important aspects:

1. The structural terminology of TF domains, which covers most of its motifs, differs from that of TRANSFAC. JASPAR uses a two-level DNA binding structural mode taxonomy introduced by Luscombe et al. (2000). This classification terminology extends an earlier taxonomy created by Harrison (1991) on a smaller number of crystal structures. The Luscombe et al. (2000) taxonomy describes 'classes' and 'families' for TFs. Classes are defined by a manual, visual comparison of structures and families by a computational clustering of the domain structures with the SSAP secondary structure alignment algorithm (Orengo and Taylor, 1996). The taxonomy in TRANSFAC extends to more detailed levels, but past the class and family-like levels appears to be defined on a rather *ad hoc* basis by the TRANSFAC curators based on the terminology introduced in literature.

2. The data is open access, and its curation is of high quality. Key annotations such as species, experimental method and primary publications which

describe the data in the database are included almost with no exceptions, unlike TRANSFAC, where for instance only 490 of the 848 records contain a reference to a peer reviewed publication.

3. JASPAR, unlike TRANSFAC, is a non-redundant database, and aims to cover different kingdoms of life with separate non-redundant datasets (currently for mammals, insects, fungi and plants are covered). This is an important effort because of the lineage specific expansion of TF domains (Wilson et al., 2008a): TF domains utilised preferentially by different kingdoms of life differ substantially (discussed in 1.1.2).

4. JASPAR 2010 contains a near to complete non-redundant motif dataset of 177 *S. cerevisiae* motifs, compared to only 38 *S. cerevisiae* motifs in TRANSFAC 12.2 which emphasises vertebrate genomes.

I used the JASPAR database in Chapter 5 to train a motif family classifier to assess computationally inferred *S. cerevisiae* motifs most importantly because of the last two points above; for an accurate organism specific classifier it is important to have a good coverage of the TF domains that are specific to the lineage being studied. For instance, there are 47 known TFs with the fungal specific zinc cluster domain (Macpherson et al., 2006) in the *S. cerevisiae* genome out of the total 99 *S. cerevisiae* zinc finger motifs. TRANSFAC 12.2 includes motifs for only 9 of them, whereas JASPAR 2010 contains 38.

1.4.2.3 UniPROBE

UniPROBE (Universal PBM Resource for Oligonucleotide Binding Evaluation) is, as the name suggests, a database containing protein binding microarray derived motifs. At the time of writing, the database included motifs for 391 proteins from eight different studies, originating from affinity tagged TFs from human (Berger et al., 2006; Scharer et al., 2009), mouse (Badis et al., 2009; Berger et al., 2008), *C. elegans* (Grove et al., 2009), budding yeast (Zhu et al., 2009), the parasites *Malaria falciparum* and *Cryptosporidium parvum* (Silva et al., 2008), as well as the Gram-negative bacterium *Vibrio harveyi* (Pompeani et al., 2008). Its focus is simply to provide a repository for downloading and searching raw PBM data,

and PWM models derived from the data with the Seed-and-wobble algorithm (Berger et al., 2006). It does not attempt to provide a rich annotated reference database of TFBS motifs, like JASPAR or TRANSFAC. I have used two motif datasets from the UniPROBE database:

1. The 168 mouse homeodomain TF motifs by Berger et al. (2008). This dataset is one of the two high-throughput studies published in 2008 of the developmentally important homeodomain TFs, in addition to the bacterial one-hybrid dataset of *D. melanogaster* homeodomain TFs (Noyes et al., 2008a). The Berger et al. (2008) dataset covers 65% of the 260 known homeodomain proteins in the mouse genome. I apply both of the above mentioned homeodomain datasets in Chapter 4 for evaluating the capacity of the **metamatti** classifier in distinguishing homeodomain motifs from members of five other common TF domain families.

2. The 89 *S. cerevisiae* TF motifs (Zhu et al., 2009). This study provides the largest protein–DNA interaction dataset recovered with a single methodology, and it is therefore a convenient comparison dataset for comparing *ab initio* predicted regulatory motifs with. The slightly larger study by Berger et al. (2008) covers 112 yeast TFs, with a combination of different high-throughput methods.

1.5 Contributions of this thesis

My goal in this dissertation is to, firstly, present a new probabilistic model for familial relationships between regulatory motifs (Chapter 2). I then apply this familial motif model to sensitively infer motifs from novel sequence (Chapter 3), and to predict the DNA binding domain responsible for binding different regulatory motifs (Chapter 4).

Finally, I conduct a *de novo* motif inference study of the budding yeast genome to infer a large regulatory motif set from its promoters with a number of commonly used motif inference tools (Chapter 5). This is done primarily to assess the ability of the different motif inference tools to discover motifs that are consistent with previously known motifs from this particularly well studied eukaryotic regulatory genome.