Chapter 1

Introduction

1.1 Life, chemistry and computers

Past several centuries of research in the field of life sciences has revealed that all life can be described as nothing but an interplay of chemical reactions involving chemical compounds called *proteins* and *nucleic acids*. Proteins are responsible for regulating important biochemical processes like those involved in metabolism or immune response. They also form a major constituent of essential biological components like blood, cytoskeleton, hair, etc. In essence, they are responsible for life itself! Nucleic acids are the basic chemical compounds that encode the information necessary to produce proteins. Long chained molecules of nucleotides form the *genome* of an organism and are responsible for the observed complexity. Studying the function and structure of proteins and nucleic acids helps in elucidating their biological relevance. Research devoted to extracting **biological** features through a **computational** analysis of such long sequences of **chemical** compounds forms the core of *computational biology*.

1.1.1 Biological preliminaries

In eukaryotes, the genetic information is encoded as a double-helix **D**eoxyribo **N**ucleic **A**cid (DNA) consisting of *nucleotides*. Each nucleotide is composed of a sugar molecule, a phosphate molecule and a *base*, which is one of adenine (A), cytosine (C), guanine (G) or thymine (T). Hence, a DNA molecule can be spoken of in terms of the constituting bases or nucleotides. The double stranded structure arises from the base-pairing of A with T and C with G. The DNA in a cell is packaged into a nucleoprotein complex known as *chromatin*. The fundamental repeat unit of the chromatin is the *nucleosome* which is generally comprised of 146 base pairs of DNA wrapped around an octamer of proteins known as *histones*.

Through a process known as *transcription*, certain stretches of the DNA known as *genes* are transcribed into *messenger* **R**ibo**N**ucleic **A**cid (mRNA). After a pre-processing step in which parts of the sequence (*introns*) are removed via *splicing*, the reduced mRNA consisting of coding regions known as *exons* leaves the nucleus. Finally, mRNA molecules are *translated* into proteins by cellular structures called *ribosomes*. This process of **DNA→RNA→Proteins** is referred to as the *central dogma of molecular biology*.

Although the genetic information is the same in all cells in an organism, not all cells have the same subset of active (*expressed*) genes. The mechanisms responsible for the *regulation*

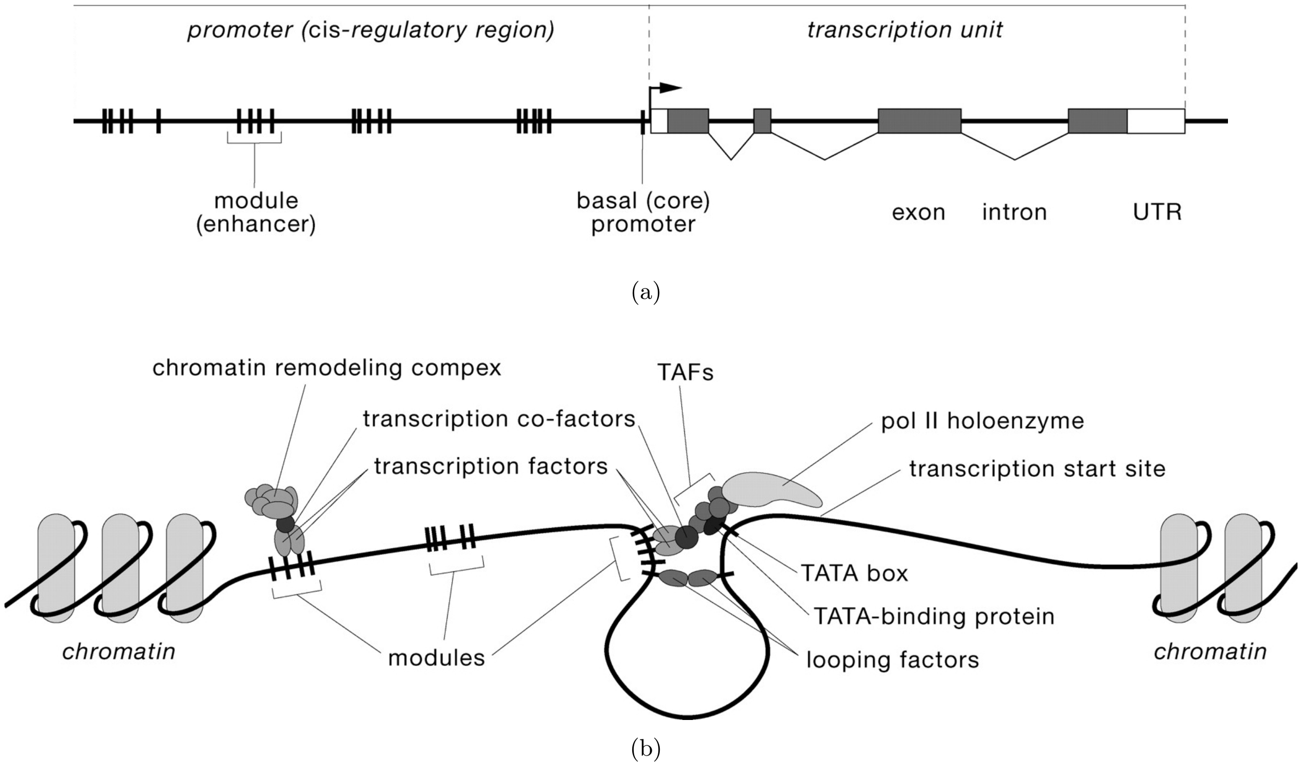

Figure 1.1: *Organization of a generalized eukaryotic gene.* **a)** The transcription unit consists of *exons* (blocks) – regions which are finally transcribed to mRNA, *introns* (bent lines) – non-coding sections of a gene that are spliced out after transcription and *untranslated region* (unshaded regions of the blocks) – portion of the mRNA that is not translated into proteins. Cis-regulatory regions of a gene consist of a basal (core) promoter region, an upstream promoter region which depending on the distance from the transcription start site may be referred to as *enhancers*. The core promoter contains binding sites of several transcription factors like RNA polymerase II, TATA-binding protein (TBP), TBP-associated factors (TAFs, also known as *general* TFs), etc. Clusters of multiple binding sites form cis-regulatory *modules*. **b)** Several factors combine together for the initiation of transcription yielding the transcription preinitiation complex (PIC) and the chromatin remodelling complexes. For a recent review on the different elements of regulation, see Maston *et al.* [122]. Figure adapted from [218].

of gene expression – that is, determining which genes are to be expressed at a given time and in response to a particular stimulus – are central to the process of cell specialization. While each step in the gene to protein path can be regulated, of particular interest to us is the regulation of transcription *initiation.*

The initiation of transcription requires information that is contained in *cis*-regulatory regions in the genome. That is, regions located on the same strand that are close to the gene being transcribed. Those lying close to the transcription start site are usually referred to as *promoters*, while those that typically lie upto a hundred kilobases away are known as *enhancers* (Figure 1.1) [208, 26]. Regulatory proteins known as *transcription factors* (TFs) bind to short ($5 - 20$ base pairs) degenerate motifs in these regions and thereby modulate the rate of transcription initiation. The successful binding of TFs relies on an open chromatin structure which facilitates accessibility of TFs to their corresponding DNA binding segments [217]. Different factors work in a combinatorial fashion to facilitate this chromatin modification through the formation of chromatin remodelling complexes (Figure 1.1 **b**). Finally, the spatial and temporal regulation of gene expression is mediated through the combinatorial action of multiple different transcription factors. Identification of the *binding sites* (BSs) of transcription factors is essential to the characterization of the functional elements of a genome. This dissertation fits in the niche of computational methods that aim to identify and predict such binding sites.

The knowledge that functional sequences tend to be conserved across different species has led to an increased effort in genome sequencing. At present, genomic builds of more than 30 different species – ranging from yeast and hedgehog to cow and mouse – have been made available. Using cross-species comparisons to extract conserved sequence segments has become the norm for filtering putative regulatory elements. Such phylogenetic comparisons that reveal evolutionarily conserved functional elements are described by the term *phylogenetic footprinting* (reviewed in [198]).

Current computational methods that use phylogenetic footprinting to predict conserved binding sites adopt a multi-step approach – individual sequences are searched for putative transcription factor binding sites, sequence comparisons performed to extract conserved regions and finally the single sequence predictions mapped to identify putative conserved binding sites. It is the goal of this work to present an algorithm and statistical framework that does the same in a novel *simultaneous* manner. Additionally, evolutionary characteristics of regulatory sites differ significantly from those of the surrounding regions. Through the proposed simultaneous framework, we also demonstrate how site specific evolutionary behaviour can be considered while annotating for conserved binding sites.

The rest of the Chapter is structured as follows. Beginning with a concise discourse on experimental methods for identifying TFBSs in the next section, we proceed to a discussion on the representation and modelling of binding sites in Section 1.2.2. This is followed by an overview of the computational methods for detecting TFBSs which do not employ phylogenetic footprinting in Section 1.3. To discuss those that do, a brief detour into the field of alignments is taken in Section 1.4. Here, we focus on methods that are most relevant to this work with the theoretical details presented later in Chapter 2. Following this, Section 1.5 presents the methods that use phylogenetic footprinting. Section 1.6 motivates the problem and presents a brief outline of the proposed approach. The Chapter ends with the outline of the thesis.

1.2 Transcription factor binding sites

As noted earlier, transcription factors bind short, degenerate motifs on the DNA. Experimentally found instances of the binding sites of a factor can be studied to extract common features shared by the sites. Usually, some positions are well-conserved across all sites while some tolerate more variability in the corresponding nucleotides. This variability allows a more flexible regulatory control over transcription. On the other hand, it results in the protein being capable of binding other non-functional motifs. Searching for such short, degenerate motifs in the genome, is a challenging task, both experimentally and computationally.

1.2.1 Overview of experimental approaches

Perhaps the most concrete means to formulating and verifying biological hypotheses is through wet-lab experiments. Annotating function to regulatory elements, establishing identity of involved proteins, studying the interplay of different constituents and deciphering the regulatory logic – all require experimental strategies for yielding biologically reasonable results. In the following, we sketch the major schools of experimental strategies; for more details see [56, 122].

A class of procedures takes advantage of the changes in the chromatin structure (presence/absence of nucleosomes) to identify putative regulatory regions. Such changes can be detected as increased sensitivity of the nucleosome-free DNA to digestion with the enzyme DNaseI. This increased sensitivity is referred to as *DNaseI hypersensitivity* [71, 114]. By finding such regions, one can surmise about their regulatory relevance. Approaches building upon this basic idea include, amongst others, the works of Vettese-Dadey *et al.* [201], McArthur *et al.* [125] and Crawford *et al.* [45]. However, the presence of DNaseI hypersensitivity only implies that the DNA segment is transcriptionally active but does not prove its functional relevance.

To verify the functionality of a transcriptional regulatory element, functional assays using *reporter genes* provide the most reliable approach. Reporter genes induce visually identifiable characteristics when expressed (eg. green fluorescent protein (GFP) glows green under UV or luciferase produces light). By placing the test DNA sequence upstream of a reporter gene, and introducing the resultant construct into a cell (*transfection*), changes in expression of the reporter gene can be measured. This helps in determining if the test DNA segment contains elements that alter reporter gene expression. The precise configuration of the reporter construct depends on the regulatory element to be identified and designing one is usually a non-trivial task. Directed studies as well as large-scale approaches in this area include the works of Strauss *et al.* [190], Siemen *et al.* [179], Hallikas *et al.* [73] and Muller *et al.* [137].

Another class of methods uses electrophoretic mobility shift assays [67, 92]. The underlying idea here is to sieve the protein-bound from the unbound DNA using a gel, the rate of moving through which depends on the size and charge of the molecule. The bound DNA being heavier and larger would migrate slower than the unbound DNA. The ratio of the bound to unbound DNA can then be used to calculate the affinity of a protein to the test DNA sequence.

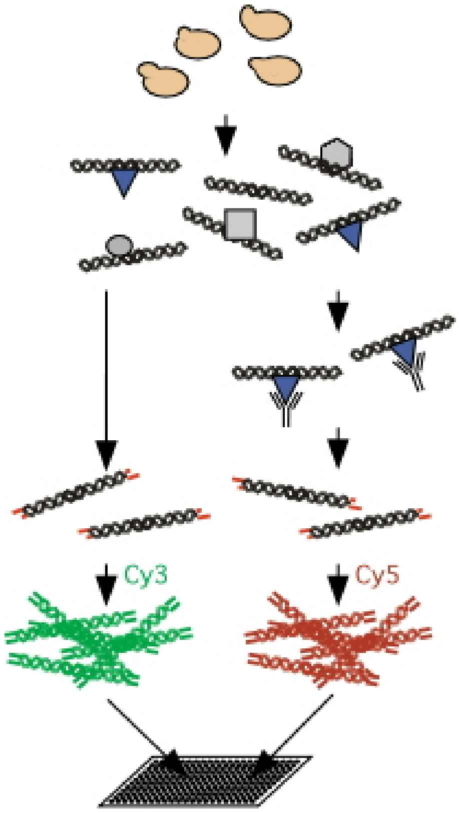

Figure 1.2: *Schematic summary of the ChIP-chip method:* Using formaldehyde, proteins are cross-linked to DNA *in vivo*. Formaldehyde is used because the crosslinks are heat-reversible which allows downstream enzymatic treatment of the DNA. After crosslinking, the DNA is sheared to smaller fragments, usually ranging from $0.2 - 1$ Kb. The DNA fragments crosslinked to the protein of interest are then enriched by immunoprecipitating with a protein-specific antibody (right). The formaldehyde crosslinks are then reversed and the DNA purified. Due to low DNA yields from immunoprecipitation, usually the fragments are amplified. Finally, the enriched DNA is labeled with a fluorescent molecule like Cy5 while the reference genomic DNA is labeled with another molecule Cy3. Both the probes are then combined and hybridized to a single DNA microarray. Ideally, to provide a comprehensive and unbiased survey of protein-DNA interactions, the microarrays used in ChIP experiments contain elements that represent the entire genome. Figure and text adapted from [1] and [34].

The technique of *DNase footprinting* [66] allows one to compare the cleavage pattern of isolated DNA against that of the DNA in the presence of proteins. If the protein binds the DNA, the corresponding stretch is protected against DNaseI cleavage and therefore fewer cleavage sites are found. In combination with the gel-shift assays described above, the protected sites can be separated from the cleaved sites. This method allows the determination of the precise location of the protein binding sites. A high-throughput approach based on the combination of gel-shift assays and DNaseI footprinting is SELEX [197, 64], also known as *in vitro selection*, where a large pool of random DNA fragments is tested against a protein. A drawback of these approaches is that often unintended DNA-protein interactions are also detected.

For a known protein, *chromatin-immunoprecipitation* (ChIP) provides a powerful *in vivo* strategy to determine its target locations. Using formaldehyde, the proteins are crosslinked to the DNA, which is then fragmented into 100-500 bp long pieces. A protein-specific antibody coupled to a retrievable tag, is used to pull down (*precipitate*) the DNA-protein complex from the pool of DNA fragments. Finally, the associated DNA is recovered, sequenced and analyzed – either through amplification or through the use of DNA *microarrays*.

In a DNA microrarray, *probe* sequences of known DNA molecules are placed on an array of inert substrate thus forming a collection of microscopic spots. By measuring the hybridization levels of *target* sequences, one can determine their enrichment under different conditions or locations. Using the DNA purified by ChIP, the precise location of the binding regions on the sequence can be identified. This technique known as *ChIP-on-chip* provides an efficient and scalable way for the identification of binding sites of DNA-binding proteins (Figure 1.2). Recently, through genome-wide analyses one can determine the binding sites of a protein throughout the genome. Examples include articles by Ren *et al.* [161], Horak *et al.* [84], etc. More high-resolution and high-coverage methods [145, 27] have also been proposed; for reviews of ChIP-chip advances see [34, 219] . Although an expensive method, ChIP-chip is gaining popularity due to its ability to identify the TFBSs in an unbiased manner. However, the dependence on a highly TF-specific antibody is usually a major hurdle in performing ChIP-chip experiments.

Summary The conclusions that can be drawn from different experimental strategies vary widely. While some approaches only identify regions that may be transcriptionally active (eg. DNase hypersensitivity), others directly pin-point the binding of a protein to a sequence (eg. ChIP-chip). Most do not clear the dilemma of whether binding necessarily implies function (except probably devoted functional assays). Additionally, few approaches are scaleable to genome-wide investigations. Recently, large scale approaches to map the chromatin structure [222, 148], as well as high-resolution tiling arrays (eg. [93]) have opened new avenues – both in the search for corroborating evidence for functional regulatory regions as well as in providing a genomic view of protein-DNA interactions.

In brief, however, the task of identifying functionally relevant sequence elements is a daunting one. It is complicated by numerous factors like the amount of non-coding sequence that is assumed to contain regulatory signals, the number and combinations of transcription factors that are anticipated to be involved, and the cost and complexity of designing experiments – all of which make experimental validation non-trivial. Integrated approaches that make use of computational strategies to extrapolate on available experimental knowledge are a critical necessity. In the following, we discuss this computational side of the problem.

1.2.2 Representation and background

To design a computational strategy one requires a well-formulated representation (*model*) of binding sites. Employing experimental knowledge of binding sites, where available, may assist in making this choice. A model needs to satisfy two criteria – **i)** appropriately reflect the shared features of experimentally verified sites and **ii)** be suitable for use in mathematical and computational applications.

The first criterion enables *comparison* against the repertoire of known binding sites to identify similar motifs. This contributes in applications aiming to *predict* novel instances of binding sites for a factor. Combined with the second criterion, it also gives the possibility to *generate* sample instances of known sites for analytical purposes. Representative models used predominantly in current methods fulfill these criteria to quite some extent.

It is clear that using a single unambiguous sequence to represent a collection of known sites with varying number of degenerate nucleotides may be insufficient. Along with being too restrictive, it does not do justice to the TFBS characteristics, especially if the degeneracy at some positions is high.

By simply aligning the known binding sites, the small invariant core sequence of known sites can be extracted and the number of occurrences of each nucleotide at each position counted. This alignment forms the basis of two broad categories of TFBS models – *sequence*-based and *matrix*-based.

Sequence-based This class uses a representative sequence (*motif*), which allows slight mismatches, as the TFBS model. Using the alignment of experimentally verified sites, a *consensus* sequence is formulated by considering the predominant base at each position in the alignment. Thus, a consensus sequence is close to all known sites of a factor with some mismatches. A more sophisticated version is one that incorporates ambiguous positions too. These are either shown as possible alternatives in the same sequence representation or by using the IUPAC nomenclature to indicate subsets of nucleotides. For example, the consensus sequence for the binding sites of the yeast transcription factor CSRE is CGGAYRRAWGG (taken from the SCPD database [224]), where Y is either C or T, R is either A or G and W is either A or T.

Prior to their usage for representing transcription factor binding sites [60], consensus sequences have long been used for multiple sequence comparisons. For example, Kozak [106] and Cavener [38], used consensus sequences to identify initiation of translation and termination sites. Waterman and colleagues [211] present algorithms for finding consensus patterns and estimating their statistical significance. A detailed comparative study of consensus-pattern finding methods in the early nineties can be found in [46].

Once formulated, a consensus sequence can then be used to search for putative TFBSs by scanning novel DNA sequences for exact or inexact matches. How close a putative site is to the consensus sequence is judged by the number of matching positions. Clearly, the sensitivity of the consensus sequence in making predictions depends on the amount of mismatches allowed to consider a site a putative *hit* while searching.

Although consensus sequences afford more leniency as opposed to strict single sequence, they do not quantitatively represent the nucleotide distribution at each position in a binding site. Additionally, when being used to predict new occurrences, it is often a complicated task to decide the level of allowed mismatches. This is where the second class of matrix based models comes handy.

Matrix-based The biases in the nucleotide distribution at each position in the TFBS can be modeled probabilistically in a matrix framework, known as *profiles*. Although the term profile has been variously used to mean position-specific count, weight or frequency matrix, here it solely refers to position-specific frequency (probability) matrix.

A profile specifies the probability of observing a nucleotide at each position in the TFBS. More formally, a profile P of length l over an alphabet Σ is an $l \times |\Sigma|$ matrix (P_{ij}) $(i = 1, \ldots l; j \in \Sigma)$, such that $\mathrm{P}_{ij} \geq 0$ for all i, j and $\sum_{j \in \Sigma} \mathrm{P}_{ij} = 1$ for all i. An example is provided later in Chapter 2.

The review by Stormo [188] provides a thorough coverage of the developments in the field of binding sites, from detection, representation to statistical properties, and is an excellent reference. Although most relevant literature is cited therein, the classical article by Berg and von Hippel needs mentioning here [20]. In this article, the authors showed that the logarithms of base frequencies are proportional to the binding energy contribution of the bases. Their work forms the basis of most modern day approaches for studying binding affinity (examples include Roider *et al.* [165]). Articles addressing the issue of representation of binding sites, evaluation, and prediction include [166, 147, 150, 200].

Profiles encapsulate the information about both the preferred and the alternative nucleotides at a position. They can be used to evaluate how likely a new motif is to be a binding site of a factor. Additionally, they can be used as a motif *generator*, where each nucleotide is generated independently according to the position-specific distribution. In other words, the probability that the profile P of length l generates a sequence $\mathbf{u} = (u_1, \dots u_l) \in \Sigma^l$ is given by $\mathbb{P}_\mathrm{P}(\mathbf{u}) = \prod_{i=1}^{l} \mathrm{P}^i(u_i)$.

On the basis of profiles, position-specific weight matrices are formulated that allocate scores to each nucleotide at a position in a string. The final score of the string is the sum of these individual scores. Hence, the consensus string would have the best score while all other sequences would have lower scores with the decreases depending on the difference to the consensus. Such a score of a novel DNA sequence is then compared to a threshold to identify whether it is a putative TFBS hit or not. Profiles form the core of the work presented here and we will present examples and discuss formally the theoretical background and application in Section 2.1 of Chapter 2.

Comments We presented the two most common classes of TFBS representation – consensus sequences and profiles. Both the classes share the assumption that each position in a site behaves independently of the others, that is each position contributes additively. This assumption is questionable as has been discussed in Benos *et al.* [17] and Bulyk *et al.* [37]. Sophisticated models taking position-dependency into account or considering non-linear contributions of each base have also been proposed [14, 55, 83, 68].

1.3 TFBS methods that do not use cross-species comparisons

One look at the literature in computational biology from the last two decades would show that a plethora of algorithms and applications for TFBS prediction have been proposed. Classifying the various approaches along a certain axis is immensely complicated – use consensus sequences, matrix models or position-dependent models, use comparative genomics or gene expression data or clustering information or a combination of a subset or all of them, use *a priori* information of binding preferences of a factor or discover new motifs – the differences are in color as well as shade. Usually, the broadest and most applicable classification is based on the last criterion, where methods are either:

- *ab initio* discovery methods – those which assume nothing about the binding preferences of a factor and use a set of DNA sequences believed to be co-regulated to learn new motifs, or
- TFBS identification methods – those which use experimentally discovered binding sites of a factor to identify novel occurrences matching the preferred description.

In this section, we focus on a subset of these classes of methods common during the pregenomic era – those which do *not* use cross-species comparisons. While their discussion is relevant to understand the current state-of-the-art, their relevance to this work, a TFBS identification method that uses cross-species comparisons, is limited.

1.3.1 Ab initio discovery methods

As Brazma *et al.* [30] put it, the computational TFBS discovery problem is "in essence extracting general rules from particular instances". A set of input sequences believed to be co-regulated is picked. These sequences are likely to share common motifs that might be bound by one or more transcription factors. The aim is to find patterns that reflect the characteristics of the most over-represented motifs in this set.

For a review over pattern discovery methods employed in computational biology, we refer the reader to the detailed work of Rigoutsos *et al.* [162]. Providing a machine learning formulation of the problem, Brazma *et al.* [29, 30] also present a formal survey of pattern discovery methods and their relation to each other. Stormo [188] and Pavesi *et al.* [150], amongst other related issues, discuss both binding site representations as well as discovery approaches. In an appreciable feat of patience and diligence, Sandve *et al.* [171] have compiled an almost-complete list of current motif discovery tools and categorized them using a framework composed of four different levels. One can get a rough picture of the number of such methods by the extended table in their supplementary material, which lists more than hundred entries. A slightly older but similar, thorough work is that by Häussler [77] who studied the various motif discovery methods and discussed representative examples. Other than these recent works, an excellent and up-to-date review on motif discovery methods is provided by MacIsaac and Fraenkel in [117] and a recent introduction into the issues related to motif discovery is described in the work of D'haeseleer [49].

Different authors adopt different sub-classification of motif discovery methods: *enumerative* and *alignment-based* [117], *bottom-up* and *top-down* [29], *pattern-driven* and *sequence-driven* [30, 149], *consensus-based* and *alignment-based* [150], or based on *enumeration*, *deterministic optimization* and *probabilistic optimization* [49]. Despite the various terminologies, in essence methods come in the two main flavors of *enumerative* and *alignment-based* and we discuss them next.

Enumerative methods This class of methods first exhaustively enumerates the solution space by considering all possible patterns up to a certain (usually user-defined) length. Next, the patterns are scored and the best scoring ones are output. The majority of methods in this class are consensus sequence-based.

Tompa [194] used an enumerative method employing z-scores to rank motifs, for finding ribosomal binding sites in prokaryotic genomes. In a similar approach for yeast downstream sequences, van Helden *et al.* [199] allowed only exact matches to consensus sequences to find short, contiguous over-represented motifs. In another tool MITRA [58], efficient data structures are employed to cover the space of IUPAC patterns and a hypergeometric score used to rank them. Other approaches include works by Staden [186], Brazma *et al.* [31], Sinha *et al.* [181], Blanchette *et al.* [22], etc.

Recently, Tompa and others performed a large-scale evaluation of 13 motif discovery tools (including non-enumerative tools) using both simulated and real data from fly, human, mouse and yeast [195] and the tool Weeder [151] showed the best performance. Weeder enumerates all motifs to a maximum length and weighs each according to the number of sequences it occurs in and the level of conservation. However, while most of the tools performed well on yeast datasets, the overall performance on other species was poor. Other contributions evaluating discovery methods include the works of Day and McMorris [46] (compared consensus discovery methods) , Pevzner and Sze [154] (their metrics for method evaluation were used in the study of Tompa *et al.* [195]), Hu and Kihara [85] (amongst other things, studied the appropriate number of input sequences required), and Sandve and colleagues [171].

Exhaustive enumeration guarantees that all motifs of a certain length have been considered. But enumerating the solution space gets increasingly difficult as the length of the patterns increases. For an input sequence set of N sequences of length m defined on an alphabet of size A, the time complexity of enumerating the solution space of motifs of length L with e allowed errors is $O(NmA^eL^e)$ [149, 117]. Hence, usually methods adopt an upper bound on the motif length as well as allowable errors. As a specific formulation, Pevzner and Sze [154] considered the case of $L = 15$ and $e = 4$ in an artificial setting. They proposed two graph-based approaches, WINNOWER and SP-STAR, that performed well in comparison to the existing approaches. Suffix-tree based approaches provide a speed-up in accessing the words of a text, and initial efforts using suffix trees in the context of motif-discovery problems were made by Sagot and colleagues [167, 121]. The tool Multiprofiler [97] searches for all substrings with exactly k mismatches to a given string, that is the k-neighbourhood of the string. This neighbourood is then scanned for the motif by searching it for recurring *wordlets*, subsequences of the motif. A hash-table based approach was proposed by Buhler and Tompa [35] where l-length substrings are put together in the same bucket if they have the same letters at x positions. These pre-processing steps save a lot of time compared to complete sample driven enumeration. Afterwards, five iterations of Expectation maximization (explained later) are carried out on every bucket and the resulting motif instances are refined using the graph-based approach SP-STAR. Recently, an improved version Aggregation [102] that provides a more than double speed-up on the original algorithm has been proposed that analyses dense regions of the subspace rather than dense points.

Alignment-based methods Instead of enumerating all existing n-mers and checking how well they are shared amongst the input sequences, this class starts the other way round. It searches for local similarities in the input sequences with the hypothesis that a combination of the most common patterns might form the basis of putative TFBSs. Thus, the only limiting parameter is the motif length. Initial methods have focused on using multiple sequence alignments to extract similar regions. However, the problem of finding the best pattern in multiple sequences has been proven to be NP-hard [3], and usually methods adopt heuristics. For review on methods as well as adopted heuristics, see Brazma *et al.* [30] and references therein.

The basic idea is to find parameters of a binding site model that best describe the observed sequence set and the corresponding optimization is carried out usually using one of two popular core approaches: *expectation maximization* and *Gibbs sampling*. For a detailed introduction into the two algorithms, we refer the reader to the excellent book by Liu [112].

Let us assume that a motif of length w is being searched in an input set of k sequences. The background model is taken to be the uniform distribution, corresponding to equal base frequencies.

- **Expectation-maximization** – Given an initial setting of parameters, in this case related to the binding site model, the aim is to find the binding site descriptor that best explains the observed sequence set. The tool MEME (Multiple Expectation Maximization for Motif Elicitation) [10, 11] can be used as a representative example for discussion. In the E-step the likelihood of observing the data under the current model is calculated. Then, in the M-step the parameters are updated so as to maximize this likelihood. The steps are iterated until no gain is obtained and the corresponding model is output. After removing the corresponding substrings from the input, the algorithm is re-started to search for more motifs. Thus, MEME searches the space of all binding site models for the one that best explains the observed input. The EM is susceptible to be stuck in local maxima and hence usually it is initialized a number of times to improve the results. Related approaches include Blekas *et al.* [25], Improbizer [8] and PhyMe [180].

- **Gibbs sampling** – The basic idea here is to find the multiple alignment of small sequence segments of the input set which is most likely to consist of samples from a common binding site model. The Gibbs sampler is the representative example [108]. Initially, w-mers are picked randomly from each sequence. Keeping one sequence fixed, a binding site model is formulated from the substrings of the rest of the sequences. Using this model, the likelihood of each w-substring in the fixed sequence is calculated versus the background model. These probabilities are then used to pick substrings in the next iteration, implying that positions which represent the model best are more likely to be picked in the next iteration. The iterative process is carried out until equilibrium is reached from which the more probable alignment can be identified. Hence, Gibbs sampling is a stochastic version of EM with a wider radius of convergence. It is computationally expensive and requires multiple runs to cover the various probability surfaces. Extensions to the basic algorithm include the works of Neuwald *et al.* [144], Rocke *et al.* [163], etc. and tools like AlignACE [87], Ann-Spec [216], GLAM [65], SeSiMCMC [61], etc.

Other methods that fall in neither or both categories also exist. For example, the Consensus method of Hertz and Stormo [79] adopts a greedy algorithm to save the best partial alignments at each step, hoping that they will eventually lead to the optimal one.

Extending the work of Harbison *et al.* [75], Hu and others [85] came to the conclusion that a combination of multiple motif discovery algorithms perform better than a single tool. In their review article, MacIsaac and colleagues [117] suggest to adopt a consistent scoring metric which is either based on hypergeometric enrichment [13] or on the area under the receiver operator characteristic curve (ROC). They also report that taking clustering of binding sites into consideration also improves performance.

1.3.2 Methods using a priori knowledge

The objective here is to use available binding site data to search for novel matches according to a pre-defined criterion. This has the pre-requisite that sufficient experimental data for a factor is available. Although modern large scale experimental approaches are encouraging, still detailed knowledge of transcription factors and their binding sites is insufficient. Probably the largest and most popular database of known binding sites is the TRANSFAC database [123, 124]. However, many well-characterized factors have multiple similar matrix descriptions in TRANSFAC, thus introducing a relatively high amount of redundancy [169, 164, 174]. A smaller but better quality database is the JASPAR [168, 204] database; other databases include SCPD [224], TRRD [103, 104], TRED [223], ABS [23], etc.

The standard approach is to get a binding site model, either by picking up one from such databases or by formulating one from a collection of known binding sites, and show its applicability on detecting known instances and predicting novel ones. Although the field of TFBS identification approaches is immense, most existing approaches use additional information. A few highlights of the basic approaches and tools for TFBS scanning are mentioned below.

Tronche *et al.* [196] formulate a weight matrix from a database of known binding sites of the transcription factor HNF1, and identify more than hundred liver-specific genes that can be putative targets of HNF1. In a similar work, Fickett [63] considers MEF2 binding sites in the upstream regions of muscle-specific genes. In a more recent work, Johnson *et al.* [89], the authors form a position-specific scoring matrix of RE1 (repressor element) and validate its applicability on a positive and negative training set.

The web tool TESS [175] for TFBS scanning has a mismatch (consensus string) or score (weight matrix) threshold which is user-defined and hence not statistically motivated. In the database and search engine MAPPER [119, 120], the aligned known binding sites are modeled using hidden markov models allowing indels. The authors built approximately thousand such TFBS models and used them to scan sequences of various species using statistically derived thresholds. The tool MATCH [98] adopts a TFBS hit scoring scheme composed of two scores, a matrix similarity score and a core similarity score. The score cutoffs for a match are statistically derived using either of three criteria: minimize false positive error rates, minimize false negative error rates, and minimize the sum of both errors. Other examples are PATSER [80] and the work of Rahmann *et al.* [158]. The latter forms the basis of the profile part of the algorithm presented in this thesis, and we shall discuss it in more detail in Chapter 2.

Summary All TFBS prediction methods – *ab initio* discovery or identification using *a priori* knowledge – irrespective of the model used (consensus- or matrix- based) suffer from the inherent problem that arises due to the short length and degeneracy of the TFBSs. On top of that the sequence search space is large making it difficult to distinguish TFBSs from background sequences. Usually additional support is used as an extra filter to reduce the immense number of false predictions. And for our purposes, the concerned information is conservation across species. Before going into the TFBS prediction methods that employ such cross-species comparisons, we present an insight into the issues, algorithms and applications of alignments, the core of comparative genomics.

1.4 Alignments – why, what and how?

Alignments are a way to represent the similarities between sequences. In the field of computational biology, their relevance cannot be sufficiently stressed. From their use in extracting meaningful information about the conserved and variable regions between sequences, identifying possible errors in molecular data to elucidating mutational processes responsible for these, alignments are indispensable. They fall in the more inter-disciplinary field of sequence comparison methods which have their roots in signal processing and information theory. How do sequence comparison methods fit into the field of molecular biology?

From a computational perspective, a genome is simply a large linear sequence of letters from an alphabet. Parts of this sequence are responsible for encoding the molecular building blocks and mechanisms needed for the sustenance of life. Different organisms share this repertoire of bio-molecules and mechanisms, either identically or with some modifications. These functional, structural and morphological similarities are attributed to the process of evolution.

Through a process of mutational events, evolution “reuses, builds on, duplicates, and modifies *successful* structures” [72] to yield the similarities and differences observed in present-day genomes. These mutational events include **a)** genome-scale changes like translocations (exchange of regions between different chromosomes), transpositions (exchange on the same chromosome), inversions (sequences reversed), duplications, etc. and **b)** local point mutations like substitutions (a letter exchanged by another), insertions or deletions. Since ancestral sequences are not available, one resorts to comparing contemporary sequences to extract regions with high sequence similarities. The hope is to identify putative *homologous* sequences, i.e. sequences that might have diverged from a common ancestor, and thereby attribute similar functional or structural characteristics to them. This is where sequence comparisons come into the picture.

1.4.1 Background

There are two perspectives of viewing the problem of sequence comparison: *edit-distance* based and, more relevant to us, alignments. In the former, a process of edit operations is used to convert one sequence to another. These operations correspond to mutational events like substitutions, insertions and deletions, and hence allow a direct evolutionary interpretation to the sequence comparison problem. Here, a sequence has the least distance to itself and greater distances to less related sequences. Alignments are better framed in a *similarity* context – a sequence is most similar to itself and this similarity decreases with increasingly different sequences. Although both perspectives provide a means of measuring the degree by which sequences are alike or different, they are not identical since different series of mutational events, or equivalently different sequence of edit operations, may yield the same alignment, which is neutral to evolutionary history. Under certain conditions, Smith and Waterman [182] showed that the two measures are equivalent in global alignments. Our focus is on similarity-based alignment methods which have been shown to be more useful for identifying local similarities [183].

1.4.2 Standard alignments – Model and Scoring scheme

The aim of alignments is to capture sequence relatedness that might reflect functional similarities. To identify the biologically interesting instances of sequence similarities, it is therefore essential to choose a scoring scheme that brings such instances to the forefront and limits false positives. Thus equipped, the problem of identifying similar regions between two

i:	1	2	3	4	5	6	7	8	9	10	11
x_i^*:	A	C	–	G	T	A	T	A	A	T	C
y_i^*:	A	C	C	A	T	A	T	A	–	T	C
α_i:	S	S	I	S	S	S	S	S	D	S	S

Figure 1.3: *Example of standard alignment.* A possible alignment of $x = \mathrm{ACGTATAATC}$ and $y = \mathrm{ACCATATATC}$ under the standard alignment model. The alignment is coded as $A = (\mathrm{SSISSSSSDSS})$. Gaps as represented by dashes. Each column is scored according to whether it is a substitution, insertion or deletion. The scores are summed to yield the alignment score.

sequences reduces simply to identifying the outstanding alignments under such a scoring scheme and model. Let x and y be two sequences of lengths m and n, respectively.

A gapped (global) alignment $\mathcal{A}$ between x and y introduces gaps in the sequences, while maintaining their order such that the lengths of the resulting sequences x^* and y^* are identical. The alignment stacks the gapped sequences one upon the other with no gap being above another (Figure 1.3).

A column with nucleotide letters in each row is called a *substitution* (S), with identical nucleotides referred to as *matches* and un-identical ones as *mismatches.* A column with a gap in the first sequence and a nucleotide in the second is called an *insertion* (I) and the other way round is called a *deletion* (D). Together, they are referred to as *indels.* Thus, an alignment can be coded as a string of letters from the set $\{\mathrm{S}, \mathrm{I}, \mathrm{D}\}$ representing the sequence of its columns. In practice, a deletion is not immediately followed by an insertion and vice versa. An example of such an alignment is shown in Figure 1.3.

Let Σ be a finite alphabet, in our context consisting of the DNA nucleotides $\mathrm{A}, \mathrm{C}, \mathrm{G}$, and T. Hence, $|\Sigma| = 4$. If we define $\ell(\mathcal{A})$ as the length of the alignment $\mathcal{A}$ and $n_{\mathrm{S}}(\mathcal{A})$, $n_{\mathrm{D}}(\mathcal{A})$ and $n_{\mathrm{I}}(\mathcal{A})$ as the respective counts of substitutions, insertions and deletions, then:

$$\mathcal{A} = (\alpha_1, \alpha_2, \ldots, \alpha_{\ell(\mathcal{A})}) \in \{\mathrm{S}, \mathrm{D}, \mathrm{I}\}^{\ell(\mathcal{A})} \tag{1.1}$$

is a valid alignment between the two sequences x and y *iff*

$$n_{\mathrm{S}}(\mathcal{A}) + n_{\mathrm{D}}(\mathcal{A}) = m \quad \text{and} \quad n_{\mathrm{S}}(\mathcal{A}) + n_{\mathrm{I}}(\mathcal{A}) = n \tag{1.2}$$

It is clear that an alignment with the above characteristics directly defines a mapping

$$\begin{aligned} \mathcal{A}\colon \Sigma^m \times \Sigma^n &\to (\Sigma^*)^{\ell(\mathcal{A})} \times (\Sigma^*)^{\ell(\mathcal{A})} \\ (x, y) &\mapsto (x^*, y^*), \end{aligned}$$

where $\Sigma^* := \Sigma \cup \{-\}$ and $\ell(\mathcal{A}) = n_{\mathrm{S}}(\mathcal{A}) + n_{\mathrm{D}}(\mathcal{A}) + n_{\mathrm{I}}(\mathcal{A})$, implying that each column constitutes a character pair from Σ^*.

Scoring Scheme Every column i in $\mathcal{A}$ is assigned a column score $\mathcal{S}_c(\alpha_i, x_i^*, y_i^*)$. Each substitution column contributes an additive substitution score, retrieved from a *substitution scoring matrix* $\mathtt{s} = (\mathtt{s}(i, j))$, $i, j \in \Sigma$. And in the simplest case, every indel column contributes a subtractive *gap penalty* g. Thus for every column i, the scoring scheme in standard alignments defines a score as:

$$\mathcal{S}_c(\alpha_i, x_i^*, y_i^*) = \begin{cases} \mathtt{s}(x_i^*, y_i^*), & \text{if } \alpha_i = \mathrm{S} \\ -\mathrm{g}, & \text{if } \alpha_i = \mathrm{D}, \mathrm{I} \end{cases} \tag{1.3}$$

That is, a score as retrieved from a substitution scoring matrix $\mathtt{s}$ is added for every substitution column while a gap penalty g is subtracted for every indel. Finally, the score $\mathcal{S}(\mathcal{A}, x^*, y^*)$ for the complete alignment is simply calculated as the sum of the column scores, or

$$\mathcal{S}(\mathcal{A}, x^*, y^*) = \sum_{i=1}^{\ell(\mathcal{A})} \mathcal{S}_c(\alpha_i, x_i^*, y_i^*) \tag{1.4}$$

It is worthwhile to mention here that treating each gap as a single mutational event and hence penalizing all single gaps identically, is the simplest gap cost case (*linear*). It is widely accepted that in reality multiple consecutive gaps are a result of a *single* mutational event. Hence, a more flexible gap cost function that takes this into account needs to be employed. Since the correct choice of score parameters is essential for the correct interpretation of alignments, we discuss this in more detail Chapter 2. For now, let us assume that the scores and gap costs are appropriately set to reflect the evolutionary distance between the concerned sequences. Once we have the model and the score parameters, it is possible to define the optimal global alignment problem:

> *Given the scoring matrix* $\mathtt{s}$ *and the gap penalty* g*, find the alignment with the highest score amongst all possible alignments between* x *and* y*.*

A naive approach to calculating the optimal alignment could be to enumerate all possible alignments, score and rank them. For a pair of sequences of length n each, it has been shown that the number of alignments is exponential in n [210], making the naive approach computationally infeasible for long sequences. More efficient dynamic programming based approaches have since then been proposed to calculate both *global* and *local* alignments. Alignments are global when the whole sequences are aligned, and *local* when the focus is on subsequence similarities. The former is useful for reconstructing evolutionary history while the latter is more relevant for finding shorter regions of high similarity within otherwise weakly related sequences.

This work uses and builds upon a pairwise, dynamic programming based local alignment approach and in the following, algorithms for multiple sequence alignments are not discussed. The interested reader is referred to reviews by Batzoglou [15], Wallace *et al.* [205], and Apostolico *et al.* [9] for an overview of multiple sequence alignment approaches.

1.4.3 Overview of pairwise alignment methods

As with TFBS prediction methods (Section 1.3), classification of alignments and related algorithms is non-trivial. In this section, we concentrate on the major algorithms in pairwise sequence alignment methods; introductory textbooks by Gusfield [72], Waterman [210] and Setubal *et al.* [177] are excellent references for a more in-depth coverage. On the broader and more detailed algorithmic issues, the book by Sankoff and Kruskal [173] is the classic reference.

The simplest way to represent the relationship between a pair of sequences is through *dot-plots*. Here, sequences are written as the top row and the leftmost column of a two-dimensional matrix and a dot placed at the point in the matrix where the corresponding row and column entries match. Dot plots provide a basic visualization of the similarities between the sequences and are especially useful to identify self-similarities or repeats. However, besides being qualitative, for large sequences they prove to be time-consuming. Available tools for calculating and visualizing dot plots include Dotter by Sonnhammer and Durbin [184], Dotlet by Junier and Pagni [91], Gepard by Krumsiek *et al.* [107], etc.

Dynamic Programming based approaches

Dynamic programming is perhaps the most widely used means of calculating alignments. The underlying idea is to break a bigger problem into several smaller similar problems and build upon their solutions to yield the solution to the bigger problem.

Saul Needleman and Christian Wunsch [143] first proposed a dynamic programming based algorithm for calculating optimal global alignments with linear gap costs, yielding an $O(n^2)$ complexity. A two-dimensional matrix is constructed where the row and column correspond to the concerned sequences. Each cell in the matrix is filled using a recursion rule that optimizes the score of the sequences uptil that point. The scores are calculated using a DNA substitution scoring matrix and gap costs. A variation of the Needleman-Wunsch (NW) algorithm was proposed by Smith and Waterman [183] to calculate local alignments. The annotated alignment algorithm is a variation of this Smith-Waterman (SW) algorithm and we formally describe the dynamic programming algorithm in Chapter 2. For a detailed coverage on the historical inception of dynamic programming into computational biology, the review by Sankoff [172] is an excellent reference. An introductory review to dynamic programming is also provided in the work of Eddy [54].

The NW and SW algorithms use a gap cost system where each gap is penalized equally. Variations of alignment algorithms that consider more sophisticated gap cost models have also been proposed. A linear gap cost version of the basic dynamic programming approach was presented by Gotoh [70].

Additionally, the above approaches have a quadratic space and time complexity in the length of the sequences. Hence, for long sequences they prove infeasible. More space- and time-efficient algorithms have also been proposed. For example, Hirschberg [82] proposed a linear space alignment algorithm which calculates the dynamic programming matrix using a divide and conquer method that breaks the matrix into two halves that are calculated independently. In another effort, Myers and Miller [140] applied these concepts for the linear gap cost case in Gotoh's approach. A review on linear space alignment methods is provided by Chao *et al.* [40].

An optimal alignment need not necessarily be the biologically correct one. Additionally, when sequence similarity is low, optimal alignment algorithms can be extremely sensitive to parameter choice while biologically meaningful similarities should be robust to such variations. To deal with such ambiguities, researchers often study alternative alignments with nearly-optimal scores, ie. suboptimal or near-optimal alignments. A quadratic complexity dynamic programming based algorithm that forms the basis of later approaches is the Waterman-Eggert [212] algorithm which calculates the k best non-intersecting alignments (described in Chapter 2). Linear space implementations for calculating suboptimal alignments have since then been proposed (eg. [86, 39, 142]). The review article by Vingron [202] traces the developments as well as the various notions of near-optimal alignments.

Heuristic approaches

Despite the advances in dynamic programming approaches in terms of efficiency, it is still infeasible to be used for genomic length sequences. Heuristic methods, also known as *k-tuple* methods, provide a faster although not necessarily an optimal solution. These methods are especially useful in large-scale database searches where it is understood that a large proportion of the candidate sequences will have essentially no significant match with the query sequence. The underlying idea is to first find short contiguous stretches of aligned nucleotides and to use these stretches as anchors to extend the alignment.

The two most popular heuristic methods are BLAST [7] and FASTA [111]. Given a query sequence to be searched against a database of sequences, the latter first builds a hash table of all k-tuple matches between the two. Nearby k-tuples separated by a constant distance in both sequences are joined into a short local alignment. Finally, using these short local alignments as seeds, it builds the longer alignment using dynamic programming. BLAST essentially follows the same strategy but evaluates only the most significant word matches. Seeds that are being extended in ways that are not typical of truly homologous sequences are thrown out. In recent years, numerous heuristic approaches on similar lines have been proposed, with varying seeding strategies (for a review see Brown *et al.* [32]). Other relevant reviews include those by Ureta-Vidal *et al.* [198] and Batzoglou [15].

Summary While the discussion here focussed mainly on pairwise alignment approaches (mostly dynamic programming based), the field itself has been much explored. For review articles providing an overview of the initial efforts in protein and DNA alignments, we refer the reader to those by Waterman and colleagues [211, 209]. Apostolico and Giancarlo [9] also trace the broad developments in the field of sequence comparisons, although the focus is more on bridging between the computational, mathematical and biological expectations from alignments. A detailed review of the more sophisticated algorithms for pairwise alignments is provided by Myers [139]. More recently, the review article by Batzoglou [15] discusses the various standard as well as novel alignment algorithms, covering wide areas like multiple sequence alignments and synteny rearrangements.

Research in the field of alignments has revolved not just around algorithms and tools, but also around other related issues like parameter choice (eg. [203]) and statistical significance of alignments (eg. [95, 6]). While both are interesting fields in themselves, for the purpose of this thesis, the discussion is restricted to the introduction of the theory behind common strategies for parameter choice (Chapter 2).

1.5 TFBS methods that use cross-species comparisons

Having given a brief insight into both, single sequence TFBS prediction methods and alignments for comparative genomics, we are now in a position to explore TFBS prediction methods that employ cross-species comparisons. Additionally, the aim of comparative sequence analysis is to take advantage of the underlying evolutionary relationship between the sequences. To this end, during the discussion we highlight those that explicitly consider TFBS evolution.

1.5.1 Ab initio discovery

Searching for short, degenerate motifs in sequence sets which can be comparatively much larger, may yield a low signal-to-noise ratio. Phylogenetic footprinting-based TFBS discovery methods usually either restrict the input sequence sets to only conserved stretches or apply a conservation filter on the predicted motifs.

McGuire *et al.* [128] used the Gibbs sampling based approach AlignACE [87] to search for motifs in microbial genomes with the input sequence set restricted to regions conserved to *E. coli*. In a similar approach, McCue *et al.* [126] used an extended Gibbs motif sampler to search for binding sites conserved in the upstream of several proteo-bacterial genomes. Xie *et al.* [221] predict motifs that are both conserved across multiple genomes as well as over-represented across the genes of the species. For reviews on the existing methods as well as issues related to using phylogenetic footprinting for motif discovery, we suggest the articles by McCue and colleagues [127] and more recently, by Prakash *et al.* [156] and MacIsaac *et al.* [117].

Methods explicitly modeling binding site evolution Although a wide variety of comparative motif discovery methods exist, only a few model evolution of binding sites explicitly.

PhyMe [180] proposed by Sinha and Tompa, combines the two axes of over-representation and conservation using expectation maximization. The authors use a probabilistic model dependent on the evolutionary distance between the respective species to model the relationship between orthologous binding sites. Like MEME [10], the objective is to find the binding site description that best explains the data. The probabilistic model for binding site evolution takes into account the observation that binding sites evolve under selectional constraints as prescribed by their representative profiles. The algorithm EMnEM [132] is also based on MEME, but here the binding site evolution is modeled by assuming an overall slower rate of evolution as compared to the surrounding sequences.

A Gibbs sampling based approach that explicitly models binding site evolution is PhyloGibbs, proposed by Sidharthan *et al.* [178]. It assumes that mutations are introduced at a fixed rate but the probability of selection to fix the mutation is proportional to the profile distribution at a position. Again, two models of evolution are considered, background and binding site, and the algorithm searches for all binding sites that can describe the input data given in the form of multiple sequence alignments.

1.5.2 Methods using a priori knowledge

Current comparative methods in this field perform the task in two main steps. In one step, conserved regions of two orthologous sequences are extracted using a method-specific alignment algorithm and a conservation criterion. In a separate step, binding site models (usually position-specific scoring matrices (PSSMs)) are used to scan individual sequences for putative TFBSs. Finally, the alignment and annotation results are combined to predict conserved TFBSs (Figure 1.4). The following exposition provides a concise overview of existing comparative TFBS identification methods. The emphasis is on method-specific strategies adopted to deal with the above-mentioned core issues, namely:

- extraction of conserved regions – this corresponds to the alignment algorithm used and the conservation criterion defined,
- TFBS scanning – this covers the TFBS model used and the derivation of a hit cutoff, and finally,
- combining the last two – this addresses how the conservation information and the single sequence scanning results are combined.

The tool ConSite [109] is the prototypical comparative TFBS identification method. Alignments are generated and conserved regions extracted. Then, individual sequences are scanned for putative hits using a score cutoff which does not consider the background letter distribution. Finally, only those hits that are situated in conserved regions and lie at equivalent positions in the alignment are output as conserved pairs. It is possible here that a true hit is missed either due to underlying alignment errors or failing to cross the matrix score threshold. Additionally, an aligned pair need not necessarily be devoid of indels (discussed later in Chapter 4). The tool was evaluated on a set of experimentally verified sites in human-mouse sequences. The authors show that incorporating conservation information improves performance significantly as opposed to single species scanning. However, the exceptionally good performance comes at the cost of more false predictions.

Another example is CisOrtho [21] which provides a multi-step framework for C.elegans and C.briggsae comparisons. The outline of the approach is shown in Figure 1.5. Briefly, binding site matrices are formulated by using the tool HMMER [53] for aligning known binding sites. These are then used to scan for the top N scoring hits. The score threshold again has no statistical motivation. Known ortholog sequence pairs that do not contain high-scoring hits are filtered out and the highest scoring hits in the rest are paired. The hit pairs are sorted according to hit scores and level of conservation. Hence, the sequences themselves are not aligned.

Dieterich *et al.* [51] use a modified version of the Waterman-Eggert algorithm [212] to calculate suboptimal local alignments and extract conserved non-coding blocks. For TFBS scanning, the statistically motivated method of Rahmann *et al.* [158] is used, where putative hits on the reference genome are identified. Finally, those hits that lie inside the conserved non-coding blocks are considered as conserved binding sites.

Other than the basic approach of aligning, scanning and combining the results, a variety of methods that include auxiliary information like gene expression, or clustering of binding sites have also been proposed. This can either be gene expression data like in oPossum

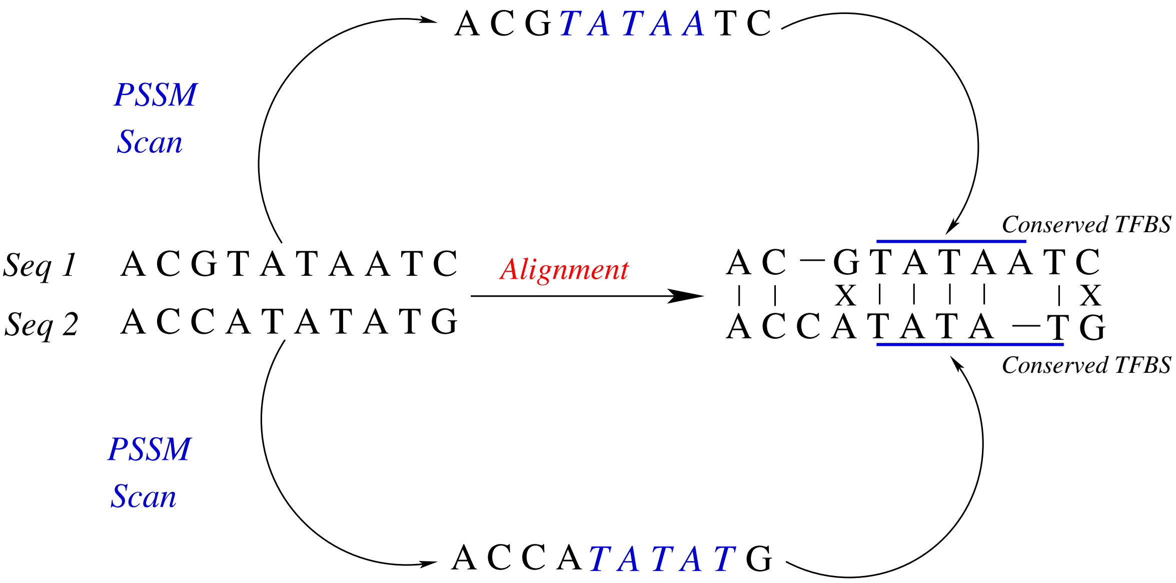

Figure 1.4: *Two step approach to conserved TFBS prediction.* Binding site models (usually PSSMs) are scanned across individual sequences to search for matches. Separately, the sequences are aligned to find highly conserved regions. Finally, the individual sequence hits are mapped onto the conserved regions to predict conserved TFBSs. As can be seen, the indels in the aligned hits are either ignored completely, or allowed only to a conservative limit.

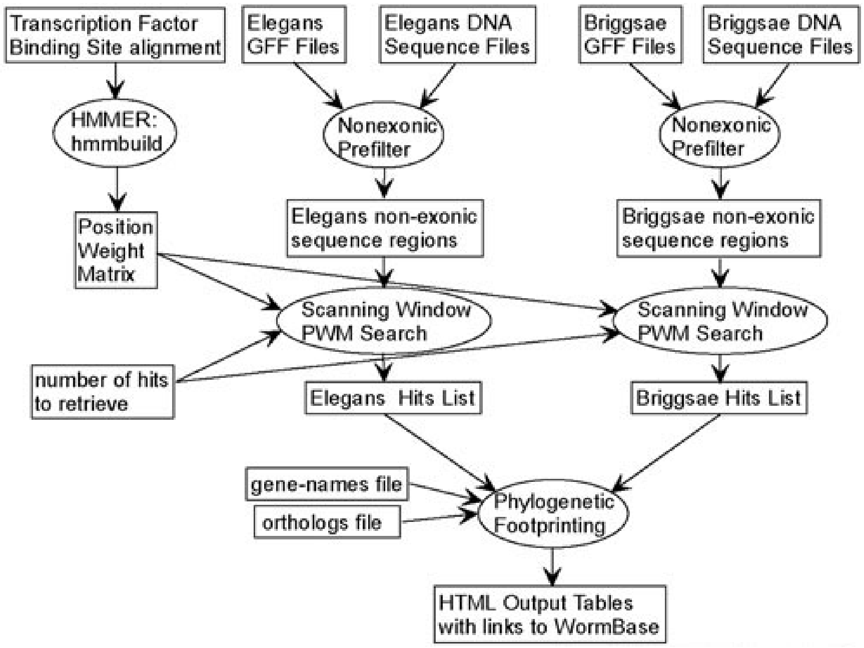
Transcription Factor Binding Site alignment
Elegans GFF Files
Elegans DNA Sequence Files
Briggsae GFF Files
Briggsae DNA Sequence Files
HMMER: hmmbuild
Nonexonic Prefilter
Nonexonic Prefilter
Position Weight Matrix
Elegans non-exonic sequence regions
Briggsae non-exonic sequence regions
Scanning Window PWM Search
Scanning Window PWM Search
number of hits to retrieve
Elegans Hits List
Briggsae Hits List
gene-names file
orthologs file
Phylogenetic Footprinting
HTML Output Tables with links to WormBase

Figure 1.5: *An example of a multi-step tool.* The program pipeline of the tool Cis-Ortho [21] is shown. Sequence information about *C. elegans* and *C. Briggsae* is processed to extract orthologous non-coding sequences. In a separate step, each sequence is scanned for putative hits using a position weight matrix. Finally, the sequences are aligned and top-scoring hits retrieved. Most current methods adopt such a multi-step approach to conserved TFBS prediction.

[191], clustering of TFBSs in conserved regions as in rVista [113], or relative positional preferences in Footer [44].

Another class of methods uses prior knowledge of TFBSs to construct the alignments. Putative TFBS hits on the single sequences are paired and used as anchors for producing either global [18] or local alignments [131]. While ConReal [18] focuses on generating an ordered chain of conserved TFBSs, thus not aligning regions that do not contain them, SITEBLAST [131] is a BLAST-like heuristic where the TFBS hits are used as seeds. The method of Hallikas *et al.* [73] also falls in this category. Here, the sequence of hit pairs is aligned using a scoring scheme that considers clustering of sites, binding affinity and conservation, though the underlying sequences themselves are not aligned. Recently, Blanco and colleagues presented different perspective to the problem of combining alignments and TFBS scanning [24]. Individual sequences are scanned for putative hits using TFBS matrices, from TRANSFAC [123] and JASPAR [168]. The putative hits are labeled with the corresponding factor. Finally, this sequence of labels is aligned using a modified version of the optimal alignment algorithm proposed by Waterman *et al.* [214] to generate TF-map alignments. Hence, the nucleotide alphabet is abstracted to the alphabet of TFBS labels. The authors show superior performance of the method in predicting conserved regulatory elements missed by other traditional approaches.

Methods explicitly modeling binding site evolution Monkey [134] is one of the few examples of methods that explicitly consider the TFBS evolutionary properties while searching for conserved TFBSs. It takes as input a pre-generated multiple alignment and then using the TFBS profiles compares the likelihood of observing the sequences under two evolutionary models, background and TFBS-specific. For the latter, it uses the Halpern-Bruno model [74], shown previously by the authors [133] to accurately model TFBS evolutionary properties. Since the alignment and annotation steps are de-coupled, the algorithm needs to employ heuristics for the aligned TFBSs containing gaps. More detail is provided in Chapter 4.

From the population genetics viewpoint, an approach has been proposed by Mustonen *et al.* [138] for modeling TFBS evolution. Here, the authors present and use a model for TFBS evolution in a method that calculates the likelihood of observing a set of aligned sequences under different modes of evolution. They also show that the specific evolution of binding loci can be integrated into a bioinformatics scoring procedure.

1.6 Motivation and proposed approach

In the preceding sections, we explored TFBS prediction methods that use cross-species comparisons to extract regions of high conservation. We saw that the rationale behind focussing on conserved regions is that they are likely to contain functionally relevant sequence features known to evolve slower than non-functional ones. Thus, a TFBS prediction approach relying on this principle should be able to align orthologous regulatory sequences and simultaneously predict conserved transcription factor binding sites. Although a plethora of approaches exist that provide a combination of conservation and TFBS annotation (reviews are provided in [198, 207]), to the best of our knowledge, none achieves this *simultaneously.*

A *conserved* TFBS pair should be one where single sequence hits represent the binding site profile *and* be accordingly evolutionarily related. It is widely accepted that evolutionary properties of TFBSs differ significantly from those of the surrounding sequences [127, 69, 48, 133, 19, 115, 105, 138, 215]. However as we saw, most current TFBS prediction methods based on phylogenetic footprinting do not consider TFBS-specific evolutionary properties while defining a conserved binding site. In one set of approaches, a conserved binding site is a site which lies in a highly conserved region, where the conservation criterion is usually percentage sequence identity. In such a region, the chances that the aligned site contains indels is almost negligible, irrespective of whether the corresponding motif on the other sequence is a TFBS or not. Examples include Wasserman *et al.* [206], Ji *et al.* [88], Dieterich *et al.* [51], etc. In another set of approaches, a site in a reference genome that is perfectly align-able with other species is taken to be conserved. Hence, the TFBS is assumed to evolve like the background. Methods in this category include Chiang *et al.* [41], Kellis *et al.* [99], Cliften *et al.* [43], etc. In yet another setting, a conserved binding site is one which has a TFBS at the equivalent position in the alignment. For example, ConSite [109], rVista [113], CisOrtho [21], etc.

Combined, the two issues discussed in the last two paragraphs address the core problem of *what constitutes a conserved TFBS?* High surrounding sequence conservation (sequence identity) does not necessarily imply high binding site conservation. In fact, even inside a binding site, conservation is not uniform. Research has shown that binding sites have a slower rate of evolution than the surrounding sequence, with functionally relevant positions evolving slower [133]. It has also been shown that degenerate positions too exhibit selectional constraints [105]. Can we take the TFBS-specific evolutionary properties into account while searching for conserved TFBSs?

Using a simple conservation criterion in the aligned binding site locations equates to ignoring the binding site specific evolutionary properties. A well-conserved sequence region may yield false predictions arising from high sequence similarity. Searching for high scoring TFBS hits, usually at equivalent positions in the alignment, implies assuming independence between the individual hits. In a more divergent sequence pair, more non-consensus nucleotide substitutions in a true TFBS may lead to a lower score, thus increasing the proportion of false negatives. Ideally, a method must be capable of predicting conserved TFBSs during alignment with background and TFBS evolutionary models to score the alignment.

In summary, most current methods for conserved TFBS prediction depend on a predetermined optimal alignment. They perform the TFBS *annotation* step separately from the *alignment* step. In the end, the predictions of the former are combined with the conservation information of the latter to output putative conserved TFBS hits. Depending on the quality of the concerned profile (ie. how specific is it?) or the relatedness of the sequences (ie. how similar are they?), it is possible that the underlying alignment fails in detecting such conserved pairs. We also saw how few methods explicitly model binding site evolution while searching for conserved TFBSs.

In this thesis, we propose a simultaneous alignment and annotation method – an extended pairwise alignment algorithm that addresses exactly these issues by providing a direct combination of the two steps to yield "annotated alignments". It introduces the possibility of annotating parts of an alignment as *pair-profile* hits. We also provide statistically motivated strategies for calculating the additional score parameters. Such an extended alignment approach and scoring scheme allows for local rearrangements in the alignment to bring together putative conserved hits. The algorithm is implemented as the **SimAnn** program [12]. To take binding site evolution into account, a modified scoring scheme is additionally presented and forms what is referred to as **eSimAnn**.

1.6.1 Brief outline

The aim of our method is to combine a locally optimal alignment of two sequences with an annotation with conserved pairs of TFBS profile hits. We therefore add the possibility of assigning parts of the alignment directly to such perfectly aligned pair-profile hits. This extension in the alignment scheme is introduced to allow for a different scoring of these hit pairs as follows.

Assume that we wish to search for conserved instances of a profile P of length l. A stretch of l consecutive gaplessly aligned positions can be scored in the extended alignment model in two possible ways. Either by scoring each aligned pair with the standard substitution scoring matrix `s`. Or by using a *profile scoring array* `PSA` and subtracting a *profile penalty p*. The profile scoring array assigns a score to every pair of strings of length l and reflects how well the gapless alignment of this pair fits to the motif described by P. The profile penalty is a tuning parameter meant to maintain the balance between the two alternatives. Hence, how a pair of strings is scored determines the difference between the standard alignment, the basic parameter estimation underlying SimAnn and the parameter estimation with evolution of binding sites underlying eSimAnn.

1.6.2 Annotated alignments – Model and Scoring scheme

To embed pair-profiles into the alignment model, along with the usual substitution and indel columns, we introduce additional columns representing pair-profile instances. For simplicity, we focus on only one TF whose binding sites are represented by the profile P of length l. Extensions to multiple factors are equally straightforward. The background profile is taken to be $\Pi = (\pi_j)_l$, $j \in \Sigma$, where π gives the uniform distribution. The definition of the alignment remains same – it is the introduction of a new labeling scheme that is novel here.

Model Given a single profile P (equally extendible for multiple profiles), an alignment $\mathcal{A}$ can now be coded as a string of letters from the extended set $\{\mathrm{S}, \mathrm{I}, \mathrm{D}, \mathrm{P}\}$, where a column labeled as P means that the corresponding l pairs of characters are gaplessly aligned and assigned to the profile P. In the example alignment of Fig 1.3, a gap interrupted the putative conserved binding site locations in the alignment of the two sequences x and y. In the annotated alignment scenario, x and y can be aligned differently such that the gap is shifted out to predict perfectly conserved TFBS hits. Fig 1.6 depicts how the binding sites are labeled as a pair-profile column in the alignment.

Thus, if the number of occurrences of the letter P in an alignment $\mathcal{A}$ of x (length m) and y (length n) is denoted by $n_{\mathrm{P}}(\mathcal{A})$, and $n_{\mathrm{S}}(\mathcal{A}), n_{\mathrm{D}}(\mathcal{A})$ and $n_{\mathrm{I}}(\mathcal{A})$ remain the respective counts of substitutions, insertions and deletions, then:

$$\mathcal{A} = (\alpha_1, \alpha_2, \ldots, \alpha_{\ell(\mathcal{A})}) \in \{\mathrm{S}, \mathrm{D}, \mathrm{I}, \mathrm{P}\}^{\ell(\mathcal{A})} \tag{1.5}$$

i:	1	2	3	4	5	6	7
x_i^*:	A	C	–	G	TATAA	T	C
y_i^*:	A	C	C	A	TATAT	–	C
α_i:	S	S	I	S	P	D	S

Figure 1.6: *Example of annotated alignment.* For the sequence pair of Figure 1.3, a possible alignment under the annotated alignment model is shown. Here the alignment is coded as $\mathcal{A} = (\text{SSISPDSS})$, where P corresponds to the pair-profile column for the profile representing the TATAA box. The putative hit location is brought forth by local rearrangement of the alignment.

is a valid alignment between x and y *iff*

$$n_{\text{S}}(\mathcal{A}) + n_{\text{D}}(\mathcal{A}) + l \times n_{\text{P}}(\mathcal{A}) = m \quad \text{and} \quad n_{\text{S}}(\mathcal{A}) + n_{\text{I}}(\mathcal{A}) + l \times n_{\text{P}}(\mathcal{A}) = n \tag{1.6}$$

where $\Sigma^* := \Sigma \cup \{-\}$ and $\ell(\mathcal{A}) = n_{\text{S}}(\mathcal{A}) + n_{\text{D}}(\mathcal{A}) + n_{\text{I}}(\mathcal{A}) + n_{\text{P}}(\mathcal{A})$, implying that a column can either be a character pair from Σ^* or a stretch l of nucleotide pairs from $\Sigma^l \times \Sigma^l$.

Scoring Scheme Clearly, the scoring scheme for annotated alignments has contributions from both the standard alignment part and the profile part. As before, each column of the alignment has an associated score. For substitution and indel columns, the standard alignment parameters $\mathtt{s}$ (substitution scoring matrix) and g (gap cost) still form the backbone of this aspect of annotated alignments.

For the pair-profile columns, two new parameters are introduced: the *profile-scoring array* (PSA) and the *profile penalty* (*pen*). The PSA of a profile P of length l assigns real-valued scores to every pair of nucleotide strings of length l, implying it is a function

$$\mathtt{PSA} : (\Sigma \times \Sigma)^l \rightarrow \mathbb{R}.$$

The profile penalty acts as a tuning parameter to decide between the two alternatives: l substitutions versus one pair-profile instance. It should be emphasized here that the PSA and *pen* need to be chosen such that a balance is maintained both *amongst the profiles* (for the case of multiple profiles) and between the *profiles and the standard alignment* such that none is unduly over- or under-represented in the optimal alignment. We elaborate upon a strategy for the derivation of these parameters based on desired error constraints in Section 3.2. For the present discussion, let us assume that an appropriate choice for each has been made.

A column representing a pair-profile instance of P is scored by summing the corresponding l entries from the PSA and subtracting the profile penalty *pen*. Hence, the column score $\mathcal{S}_c$ is defined as:

$$\mathcal{S}_c(\alpha_i, x_i^*, y_i^*) = \begin{cases} \mathtt{s}(x_i^*, y_i^*), & \text{if } \alpha_i = \text{S} \\ -\text{g}, & \text{if } \alpha_i = \text{D}, \text{I} \\ \mathtt{PSA}(x_i^*, y_j^*) - pen & \text{if } \alpha_i = \text{P and } x_i^*, y_j^* \in \Sigma^l \end{cases} \tag{1.7}$$

Analogous to the definition given in Equation (1.4), the score of the alignment is again the sum of scores of the individual columns and the optimal alignment is the one with the maximum score. Since our focus is on short conserved stretches, we then modify the standard local alignment algorithm to generate optimal alignments with pair-profiles.

1.7 Thesis structure

This thesis proposes *annotated alignments*, a novel alignment approach that combines binding site *profiles* and *alignments*. Each juncture in the thesis is discussed with particular emphasis on both aspects.

Chapter 2 formally presents the theoretical concepts needed for describing profiles and alignments. With regards to the former, the discussion is segregated into two parts: First, the construction of profiles from experimentally verified binding site sequences is formalised. Second, the use of profiles for searching putative novel binding sites is described. With regards to alignments, we discuss the standard local alignment algorithm and parameter choice. The chapter ends with a brief discourse on probabilistic models for modelling evolutionary processes.

Annotated alignments are introduced and described in detail in Chapter 3. At the onset, the extended dynamic programming algorithm is discussed. Next, a statistical framework for estimating profile-related parameters is formalised. In this context, we discuss two variations: a basic formulation that considers independence of binding sites and an extended formulation which incorporates position-specific evolutionary considerations for modelling binding site relatedness. Following this, we deviate slightly to formalise annotated alignments in a pair Hidden Markov Model framework. The chapter ends with an assessment of characteristics of annotated alignments and algorithm complexity.

In Chapter 4, we test various aspects of the proposed approach. A proof of principle for the theoretical derivations for parameter choice is provided through simulations. Through simulated as well as real data analysis, different aspects of the annotated alignment approach are compared to multi-step approaches. The influence of varying evolutionary distance and profile quality is assessed.

The thesis ends with summarizing comments and perspectives for future directions in Chapter 5.