UNIVERSITÄT LEIPZIG
Fakultät für Mathematik und Informatik
Institut für Informatik

BBQ in Tanimoto Scores

Novel Scoring Schemes for *cis*-Regulatory Module Discovery

Diplomarbeit

Leipzig, 23. Oktober 2006

vorgelegt von

Peter Menzel
geb. am: 14. Februar 1981

Studiengang Informatik

Zusammenfassung

Nach der kompletten Sequenzierung mehrerer Genome von höheren Organismen besteht eine neue Aufgabe der Molekularbiologie in der Entschlüsselung der regulatorischen Mechanismen, welche für die gezielte Expression der Gene einer Zelle verantwortlich sind. In der Kontrolle der Transkription von kodierender DNA in messenger-RNA besteht die wichtigste Möglichkeit diesen Prozess zu steuern. Die Transkription beginnt mit dem Anheften der RNA-Polymerase II an spezifische Bindungsstellen, die stromaufwärts der Transkriptionsstartseite liegen. Dieser Vorgang wird von verschiedenen Proteinen kontrolliert, die ebenfalls an die DNA binden und so die Formierung des Initiationskomplexes um die RNA-Polymerase stimulieren oder hemmen können. Diese sogenannten Transkriptionsfaktoren stehen oft in Wechselwirkungen miteinander. Deshalb finden sich Bindungsstellen von Faktoren oft gruppiert innerhalb eines kurzen Intervalls in den regulatorischen Sequenzen eines Gens. Diese Gruppen werden allgemein als *cis*-regulierende Module bezeichnet.

Die folgende Arbeit beschäftigt sich mit Algorithmen zur Erkennung solcher *cis*-regulierender Module, welche in den regulatorischen Sequenzen verschiedener Gene vorkommen. Das dargestellte Best-Barbecue-Problem formalisiert die algorithmischen Grundlagen für solch eine Erkennungsmethode. Der Algorithmus sucht nach größtmöglichen Gruppen von Bindungsstellen innerhalb eines Intervalls bestimmter Länge, die in allen Eingabesequenzen vorkommen. Das `bbq`-Programm löst dieses NP-vollständige Optimierungsproblem. Der Algorithmus findet zwar in jedem Fall die optimale Lösung gemäß des Best-Barbecue-Problems, allerdings scheitert es an typischen Instanzen. Wenn einige der Eingabesequenzen nur wenige Bindungsstellen mit den anderen Sequenzen gemeinsam haben, ist das Gesamtergebnis auf deren kleinsten gemeinsamen Nenner beschränkt. Das Ziel der vorliegenden Arbeit ist es, diese Beschränkungen durch eine Erweiterung des Suchalgorithmus zu überwinden. Eine Bewertungsfunktion soll nun die Suche nach Gruppen von Bindungsstellen erlauben, die einerseits möglichst groß sind und andererseits in möglichst vielen Sequenzen vorkommen. Es wurden drei verschiedene Bewertungsfunktionen basierend auf dem sogenannten Tanimoto-Score implementiert und in `bbq` integriert. Um die Leis-

tungsfähigkeit des erweiterten Algorithmus zu testen, wurde er an zwei verschiedenen biologisch relevanten Datensätzen evaluiert. In beiden Fällen hat der Algorithmus mit der Bewertungsfunktion gute Ergebnisse gezeigt, während der Standardalgorithmus nicht die erwarteten Ergebnisse erbracht hat. Das Programm ist als Open-Source-Software zum Download verfügbar.

Abstract

After the sequencing of the complete genomes of several higher organisms, molecular biologists face new challenges in understanding the regulatory mechanisms which control the expression of protein coding genes. The expression of a gene into a functional product is a fairly complex process consisting of several steps, with each step underlying some regulatory mechanism. The first major step in expression of protein coding genes is the transcription of the DNA into a messenger-RNA by the RNA polymerase II. For initiation of transcription the polymerase must locate and bind to a site upstream of the protein-coding region of the gene. This binding is controlled by several proteins, known as transcription factors, which also bind to the DNA and stimulate or repress the formation of the initiation complex around the RNA polymerase. DNA binding sites of transcription factors are known to occur in clusters among a small range, commonly referred to as *cis*-regulatory modules. Since transcription factors affect transcription control, genes which share the same factors or modules tend to have the same expression patterns – they are said to be co-expressed.

In this work we deal with algorithms for discovering such *cis*-regulatory modules, which occur in the regulatory regions of multiple different genes. The presented Best-Barbecue-Problem provides the formal algorithmic foundation of a discovery method, which searches in a set of genomic sequences for the largest clusters of binding sites, that are located within a window of a certain length and occur in all of the input sequences. The `bbq` program solves this NP-complete optimization problem. Although `bbq` always finds the best solution, it fails in certain instances of input data. If only some of the sequences do not share a few binding sites with the others, the overall result is restricted to the lowest common denominator. The topic of this thesis is to overcome these limitations by extending the Best-Barbecue algorithm by novel scoring schemes. The search is now aimed on finding sets of binding sites with largest possible cardinality which occur in the majority of the sequences. We implemented three scoring functions based on the so-called Tanimoto score and integrated them into the `bbq` algorithm. To evaluate the performance of the new search method, we tested `bbq` with two biologically relevant sets of regulatory sequences which have been studied before.

In both data sets the new scoring method performs well, whereas the traditional algorithm returns unsatisfactory results. Furthermore we created artificial data sets and evaluated the properties of the scoring functions. The software is freely available for download.

Contents

Abbreviations

bp	Base pairs
nt	Nucleotides
CRM	*Cis*-Regulatory Module
DNA	Desoxyribonucleic acid
RNA	Ribonucleic acid
mRNA	Messenger RNA
tRNA	Transfer RNA
snRNA	Small Nuclear RNA
ncRNA	Non-Coding RNA
PCM	Position Count Matrix
PFM	Position Frequency Matrix
PWM	Position Weight Matrix
TF	Transcription Factor
TFBS	Transcription Factor Binding Site
TSS	Translational Start Site

Chapter 1

Introduction

In the post-genomic era, a major challenge for molecular biologists is to decipher the regulatory code which governs the transcription of protein coding DNA. This thesis deals with computational approaches for discovering certain elements of transcription regulation – *cis*-regulatory modules. We study a comparative genomics based module discovery algorithm, the so-called Best-Barbecue-Problem. This first chapter introduces the foundational biological principles underlying the transcription of DNA and its regulation in eukaryotic cells. The first section covers the pathway of converting the genetic information stored in DNA into functional gene products, such as proteins (description of basic facts follows the textbook [8]). Afterwards we take a closer look at expression regulation at the level of transcription from DNA into RNA. The last section covers regulatory modules and cooperatively acting transcription factors.

1.1 From DNA to Protein

One can think of the DNA assembled in chromosomes as a storage for encoded information about structural and functional features of cellular components. Each gene in this library contains the construction plan for such a component. These genetic plans are accessed and converted into functional gene products when they are required by the cell. This so called *expression* of a gene is a fairly complex process, following multiple steps and pathways depending on the type of the gene product. The products are typically proteins, but might also be different kinds

of RNAs, such as tRNAs, microRNAs or snRNAs, which serve special purposes in cell function [16]. Although these *non-coding* RNAs perform a remarkable range of functions in all cells [10], the major ingredients in the metabolism of a cell are proteins, which act as enzymes in most metabolic pathways [26]. The so called *central dogma of biology* states that genetic information which make up the proteins are transferred from the DNA into (messenger-)RNA (by the means of transcription) and from the RNA into proteins (translation). Additionally the replication of DNA allowed for transfer from DNA into DNA and the existence of RNA viruses made at least the transfer from RNA into RNA probable. The original formulation of this dogma by Francis Crick [13] did not allow transfer from RNA into DNA nor any transfer starting at protein. Later discoveries, such as reverse transcription [54] and prions [25], softened the rules of allowed information transfers, but still the most common ways of transferring information is from DNA into RNA into protein. Thus we will concentrate on this pathway in the next sections.

The expression of genes into proteins is subdivided in three major steps. The first step is the **transcription** of a gene into *messenger-RNA* (mRNA). This messenger-RNA typically underlies some post-transcriptional modifications, also called **processing**. After this follows the **translation** of the processed mRNA into a chain of amino acids, which are in turn folded into the protein. In each step the expression flow is under control of regulatory mechanisms, which affect the expression level of the genes.

The first step in the expression chain is the transcription of a part of the DNA into a primary RNA transcript. In eukaryotes the DNA is kept in a highly organized and compressed structure, the chromatin, that allows for the very long DNA macromolecule being stored in the nucleus of the cell. The backbone of the chromatin and thus the chromosomes is formed by protein heterodimers, the *histones*. About 146 nucleotides of the DNA are winded in 1.65 loops around one single histone complex. These DNA-protein complexes are called *nucleosomes*. Two neighboured nucleosomes are connected by a short sequence of Linker-DNA with a length of about 50–70 nucleotides. By winding at higher levels, this chain of nucleosomes is twisted into chromosomes. For the purpose of accessing a certain section of the DNA, this structure has to be remodeled and unpacked.

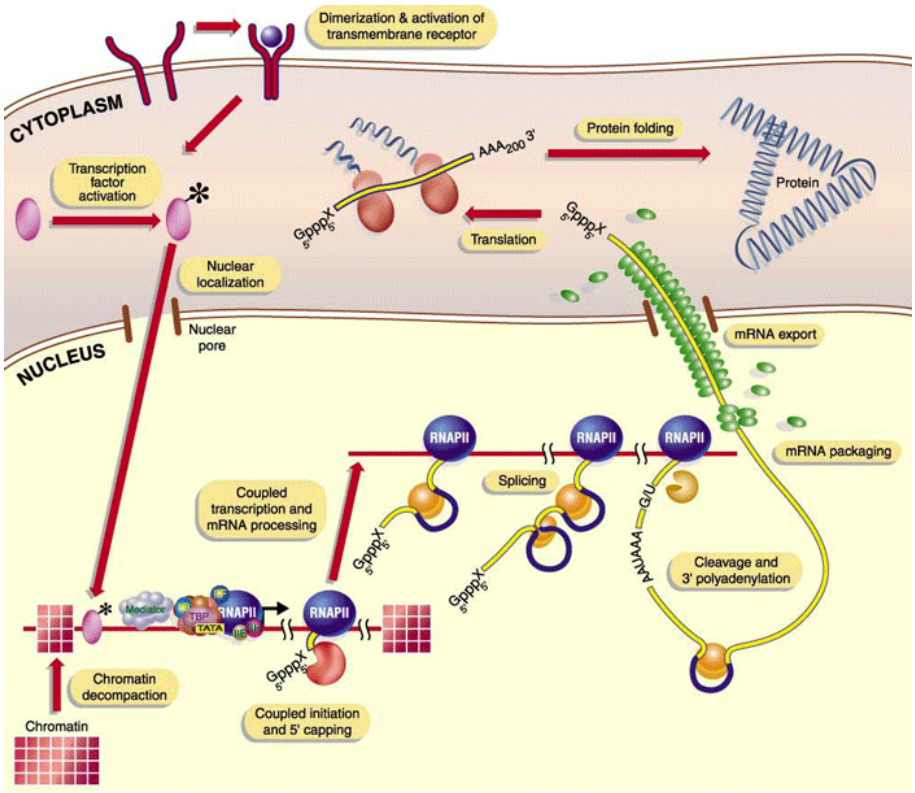


Figure 1.1: The figure shows the different steps involved in the expression pathway of a gene into a protein.

Decompressing the DNA into a relaxed configuration involves acetylation of the histones, DNA methylation and phosphorylation. After making the DNA accessible, the actual task of reading the DNA is performed by a complex conglomerate of proteins, the *RNA Polymerase*. In contrast to prokaryotes, three types of polymerases are known in eukaryotes (RNA Polymerase I, II and III or shorter Pol I, Pol II and Pol III) [43]. Each is responsible for transcribing different kinds of genes (see table 1.1). The transcription happens in three phases. With the help of several co-factors the polymerase binds to a specific point on one of the two strands of the DNA molecule. This first phase is called the **initiation** phase. Beginning at a dedicated *transcription start site (TSS)* the polymerase starts copying the DNA template strand into a RNA transcript. The transcription always proceeds in the 5'→3' direction, i.e., the DNA sequence is read in the 3'→5' direction and the complementary RNA polymer is generated from the 5' to the 3'

end. This **elongation** phase is supported by proteins associated with the polymerase complex called *helicases*, which unwind the DNA double helix before the polymerase passes by, and rewinds the helix afterwards. In a last **termination** step the polymerase stops progressing and is released from the DNA.
The most important step in this process is the initiation of transcription. The polymerase binds to some point in the region *upstream* of the transcription start site (TSS), the *promoter* region of the gene. Specific short sequences in the DNA sequence act as binding sites for components of the DNA polymerase and many proteins, which are also called *transcription factors* (TFs). The absence or presence of TFs at the promoter cause the polymerase to bind to the promoter or not. Thus the presence of *transcription factor binding sites* (TFBSs) govern the regulation of the gene expression at the level of transcription initiation. The next section covers this connection in more detail.
After the primary transcript is released by the polymerase it undergoes several modifications, which prepare the mRNA for translation. This step only occurs in eukaryotes, whereas in prokaryotes the translated mRNA is a direct copy of the gene. Therefore the next paragraphs only address eukaryotic mRNAs.
Both ends of the primary transcript are modified in two different ways. The 5'-ends of all mRNAs are complemented by a so called *cap* consisting of a 7-Methylguanin base. This cap supports the translation initiation at the ribosome and secondly helps protecting the mRNA from nucleases. Some mRNAs also receive modifications at the 3'-terminus, where the enzyme Poly(A)-polymerase appends a tail of up to 200 Adenine nucleotides. This *polyadenylation* is important for cellular transport of the transcript and additionally the poly(A)-tail helps stabilizing the transcript, just like the 5'-cap. Eukaryotic genes contain

Polymerase	Gene products
RNA Polymerase I	ribosomal RNAs
RNA Polymerase II	proteins, snRNAs
RNA Polymerase III	tRNAs, snRNAs, 5S-rRNAs, other small RNAs

Table 1.1: Three types of RNA polymerases are known in eukaryotes. Each is responsible for transcribing a subset of all genes contained in the genome.

non-coding regions, called *introns* in contrast to coding *exons*. Before translating the mRNA, these introns must be cut from the transcript and the remaining exons must be glued together. This process is called *splicing*. Several types of introns are known and different cellular mechanisms are applied during splicing. A special group of RNAs, the small nuclear RNAs (snRNAs), is known to catalyze the splicing of so called GT-AG-introns. *Alternative splicing* [36] allows the cell to combine exons differently in the final mRNA by including only a subset of all exons in the primary transcript. This feature allows for another regulatory step in the gene expression process. Another type of post-transcriptional modification is the so called *RNA-Editing*. This can be seen as some kind of revision step, in which the mRNA sequence is modified by inserting new nucleotides or deleting and editing existing nucleotides. These changes only occur at single highly specific positions in the sequence. A typical modification is the change from a normal amino acid coding triplet into a stop codon, resulting in translation into a shorter protein.

After all modifications are applied to the mRNA molecule, it is transported from the nucleus into the cytoplasm where the ribosomes are located. Similar to transcription, the translation in eukaryotes also happens in three phases. During the initiation phase, the small 40S-subunit of a ribosome recognizes the 5'-cap structure of the mRNA and binds to the outer end of the molecule. It then moves downstream the mRNA until it finds the start codon `AUG`, which encodes the first amino acid methionine. After the first tRNA carrying the methionine bound to the small subunit, the large ribosomal subunit joins this complex and starts the elongation of the amino acid chain. Several co-factors support this initiation process. During elongation the ribosome moves stepwise downstream the mRNA molecule and in each step a codon triplet is matched with a corresponding anticodon in a tRNA molecule. Each tRNA carries an amino acid at its 3'-end. The ribosome removes the amino acid from the tRNA and appends it to the carboxyl end of the growing chain. The elongation is terminated by one of three possible stop codons (`UAA`, `UGA`, `UAG`). These codons are not recognized by tRNAs, but by proteins called *release factors*. These factors trigger the dissociation of the ribosome and the synthesized polypeptide chain is released. Note that the mRNA is not altered during translation and thus it can take part in several, possibly

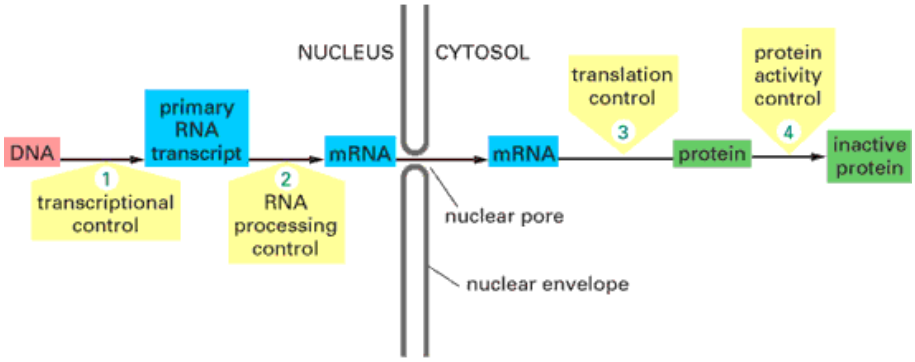


Figure 1.2: Four possible regulatory checkpoints during expression of a gene into a protein.

parallel, rounds of translation.
After synthesis the polypeptide chain may underlie some post-translational modifications and finally folds into its tertiary structure.
Each step in this complex expression pathway is regulated by a considerable number of substrates, co-factors and environmental conditions. At every checkpoint in this pathway the expression may be catalyzed, halted or even aborted. Additionally numerous other possibilities exist, for influencing expression of eukaryotic genes. See Prohaska et al.[43] for a more complete overview of major regulatory mechanisms.

1.2 Transcription Regulation by Transcription Factors

We have seen that a cell has numerous ways to control the expression pathway from a gene to its functional product. The most prominent and obviously most reasonable control point for the switch whether a gene should be expressed or not, is at the level of transcription regulation. Even if there is only a partial correlation between transcript and protein concentration in the cell[21], the selective transcription of genes by the RNA Polymerase II plays a crucial role in the overall regulatory system. Thus the initiation of the transcription, i.e., the binding of the polymerase to specific nucleotides in the promoter region of a gene,

deserves closer attention.

The polymerase is not attached to random positions in the genome, but selectively binds to certain sequence motifs upstream of a gene. Thus specific binding sites in the promoter must exist and must be detected by the polymerase. We call these binding sites which are recognized by the polymerase *core promoters*. Core promoters are located immediately near the TSS, about -50 to +50 nucleotides relative to the TSS. Additionally there are typically multiple binding sites for other specific *transcription factors (TFs)* which are located in a region more upstream of the TSS, the *proximal promoter*. Such additional TFBSs are sometimes referred as *Upstream Promoter Elements (UPEs)*. This proximal promoter ranges from around -50 to -200 nucleotides relative to the TSS. Conveniently one calls both the core and proximal promoter regions just the promoter of a gene.

In prokaryotes only one type of RNA polymerase is known, which is responsible for the transcription of every gene. The RNA polymerase of *E. coli* consists of five subunits, written as $\alpha_2\beta\beta'\sigma$, with α occurring twice. This structure is called the *holo-enzyme* and is different from the *core-enzyme* which does not contain the σ subunit. The core promoter of *E. coli* contains two different nucleotide sequences located at positions -35 and -10 respectively. During initiation of transcription both the -35-Box and the -10-Box are recognized by the σ-subunit of the holo-enzyme. After binding to the promoter the holo-enzyme covers around 60 bp of the DNA helix. Then it starts unwinding and separating the both DNA strands and the σ-subunit dissociates and the holo-enzyme is transformed in the core-enzyme. At this point the first two ribonucleotides can pair with positions +1 and +2 and the elongation of the RNA begins.

The situation in eukaryotes is much more complex. There are not only three types of polymerases, but the promoters are also richer with binding sites for numerous different TFs. Additionally further upstream control regions, called *enhancers*, contain binding sites for transcription factors, that influence the binding of the polymerase to the promoter. These enhancers can be located several kb upstream from the core promoter or even act between different chromosomes. Furthermore a group of TFs that prevent the binding of the polymerase to the DNA, called *repressors*, can bind to sites near the proximal promoter. These control regions are called *silencers*.

All sequence elements that serve as anchors for transcription factor binding are also referred to as *cis-regulatory elements*.
As discussed in [34] there is emerging evidence that the complexity of an organism arises from progressively more elaborate regulatory systems of gene expression. As shown in figure 1.3 typical regulatory regions of unicellular eukaryotes aren't by far as complex as in higher organisms, such as vertebrates. Furthermore the amount of transcription factors increases in higher organisms as well. Whereas the yeast genome encodes for $\sim$ 300 different factors, there may be as many as 3000 factors in the human genome.

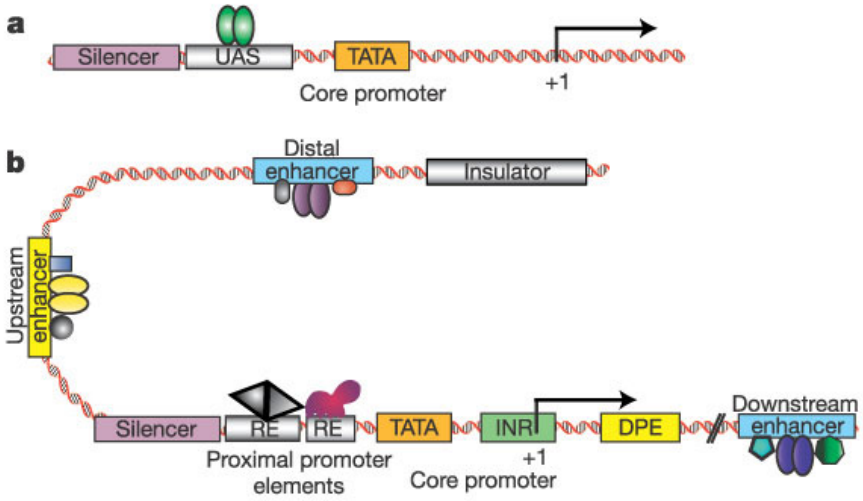


Figure 1.3: Comparison of a simple eukaryotic and a more elaborate metazoan promoter system. **(a)** This simple transcriptional regulatory unit is typical for unicellular eukaryotes. It contains a core promoter element with an upstream activator sequence (UAS) complemented by a silencer region. **(b)** Most control units of metazoans are more complex. Beside the core promoter we find additional promoters located upstream and downstream the TSS. Additionally proximal and distal enhancer modules affect transcription control. (picture from [34])

Each of the three different RNA polymerases interacts with different types of promoter sites. Thus all genes which are transcribed by the same type of polymerase must share the same or at least very similar promoter elements. Typically RNA polymerase II regulation is more elaborate than regulation of the RNA genes

transcribed by the other polymerases. There are at least three different core promoter elements yet known that can recruit the binding complex of RNA polymerase II, see e.g. [52] for a description of the RNA polymerase II core promoter. The most conserved core promoter element is the *TATA-box*, which is usually located at a position around -30bp relative to the TSS. Additionally the initiator element (*Inr*), TFIIB recognition element (*BRE*) and the downstream promoter element (*DPE*) belong to core promoter elements.

Binding of RNA polymerase II to the promoter involves several steps. First a subunit of the *TFIID* transcription factor, the *TATA Binding Protein (TBP)*, binds to the TATA-Box in the core promoter. Then the factors *TFIIA* and *TFIIB* join the initiation complex and stabilize the TBP-DNA binding. Separate from the initiation complex *TFIIF* binds to the RNA polymerase II, enabling the polymerase to recognize the initiation complex. After the polymerase joins the initiation complex around the core promoter, two additional factors, *TFIIE* and *TFIIH*, bind to the polymerase. TFIIH is especially important, because it is a helicase, which is responsible for unwinding the DNA. These TFs which are responsible for assembling the transcription complex around the RNA polymerase are also referred to as *basal* TFs.

The formation of the transcription complex at the core promoter is essential for a successful transcription initiation. But alone it is not sufficient for modulating the rate of transcription. Thus another group of transcription factor besides the intrinsic basal factors plays the major role of stimulating or repressing the rate of transcription. These factors are located in the proximal promoter and in different enhancer regions. Whereas the proximal promoter is known to contain binding sites for a subgroup of general transcription factors common to many genes, such as *SP-1* recognizing a GC-Box, *C/EBP* (CCAAT-box Enhancer Binding Protein) or *NF-Y*. The proximal promoter regions is also referred as a *Upstream Activation Sequence (UAS)*, because TFs acting in the proximal promoter typically stimulate binding and activation of the transcription complex. Besides the promoter region, each gene might be under the influence of several enhancer regions located either upstream or downstream or even in introns of the gene. A manifold of enhancers exist, with sizes ranging from 50bp up to 1.5kbp. A typical enhancer has a length of 50bp and contains around 10 binding sites for at least three different

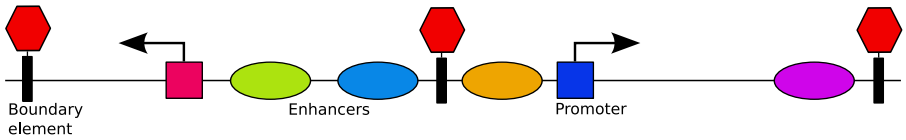


Figure 1.4: Different genomic elements affecting transcription control. Squares depict promoters containing binding sites for core promoter motifs (such as TATA) as well as proximal promoter motifs (such as SP-1). The arrow denotes the direction of transcription depending on the strand where the gene resides. Enhancers, depicted by ovals, are typically located further upstream of the TSS, but can also appear downstream of the promoter. The hexagons denote boundary elements, which separate the regulatory domains. Enhancer activity can not overcome those barriers.

TFs[14]. Enhancers can be located in a distance of several 10kb away from the TSS, but do still interact with the basal transcription machinery. This is due the spatial conformation of the DNA molecule packed in the chromatin. The tertiary structure allows for spatial vicinity, although large distances along the DNA sequence. Transcription factors binding to enhancers either act as *activators* or *repressors*. Activators are linked to the basal transcription machinery by another group of TFs, the *co-activators*. The interaction of activators with the transcription complex increases the rate of transcription. This interaction requires the DNA segment between the transcription complex and the enhancer to form a loop. Repressors on the other hand bind to enhancer regions in the role of a silencer which are located adjacent or overlapping to activator binding sites. This prevents the binding of activators to their corresponding sites, and thus prevents stimulation of transcription.
In higher eukaryotes the transcription of a single gene is typically governed by multiple autonomous enhancer modules. Each of these modules is responsible for a subset of the total gene expression pattern. They usually control the expression within a specific tissue/cell type or at a particular stage in development. This means that the enhancers are the major control units of the transcription regulation of genes specific for a cells role in the organism.
Several different models for the function of enhancers have been suggested, rang-

ing from simple "On-or-Off"-models to progressive models[5]. Whereas in the On-or-Off-model, activators bound to enhancers increase the probability of transcription, the progressive model suggests an uniform increase in the amount of transcription, dependent on the strength of the enhancer. Arnosti and Kulkarni[1] discuss the concept of "enhanceosomes" versus a so-called "Billboard"-model. The enhanceosome model proposes that the factors bound to an enhancer interact in multiple ways and form an unified nucleoprotein complex. Due to the high degree of cooperation between the factors, the overall function of the enhanceosome is more than the sum of individual contributions of each factor. The enhanceosome then actively interacts with the basal transcription complex and stimulates transcription. On the other hand, a billboard enhancer must not function as an entire cooperative unit. It rather serves as docking station for independent TFs, which independently affect the transcription. In this model the active component is the transcription complex which "reads" the TFs bound to the enhancers, which can be seen as billboards in this sense. The exact positioning of bound TFs plays a less important role here than with an enhanceosome. Both models can be seen as some kind of extremes at the scale of enhancer activity. Most cellular enhancers do not act precisely in the sense of one of two models, but in a way that features properties of both.

Because enhancers can act on large distances, it must be assured that they may only affect transcription of their designated genes. So called *boundary elements* or *insulators* are DNA regions which segment the genome into independent topological domains. This means that the expression of genes between two boundary elements is independent of regulatory elements, especially enhancers, that are over both insulators. Secondly, enhancers are capable of shielding sequence fragments from switching the chromatin state, thus building a barrier against the repressing effects of heterochromatin[9]. Figure 1.4 shows a schematic view of the organisation of enhancers, promoters and boundary elements. The precise molecular mechanisms by which insulators are able to partition the genome into independent segments are yet unknown, although several models exist, see [9] for a review.

This elaborate organization of regulatory elements and the interactions between promoters, enhancers, silencers and insulators permit a precise control of gene

expression at the level of transcription regulation.

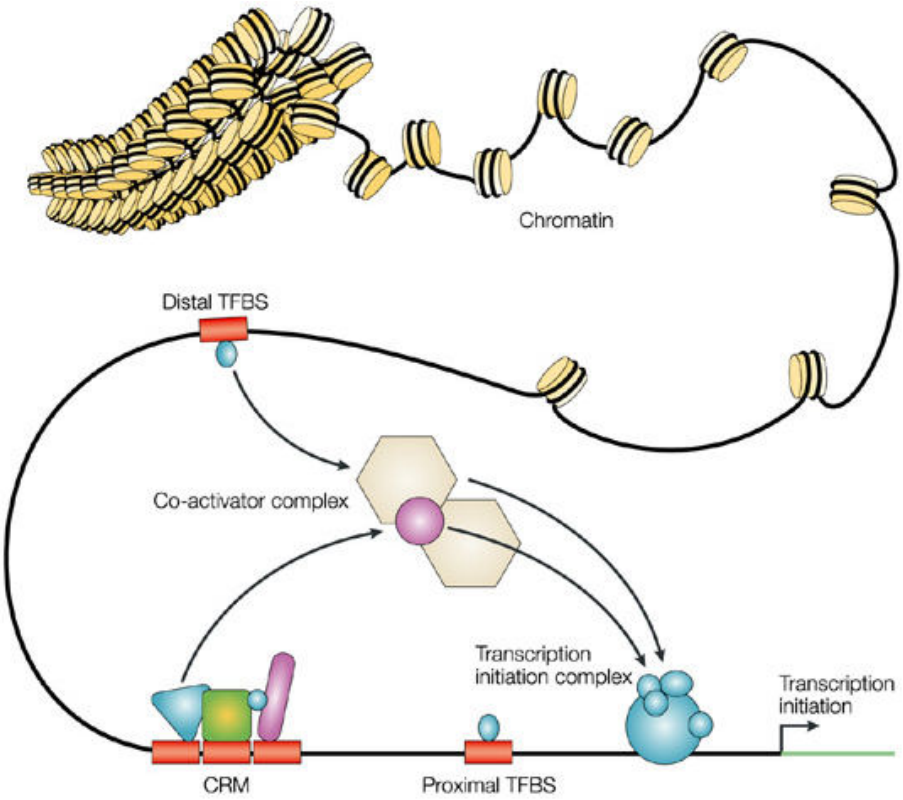


Figure 1.5: The graphic shows the influence of transcription factors bound to proximal and distal enhancer regions. The interaction between the proteins is mediated by a co-activator complex. Transcription factors within one enhancer region can be organized into cis-regulatory modules. (picture from [58]).

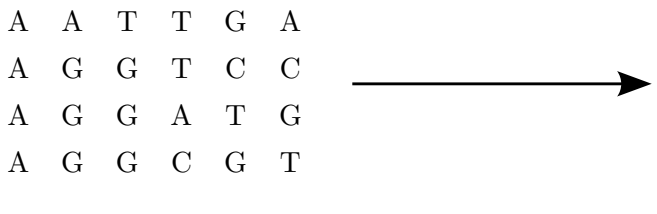
1.3 *Cis*-Regulatory Modules

As mentioned above enhancer and promoter elements typically contain binding sites for multiple different transcription factors. These clusters of binding sites are commonly referred to as *cis-regulatory modules (CRMs)*.
The interactions between TFs can be either synergistic or antagonistic[29]. In the former case simultaneous binding of factors to closely situated target sites results in cooperative activation of transcription. Factors might join to a protein-protein complex, resulting in a new protein surface with an activating domain. The independent binding to the regulatory region might also stimulate transcription activation. Additionally a number of factors are known to bend the DNA and thus permit binding of other factors. In the antagonistic case, factors interfere with each other. They might compete for overlapping sites which leads to mutually exclusive binding. Furthermore two factors might bind simultaneously, with one repressing factor masking the activation domain of an activation factor. Theses scenarios are closely related to the above mentioned billboard and enhanceosome models for enhancer activity.
A major observation on enhancer activity is that the orientation and exact position of the enhancer module on the DNA strand is not as much important for an effect on transcription regulation as the mere presence of the enhancer module at all. While order and orientation are typically conserved among homologous genes, this is not necessarily true for genes within the same organism which are regulated by the same combination of transcription factors. Such genes are said to be *co-regulated*. Additionally CRMs of co-regulated genes tend to share a significant number of common TFBSs, but must not contain precisely the same set of sites. For transcription factors to act cooperatively, their binding sites must occur in close vicinity, thus CRMs have typically a length of only several hundred nucleotides.
This hypothesis of the organisation of *cis*-regulatory modules is the foundation of the best-barbecue approach discussed in chapter 3. The `bbq` algorithm searches several genomic sequences for clusters of binding sites simultaneously observed in a window of a specified length. See chapter 3 for a formal problem definition.

1.4 Representation of Transcription Factor Binding Sites

We have seen that numerous different transcription factors exist, and that each transcription factor recognizes binding sites in promoters or enhancers of several distinct genes. Since studying the first experimentally determined nucleotide sequences of the *E. coli* -10-Box[42], it became clear that there are considerable differences between any two sites. Despite this variability, it was observed that all sites are at least similar to a common *consensus sequence*, making it possible to find features occurring in most sites. One can define the consensus sequence of a set of sites as the sequence that matches all of the sites closely, but not necessarily exactly. It is typically represented by a string over the DNA alphabet extended by letters of the IUPAC alphabet, thus allowing for ambiguities.
The characterization of TFBSs by a consensus sequence means a major distinction between regulatory sites and sites recognized by restriction enzymes. Such sites can be easily written as a sequence over the DNA alphabet (e.g. `GGATCC` for *BamH1*), sometimes allowing for ambiguity at certain positions. Restriction enzymes will recognize exactly these sequences in DNA and only matching sites will be cut. This substantial difference between variable regulatory sites and fixed restriction sites makes sense considering their biological context. Because restriction enzymes act as a defense mechanism against viral infection and attack foreign DNA, they need an *all-or-none* activity and must be so specific to not cut the cell's own DNA. On the other hand, regulatory systems take advantage of the variability of TFBSs to better control gene expression. Promoter activity can therefore be controlled by having different levels of binding affinity by TFs to their sites [53].
It soon became obvious that the representation of a collection of sites in the form of a consensus sequence was not powerful enough. Due to their simple design, consensus sequences lack major information contained in the sequences used to derive the consensus model. A major drawback is the loss of the occurrence frequencies of each nucleotide position. Consensus sequences have been shown to provide either a good sensitivity or a good precision, if they are used to predict occurrences of new sites. The sensitivity of a representational model of binding

sites denotes the ability to detect a true signal instance at a given level of significance. On the other hand, the precision is a measure for qualifying the rate of false positive hits. The less false positive signals are detected, the better is the precision. For any method for searching TFBSs based on a representational model of its structure, a high sensitivity and and high precision is desirable [53]. A more sophisticated method of representing TFBSs is in the manner of a so called *Position Count Matrix (PCM)*, sometimes also called *Position Specific Score Matrix (PSSM)*, *Position Frequency Matrix* or just *profile* of a motif. The goal of this representation model is to summarize an alignment of motifs in a concise format without much information loss. Furthermore the special properties of TFBSs must be considered, namely the imbalance of the conservation and nucleotide frequencies at different sequence positions. The PCM approach meets these requirements optimally.
Starting from a multiple alignment, the corresponding position count matrix M is constructed by counting the frequencies of each base in each position at all sites in the alignment. An matrix element m_ij yields the number of times the letter i occurs at position j in the alignment, figure 1.6 shows an example. Furthermore PCMs can be normalized, such that each row contains the occurrence probabilities of the base. In this case the values in each column add up to 1.0. Typically these count matrices do not allow for gap positions. Furthermore the



M	1	2	3	4	5	6
A	4	1	0	1	0	1
C	0	0	0	1	1	1
G	0	3	3	0	2	1
T	0	0	1	2	1	1

Figure 1.6: A multiple sequence alignment can easily be transformed into a position count matrix. This example (from [23]) shows the PCM representation (right) of four nucleotide sequences of the alignment (left). The consensus sequence of the alignment would be `AGGTGN`.

process of transforming an multiple alignment into a PCM is unidirectional. The "reverse engineering" of an alignment based on its matrix profile is not possible. Note that in PCMs each column is seen independently from its neighbors. Thus

PCMs lack the "horizontal" information which are inherent properties of multiple sequence alignments. Therefore most TF motif databases store the PCM as well as the multiple sequence alignment, allowing for tools to access full information on binding site motifs.

Position count matrices are typically only an intermediate step during the more elaborate construction of *Position Weight Matrices (PWM).* In that representation model, each matrix entry denotes a weight instead of a simple counter. These weights are used when comparing a test sequence to the matrix profile. The *match score* of a test sequence is then obtained by first aligning the sequence to the matrix and summing the weights in the respective matrix row over all positions. Several methods have been proposed to determine the weights in the profile, they are typically based on the information content, incorporate a background distribution or are determined experimentally[53]. PWMs are the yet best known representational model for TFBSs. It has been shown that the scores of such a matrix, generated from a well-known reference binding site selection, approximate the binding energy of the TF and provide a high specificity and sensitivity when searching for novel occurrences of the binding site[17].

Figure 1.7 shows an example of calculating the weights.

M	1	2	3	4	5	6
A	4	1	0	1	0	1
C	0	0	0	1	1	1
G	0	3	3	0	2	1
T	0	0	1	2	1	1

→

M'	1	2	3	4	5	6
A	1.20	0	−1.60	0.00	−1.60	0.00
C	−1.60	−1.60	−1.60	0.00	0.00	0.00
G	−1.60	0.96	0.96	−1.60	0.59	0.00
T	−1.60	−1.60	0	0.59	0.00	0.00

Figure 1.7: The position count matrix from our example is converted into a position weight matrix. The weights are calculated using the formula $m'_{ij} = \ln \frac{(m_{ij}+p_i)/(N+1)}{p_i}$. N denote the total number of sequences (4 in our example), p_i is the *a priori* probability of a letter (0.25 in our example). These probabilities can be modified according to a given background distribution.

Additionally PWMs can be obtained from statistical multiple alignments, which are calculated by tools like MEME or AlignAce (see section 2.2).

Although PWMs are a powerful method for modelling TFBSs, the quality of a such a profile strongly depends on a careful selection of the underlying sequences.

Figure 1.8: A sequence logo constructed from 350 aligned -10 sites of the *E. coli* promoter.

The power of a PWM is often measured by calculating its information content or other statistical methods. An in-depth analysis of the profile quality in the TRANSFAC database (see section 2.1.1) was performed by Rahmann et al.[46]. They developed a measure based on both sensitivity and selectivity of a profile and scored every TRANSFAC profile considering three different background distributions. It has been shown that, only a small fraction of the profiles has significant sensitivity and selectivity to detect signals with 5% error rate.
Both the terms Position Count Matrix and Position Weight Matrix are often used synonymously in literature and practical bioinformatics applications or databases. Throughout the remaining chapters we will just distinguish between Position Count Matrices, containing only nucleotide frequency counts at each column of the matrix, and Position Weight Matrices, which contain weighted values at each matrix position.
A nice visualization of a TFBS's consensus sequence is accomplished with *sequence logos*[49]. Figure 1.8 shows an example sequence logo of the alignment of 350 *E. coli* `TATAAT` (-10-Box) sites. The graphic is generated from a multiple alignment of similar sequences. This graphical representation yields different information in one picture: The order of predominance in each column is determined by the position of each letter in the stack, with the most frequent letter at the top. The general consensus sequence can therefore be found by just reading the letters at the head of each column. The overall height of a column denotes

its information content in bits, whereas the different heights of letters within a single column depict their relative frequencies between each other. The sequence logo approach is not restricted to genomic sequences, but is also applicable to amino acid sequence alignments. A web service is available for easy generation and customization of own logos[1].

Although PWMs are used widely for prediction of TFBSs and elaborate software tools are available for construction of such motif profiles, they lack the ability to model certain biologically important circumstances, such as cooperativity between factors and properties of the flanking region of a binding site, e.g. the GC-content which is important for the melting temperature of a DNA segment. One approach to solve this problem might be to model sequence features of higher order as random variables in a Bayesian network [44].

Besides the distinction of basal TFs and modulatory TFs found in enhancers and proximal promoter regions, transcription factors can be classified by various methods. The most obvious method is a classification by the cellular or tissue-specific function of the gene with which the factor interacts. E. Wingender[60] introduced a classification scheme for eukaryotic transcription factors based on structural properties, primarily concerning the DNA-binding domain of the protein.

[1] `http://weblogo.berkeley.edu`

Chapter 2

Tools and Databases

Before introducing the Best-Barbecue-Problem in the following chapter, we give a brief overview about some resources which are directly related to regulatory module discovery. These are foremost databases on regulatory regions associated with known genes. Note that in "real life", the distinction between the different regulatory regions surrounding a gene becomes fluent, and one conveniently calls all non-coding regions upstream and downstream of a gene just promoter regions. Additionally there exist databases which store collections of TFBS motifs. These are typically represented by a PCM. Typically both types of databases maintain hyperlinks between each other, associating genes with TFs and vice verse. Secondly, we introduce some existing approaches for creating, searching and clustering TF motifs. Some of these approaches are similar to the `bbq` program, but differ in certain features and algorithmic foundations.

2.1 Promoter and Transcription Factor Motif Databases

With the advent of the Internet [4], researchers and companies developed numerous biological databases, which are available to a world wide audience. Each database serves a special purpose, but nearly all databases are connected to other databases, because most data only make sense in a certain context, which is provided by links to data stored elsewhere. There also exist many meta databases,

which only integrate data from different sources into one database, without providing unique data. The number of available databases increases every year, thus keeping track of the available sources is becoming a primary concern. The European Bioinformatics Institute maintains a list of more than 500 different databases categorized by their main application field[1]. The NAR Molecular Biology Database Collection [19] lists more than 850 entries.
In the next sections, some of the biological databases which provide input data for the `bbq` tool are introduced.

2.1.1 TRANSFAC

TRANSFAC is by far the largest database on eukaryotic transcription factors and their binding site profiles. Since its start in 1988[59] the database was continually updated with more data sets and the types of stored information were expanded as well. Now TRANSFAC consists of multiple modules making up the TRANSFAC system on gene expression regulation[61]. The major component still contains the information on transcription factors and their corresponding binding sites. Binding site profiles are represented by PCMs, and secondly the alignment of the source sequences is available.
The complete TRANSFAC database is only available under a commercial license. These licensed versions are frequently updated and are accompanied by user interfaces and supporting tools. An older release of TRANSFAC is publicly available[2].

2.1.2 JASPAR

JASPAR[47] is an **open-access** database for transcription factor binding site profiles of multicellular eukaryotes. Currently JASPAR comes in three flavours:

1. The **JASPAR CORE** database contains high-quality profiles from published articles. These profiles are constructed from a collection of target sequences, which are experimentally determined.

[1] `http://www.infobiogen.fr/services/dbcat/`
[2] `http://www.gene-regulation.de`

2. **JASPAR FAM** consists of so called *familial profiles*, which are not specific to a certain factor but rather to a set of TFs which are related in structure. Therefore these models describe the shared binding properties of a class of TFs. FAM models are a good choice, when searching large genomic sequences without prior knowledge of contained TFBSs.

3. **JASPAR PHYLOFACTS** contains profiles that are derived from phylogenetically conserved upstream elements[62]. Promoters of human genes were aligned with promoters of orthologous genes of mouse, rat and dog. The consensus sequence is searched for motifs of length 6 to 18, containing only letters over the alphabet {A,C,G,T,R,Y,K,M,S,W,N}. Motifs are regarded as conserved when they appear in the human as well as the other three genomes. A conservation rate is defined as the number of motif occurrences found in all species divided by the number of occurrences in the human genome only. Only motifs with a conservation rate significantly higher than an estimated conservation rate from random motifs are retained in the result set. The resulting motifs are compared against the CORE matrices and similar entries are annotated. Thus PHYLOFACTS contains both known and yet undefined motifs and is best used together with the CORE matrices.

All data from JASPAR can be downloaded in a flat-file format or as a MySQL dump. Additionally a tight integration with the Perl TFBS module[33] offers application developers an easy API to the JASPAR database. The key differences to TRANSFAC are JASPAR's open-access license, the non-redundancy of the binding profiles and the easy access through the JASPAR API. Furthermore JASPAR is expanding to an even more open community resource on TFs, explicitly inviting researchers to contribute new TFBS models[56]. Therefore JASPAR provides a convenient platform where researchers can make their results available to the research community.

2.1.3 EPD

The Eukaryotic Promoter Database[11] contains a non-redundant collection of eukaryotic Polymerase II promoters. Basically EPD contains a list of pointers

to transcription initiation sites (TSS) in the EMBL[27] sequence database. This means, that EPD does not store separate sequence data, because they are included in EMBL anyway.
This concept actively avoids data redundancy and therefore reduces maintenance overhead. Additionally an arbitrary choice of the promoter region's size is avoided. Therefore the user can choose the borders by his requirements. Obviously the pointer based design requires co-ordinated update procedures by both databases, resulting in a stronger interconnection compared to mere cross-referencing between entries.
An EPD entry corresponds to a single (TSS, species) combination and only one entry per (TSS, species) may exist. This non-redundancy policy requires all information about the same TSS in a genome to be aggregated in a single entry. Only promoters which fulfill certain conditions are included in EPD. The promoter must be active and biological functional in a higher eukaryote and must be recognized by the RNA Polymerase II. The TSS must be mapped accurately and its DNA sequence must be available in the EMBL database. These criteria ensure a high quality, which is one of EPD's main objectives.
Entries include several information beyond the mere position of the TSS: ID and Accession number, description (typically including a gene name), experimental evidence, promoter classifications, links to promoters in close proximity, information on regulatory properties, references to other databases and bibliographic references.
The EPD is one of the oldest promoter databases, starting as a compilation of promoter sites retrieved from experimental data presented in research publications. A big effort was made to maintain accuracy of all entries. A complete coverage of all promoters was never considered as realistic objective. Twenty years later, this situation changed. Due to new experimental and computational methods complete genomes can be scanned for promoters. Therefore EPD redefined its policies reflecting these new developments[48]. One new objective is the complete coverage of all promoters for three model organisms (human, fly, rice). Additionally a new class of promoter entry types has been created: *preliminary entries*. The inclusion requirements for such entries are less stringent than for standard entries. Typically, preliminary entries are generated automatically from

mass genome annotations.
The EPD can be accessed by a web interface[3] and by FTP.

2.1.4 COMPEL

COMPEL[29] is a database on composite regulatory elements. A composite regulatory element (CE) is defined to be a pair of two different TFBSs which are closely situated in a promoter region, complemented by the corresponding TFs, the protein-protein interaction between them and the expression patterns provided by the combinatorial regulation.
CEs are seen as minimal functional units of transcription regulation, where different kinds of TF interaction contribute to specific regulation patterns. Two main types of interactions between TFs are distinguished. TFs act either synergistic or antagonistic. Furthermore they are classified by the structure of the DNA-binding domain. Composite elements are collected from literature and must be supported by experimental evidence. The database consists of three major tables, yielding entries for composite elements, interactions and literature references. Entries in the CE table include Accession number, Gene Identifier, organism, sequence, homology to other CEs, type of CE and the experimental evidence. CE entries are linked to specific entries in the interactions table, which contains information on the factors binding to a CE, their names, positions, DNA-binding domains and the cell types in which this factor occurs. Most CEs are linked to the EMBL database, the corresponding genes are linked to TRRD and references are linked to MEDLINE. The TFs binding to CEs are linked to TRANSFAC.
COMPEL 3.0 is freely available for non-commercial users and is distributed in flat-file format. Additionally a web interface[4] is available for browsing and searching the database. A more recent version, called TRANSCompel, containing more CEs is available only under a commercial license. TRANSCompel is closely connected to the TRANSFAC database.

[3] `http://www.epd.isb-sib.ch/`
[4] `http://compel.bionet.nsc.ru/`

2.1.5 TRRD

The Transcription Regulatory Regions Database [31] collects information about whole regulatory regions specific for single eukaryotic genes. A basic entry corresponds to a gene. Each gene is associated with regulatory regions (5' and 3' regions, introns and exons), which containd in turn one or more regulatory units (enhancers, promoters or silencers). Finally regulatory units contain TFBSs. See figure 2.1 for an overview of this hierarchical organization. Each regulatory region is associated with a description of its expression pattern, which contains the conditions under which a gene is expressed, e.g. cell cycle stage, developmental stage, cell type (tissue, organ) or the influence of external signals. This allows for clustering genes according to various criteria depending on the expression profile. Only experimentally confirmed information are included in TRRD. The data are collected from publications and are annotated according to the type of experiment conducted. The database is available in flat file format and can furthermore be accessed by a web interface[5].

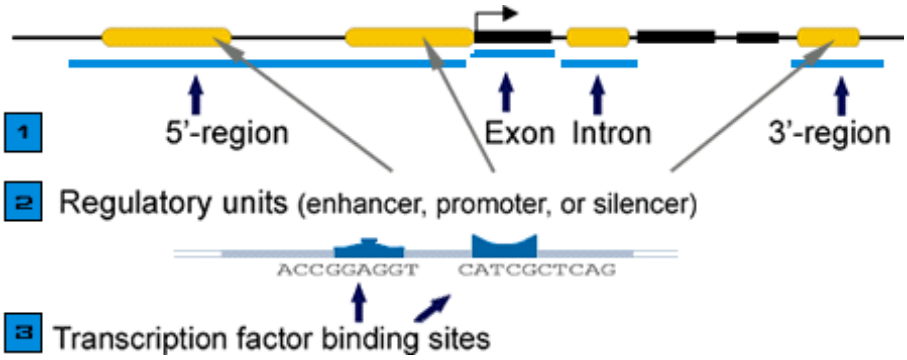


Figure 2.1: The TRRD entries are organized hierarchially into 3 layers, which denote the organizational levels of regulatory elements.

2.1.6 Specialized Promoter Databases

Additionally to the general databases, there also exist a handful smaller more specialized databases, which focus on promoters of genes belonging to a specific group of cell types or to specific metabolic roles. Furthermore some databases

[5] http://www.bionet.nsc.ru/trrd/

only cover a particular phylogenetic range, typically a single species. Such specialized databases aim at supporting research in their respective context, due to the aggregation of contextual data and research results.

FlyReg

FlyReg is a database of DNase I footprints for the fruitfly, *D. melanogaster* [3]. The footprint sequences are derived from systematic literature reviews. Currently the collection contains data from 201 primary references. The 1367 annotated footprints belong to binding sites for 87 transcription factors in 101 target genes. Genes are linked to corresponding genome positions in FlyBase [20] and the UCSC genome browser and the exact postions of the binding sites are annotated. FlyReg can be browsed online[6] and is available for download in GFF, Fasta or SQL format.

AGRIS

The Arabidopsis Gene Regulatory Information Server [15] is a part of the *Arabidopsis thaliana* Functional Genomics Project, which has to goal to determine the regulatory networks controlling the expression of all *A. thaliana* genes. The AGRIS server consists of AtTFDB and AtcisDB. AtTFDB is a collection of currently 1770 TF sequences grouped into 50 families which have been observed in *A. thaliana*. ATcisDB contains promoter sequences of all 27975 annotated *Arabidopsis* genes. Currently only 5'-regions are included. Occurences of binding sites in the promoter sequences are mapped to AtTFDB with an annotation whether they are experimentally known or are computationally predicted. The database can be browsed online[7] and downloads are available for registered users.

HemoPDB

The Hematopoiesis Promoter Database [41] focuses on transcription regulation during hematopoietic development, i.e. the formation and development of blood cells. Studying the regulation of these developmental programs is important

[6] `http://www.flyreg.org/`

[7] `http://arabidopsis.med.ohio-state.edu/`

for understanding the abnormalities which may occur during proliferation and differentiation of blood cells. These may cause severe blood diseases, such as leukemia. HemoPDB's authors manually collect data from literature by periodic reviews. Additional data on TFs is collected from public databases by an automated data-mining pipeline. The database stores *cis*-regulatory regions with associated relative positions to the TSS of the corresponding gene. TFs are linked to positions in promoter sequences, complemented by binding site information. HemoPDB currently contains 246 promoter sequences and binding site motifs for 187 TFs. The database can be accessed by a web interface[8].

SCPD

The *Saccharomyces cerevisiae* Promoter Database [63] yields information on promoter regions and TFBSs in the yeast genome. The relatively simple database structure provides entries for genes, promoter regions and putative TFBSs. Consensus patterns for TFBS motifs are available as PWMs. SCPD is accessible by a simple website[9], which provides basic means for searching an browsing, although not all functions are currently available.

LSPD

The Liver Specific Promoter Database (LSPD) stores promoter regions of genes expressed in especially in liver cells. Currently about 300 promoter regions of liver genes from human and rodents are contained. The web interface[10] provides basic means of browsing, but is not well curated.

2.1.7 Orthologous Promoter Databases

The search for TFBSs must not be restricted to regulatory regions of genes belonging to a single species. While comparative genomics are widely used for discovering genes, the regulatory regions of orthologous genes yield information about different expression patters among different species. Recently several databases

[8] `http://bioinformatics.med.ohio-state.edu/HemoPDB/`

[9] `http://rulai.cshl.edu/SCPD/`

[10] `http://rulai.cshl.edu/LSPD/`

focusing on storing information on orthologous promoters were established by several groups. The OMGProm [39] database started in 2005 with collecting alignments of orthologous promoters of mice and men. The long term aim is to extend this collection with orthologous promoters of other mammals, thus building an Orthologous Mammalian Gene database (OMGProm). OMGProm can be accessed by a web interface[11]. Another recently started project is DoOP (Database of Orthologous Promoters)[12].

2.2 Discovery of TFBSs and CRMs

In the last years numerous algorithms and tools have been developed to computationally find TFBS motifs and clusters of TFBSs. These tools can be divided into different categories representing the classes of the biological problems they try solve. Tools like Consensus [23], AlignACE [24] or MEME (see below) search in a given set of sequences for overrepresented motifs. They then construct an alignment of these motifs and finally a PWM can be obtained. These algorithms require no prior knowledge of motifs contained in the input sequences. See [55] for a recent overview (including benchmarks) of tools belonging to this class of programs. Given a set of TFBS motifs, possibly represented by a PWM, one usually wants to find occurrences of these motifs in genomic sequences. This class of programs originate typically from the first string search and local alignment algorithms. Current examples are MATCH, ConSite [32] (which uses phylogenetic footprints in orthologous sequences), rVista or the TFBS Perl module. Note that these algorithms rely on a prior knowledge of the binding site motifs. Another challenge is the discovery of clusters of TFBSs. Several algorithms have been developed, among them Cister, CoBind, Creme and the `bbq` tool which is the topic of this work. In the next sections, some of the most widely used tools are introduced.

[11] `http://bioinformatics.med.ohio-state.edu/OMGProm/`
[12] `http://doop.abc.hu/`

2.2.1 MATCH

The MATCH algorithm[28] searches DNA sequences for occurrences of known TFBSs. Originally it was developed as a search tool accompanying the TRANSFAC database, replacing MatInspector[45]. The `bbq` program applies the MATCH algorithm to find TFBS motifs in the genomic sequences. Hence we explain it in more detail in this section.

MATCH uses two score values which measure the quality of a match between the sequence and a position count matrix: the matrix similarity score (MSS) and the core similarity score (CSS). Both values are computed with the same formula and range from 0.0 and 1.0, where 1.0 denotes an exact match. While the MSS is calculated using all positions of the matrix, the CSS is calculated using only the five core positions of the matrix. The core of a matrix is defined as the first sequence of consecutive positions that are most conserved. The matrix similarity score for a subsequence of length L is calculated in the following way:

$$MSS = \frac{Current - Min}{Max - Min} \tag{2.1}$$

where

$$Current := \sum_{i=1}^{L} I(i) f_{i,b_i} \tag{2.2}$$

with f_{i,b_i} denoting the absolute frequencies of nucleotide B in column i of the matrix. Min and Max denote the fragment's score for the least and most conserved positions:

$$Min := \sum_{i=1}^{L} I(i) f_i^{min} \tag{2.3}$$

$$Max := \sum_{i=1}^{L} I(i) f_i^{max} \tag{2.4}$$

where f_i^{min} and f_i^{max} are the frequencies of the least/most conserved nucleotides at position i. The information vector I describes the information content of each position in the matrix:

$$I(i) = \sum_{B \in \{A,C,G,T\}} f_{i,B} \ln(4 f_{i,B}) \tag{2.5}$$

The more conserved the position i is, the higher is its information content. Equation (2.1) ensures that the score ranges from 0.0 to 1.0 and 1.0 is reached when the given subsequence equals at all positions the most conserved nucleotides in the PCM, resulting in $Current = Max$. The core similarity score is calculated the same way, but with $L = 5$ and only the five most conserved nucleotides of the subsequence are used.
Recently the MATCH algorithm received an upgrade and was extended to **P-MATCH**[12]. P-MATCH extends the PWM approach by pattern matching, thus providing higher accuracy of recognition than only PCM matching. Again, P-MATCH is closely related to the TRANSFAC database, which contains for each TF an alignment of binding sites and a PCM.

2.2.2 MEME

MEME[2] belongs to a group of programs which are designed to discover common genomic elements in a given set of sequences, without prior knowledge of TFBSs contained in these sequences. MEME takes as input a set of genomic sequences and searches for similar DNA motifs occurring in as much sequences as possible. The underlying "MM" algorithm is based on a maximum-likelihood estimation of the parameters of a finite mixture model. This model contains two components: a background model and a motif model. In the background model, each position in a subsequence which is not part of a motif is generated independently, whereas in the motif model, each position of a subsequence belonging to a motif is generated by a independent random variable describing a multinomial trial with parameters denoting the occurrence probabilities for each nucleotide, like in a PWM. The overall model which generates the sequences chooses between the motif model (with probability p) and the background model (with probability $1 - p$). The maximum-likelihood estimation searches for model parameters which maximize the likelihood that the given data are produced, i.e. the motifs produced by the motif model and and not by the background model. The output contains the motifs found including the expectation value, their positions in the sequence set and a PCM for each motif.

MEME is is accessible online[13] through a HTML web interface. The submit form lets you upload a sequence set and control some parameters, like the expected motif distribution (one, zero or one, any motif occurrences per sequence), the minimum and maximum motif length and number of motifs to find. Additionally MEME can be downloaded.

2.2.3 rVista 2.0

The rVista tool[35] combines pattern recognition with comparative sequence analysis to search for conserved occurrences of TFBSs. The input for rVista is essentially a multiple sequence alignment, which can be imported from different sources. The processing involves 4 steps: First rVista searches for TFBS matches in each sequence in the multiple alignments using PWMs from the TRANSFAC repository or user submitted profiles. In the next step pairs of locally aligned TFBS are identified. These occurrences are then filtered by the conservation level of a sliding window around 20bp around the binding site. The remaining TFBSs can be searched for modules consisting of multiple different TFBSs. rVista 2.0 provides a new web based interface[14] and other ways of submitting data. It accepts either user submitted blastz alignments or two input sequences in FASTA format which are automatically aligned with the zPicture program. Additionally the ECR Browser and the GALA database provide automatic submitting of alignment and annotations to rVista.

2.2.4 CREME

CREME (Cis-REgulatory-Module-Explorer)[51] constitutes a framework for finding and scoring clusters of TFBSs in evolutionary conserved promoter regions. The creme web server[50] provides a user interface to search putative CRMs in a set of human promoter sequences which tend to be co-expressed (or co-regulated). The algorithm relies on a precomputed database of putative TFBSs across the promoter regions of all known human genes. The TRANSFAC database serves as repository of vertebrate PWMs. The ECR Browser[38] provides alignments

[13] `http://meme.sdsc.edu`

[14] `http://rvista.dcode.org`

from human/mouse and human/rat genomes. The rVista 2.0 tool was applied to search for PWM matches in highly conserved regions of the alignments. Only matches with a similarity score above 0.8 were considered. Promoters are selected by entering a list of accession numbers/LocusLink ID numbers in CREME's web interface. Additionally the PWM similarity threshold, the maximum module length and the maximum number of TFs per module must be chosen for each run. After all is set up, the computation follows three steps:

1. Collect a set of non-redundant TFBSs found in the selected promoters, which scores above the chosen similarity threshold.

2. Enumerate all combinations of these PWMs which occur in an interval of the chosen window size.

3. These combinations are evaluated statistically, based on their frequency in the promoter set as well as the similarity of the contained PWMs.

The resulting putative CRMs are visualized and presented in a HTML page.

2.2.5 Genome-wide Module Search

At time of writing, Blanchette et al.[6] presented another approach for a genome-wide prediction of regulatory modules in the human genome. Their algorithm again relies on the assumption that CRMs contain a small set of phylogenetically conserved TFBSs. Basically a sliding window is sweeped over a MULTIz[7] alignment of human, mouse and rat genomes. The window size is at most 2000bp. The TRANSFAC database served as repository for PWMs of 229 vertebrate TF families. Matching of the PWMs was conducted by a log-likelihood ratio score with a third-order Markov background model adapted to the local GC content. Each match gets a hit score, where the scoring method favors matches with which occur simultaneously in all three species. After identifying putative TFBSs in the genome, the sliding window method is used to search for regions which contain a significant number of up to 5 different TFs. A module score and P-value is assigned to each region, depending on amount and density of TFBSs in the module. More than 118,000 putative CRMs have been identified in the examined parts of the human genome. Due to the genome-wide range of this procedure, one can

obtain a global view of the "regulatory landscape" of the underlying genome and derive some statistics about the distribution of TFBS and CRMs:

- The module density varies widely across the genome. In the average one can find four modules per 100kb.
- Locations of putative CRMs are significantly correlated with a gene's TSS and many putative CRMs are located downstream of the TSS (introns, 5'-UTR).
- Many modules can be found at regions of transcription termination and the 3'-UTR.
- Genes in locations with a high density of predicted modules are mostly responsible for development, regulation of transcription, morphogenesis, organogenesis ans neurogenesis.

The complete set of predicted CRMs is available in the public database `PReMod`[15].

2.2.6 Genetic Algorithm

Perco et.al. [40] applied a genetic algorithm to find clusters of binding sites occurring in multiple sequences. The problem they want to solve is similar to the `bbq` approach, which is discussed in the next chapter. Starting with a set of promoter sequences, the MATCH program is used to search for matches of TFBSs in each sequence and a matrix M with columns corresponding to the TFs and rows corresponding to the promoter sequences is created. Then a population of individuals, each holding a bitstring denoting a cluster of TFs by setting the corresponding bits to 1. In each iteration step all individuals are scored to M with a Tanimoto score function (see section 4.1). The best scored individuals are recombined and randomly mutated to construct a child population which is in turn evaluated in the next iteration. This procedure is continued until some convergence criteria is met, typically a fixed number of generations or constant fitness of the best individual.

[15] `http://genomequebec.mcgill.ca/PReMod`

Chapter 3

The Best-Barbecue-Problem

In the following sections we formalize the problem of finding a maximum set of short cis-regulatory elements occurring clustered on several large genomic sequences within a window of fixed length. We call this problem the *Best-Barbecue-Problem* [37]. In the first sections we formalize the problem and then provide two algorithms to solve it.

3.1 L-Occurrences and Interval Arrangements

Let Σ be a finite alphabet, usually $\Sigma = \{A, T, G, C\}$ in our biological setting. An interval between the integers a and b is denoted by $[a : b] := \{a, a + 1, ..., b\}$. An integer x is said to *stab* an interval $[a : b]$ if $x \in [a : b]$. To describe occurrences of fragments $s \in \Sigma^+$ in genomic sequences $S \in \Sigma^+$, we say that s occurs in S at some position x iff $1 \leq x \leq x + [s] - 1 \leq [S]$ and $s = S[x, x + |s| - 1]$, where $[S]$ denotes the length of a string S and $S[a, b]$ denotes the subsequence of S between the indices a and b. Clusters of fragment occurrences in a string S are defined by introducing a *cluster length* L. We say that a set of fragment occurrences are L-clustered, if these occurrences are contained within an interval of length L.

Definition 1 *Let $S = \{s_1, ..., s_m\}$ be a set of fragment sequences, $T \in \Sigma^+$ a genomic sequence, $L \in N$ be the cluster length. We say that $A \subseteq S$ is an L-occurrence in T if there is an index i_s for each $s \in A$ such that*

1. *s occurs in T at position i_s for each $s \in A$ and*

2. $|i_s + [s] - i_t| \leq L$ *for all* $s, t \in A$.

To find maximum clusters of fragments, we search for L-occurrences with maximum cardinality. In our biological setting, we want to find maximum-cardinality L-occurrences which occur simultaneously in several genomic sequences $T_1, ..., T_K$. To solve this more complex problem, we first introduce a geometric view on the problem of finding L-occurrences in a single sequence and extend this problem to the case with multiple genomic sequences. Now we extend the occurrences of

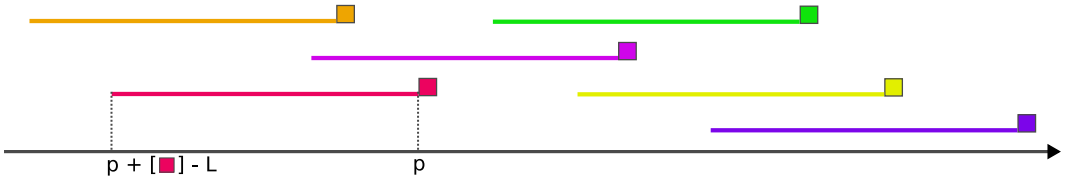


Figure 3.1: Construction of colored intervals. The black arrow denotes a genomic sequence, colored boxes represent occurrences of transcription factor binding sites. Each site s is associated with a colored interval of length $L - [s]$.

fragments to occurrences of *colored intervals* where the colors are associated to the different types of fragments. A colored interval is a pair $([a : b], c)$, denoting that $[a : b]$ has color c. First each fragment $s \in S$ is identified with an unique color c_s by a bijective mapping $c : S \rightarrow [1 : m]$. Then we associate each occurrence of some $s \in S$ at position p with an interval $[p + [s] - L : p]$ with color c_s. The set of colored intervals

$$\mathcal{I} = \{([p + [s] - L : p], c_s) \mid s \text{ occurs in } T \text{ at position } p\} \tag{3.1}$$

is called the *set of intervals induced by* S *in* T *with cluster length* L. Figure 3.1 shows the construction of colored intervals corresponding to binding sites. There exists an one-to-one correspondence between the intervals $\mathcal{I}$ induced by S in T and L-occurrences in T: A set of fragments $A \subseteq S$ is an L-occurrence in T w.r.t. S iff there exists an integer x such that for all $s \in A$, x stabs an interval in $\mathcal{I}$ with color c_s. A proof of this lemma is given in [37]. In order to find a maximum L-occurrence, we search for an integer x that stabs a maximum number of colored intervals from the set of induced intervals in T. For a better understanding, we can illustrate this problem by a real world analogon. Assume

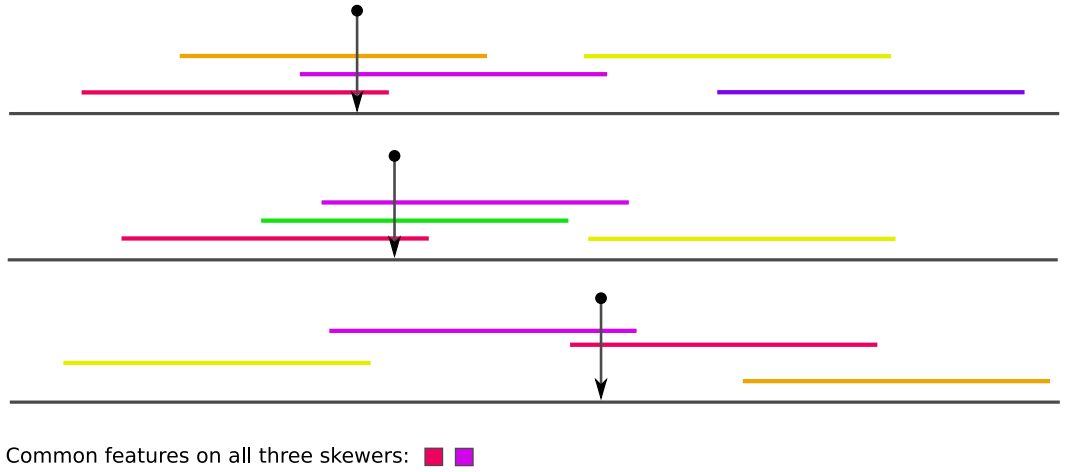


Figure 3.2: Stabbing skewers into BBQ ingredients. The black line denotes the BBQ plate and the colored intervals represent the BBQ ingredients on the plate. We stab one skewer in each plate and want to maximize the subset of ingredients which can occur on all skewers: the Best-Barbecue-Problem. In this example each skewer contains the red and purple ingredients. This is also a best BBQ. Note that the first two skewers contain additional features.

that the m different colors correspond to m different barbecue (BBQ) ingredients which are placed (possibly overlapping) on a barbecue plate denoting the string T. In order to have a tasty barbecue we want to *stab* as many different ingredients as possible on a skewer by stabbing only once in the plate. We call this problem the *single person Best-Barbecue-Problem* if only one BBQ plate is involved. In a scenario with more than one BBQ plate, the requirements for stabbing the ingredients is slightly different. Suppose we have K guests and each guest gets a BBQ plate with our m different ingredients. The ingredients are placed randomly and possibly overlapping on the plate and each ingredient may occur an arbitrary number of times. Now we stab one skewer in each plate. Since we want to treat all guests as equal as possible, we want to maximize the set of ingredients that is contained on all skewers, but we allow that some skewers may contain additional ingredients. We call this maximal set of common ingredients on all skewers a *best BBQ*. The problem of finding such a best BBQ is called the *Best-Barbecue-Problem*. Returning to the colored intervals, we can formalize this problem as following: Let $\mathcal{I}_1, ..., \mathcal{I}_K$ denote sets of colored intervals. We say that a set of

colors $A \subseteq [1:m]$ is an $(\mathcal{I}_1, ..., \mathcal{I}_K)$-*barbecue* if for each $i \in [1:K]$, there is an integer x_i, such that for each color $a \in A$, x_i stabs at least one interval of color a in $\mathcal{I}_i$.
Since the colored intervals and L-occurrences are equivalent, a best BBQ in the interval arrangement $\mathcal{I}$ corresponds to a maximum L-occurrence that occurs clustered in the K sequences. Therefore by solving the Best-Barbecue-Problem, we solve the problem of finding a maximum cardinality set of sequence fragments occurring clustered in a set of genomic sequences.

3.2 Combinatorial Best-Barbecue-Problem

The problem of stabbing integers in sets of colored intervals can be rephrased to the problem of finding intersections with maximum cardinality of sets of colors. We call an interval C a cell w.r.t. $\mathcal{I}$ if each integer $x \in C$ stabs the same set of colors and every interval $C' \supset C$ contains at least one x' which stabs a different set of colors than some $x \in C$. Thus from the set of colored intervals $\mathcal{I}$, one can obtain an arrangement of cells $\mathcal{C}$, where each $\mathcal{C}_i$ contains the cells induced by $\mathcal{I}_i$, i.e. $\mathcal{C}_i$ is a set of subsets of $[1:m]$. Figure 3.3 shows the correlation between colored intervals and an arrangement of cells. From this point of view, we consider the Best-Barbecue-Problem as a pure combinatorial problem. An instance of the *Combinatorial Best-Barbecue-Problem* can be stated as following: After identifying the colored intervals on the K sequences we obtain for each sequence i a cell set $\mathcal{C}_i = \{C_{i,1}, ..., C_{i,\lambda_i}\}$ with $C_{i,j} \subseteq [1:m]$i and $\lambda_i := |\mathcal{C}_i|$. The

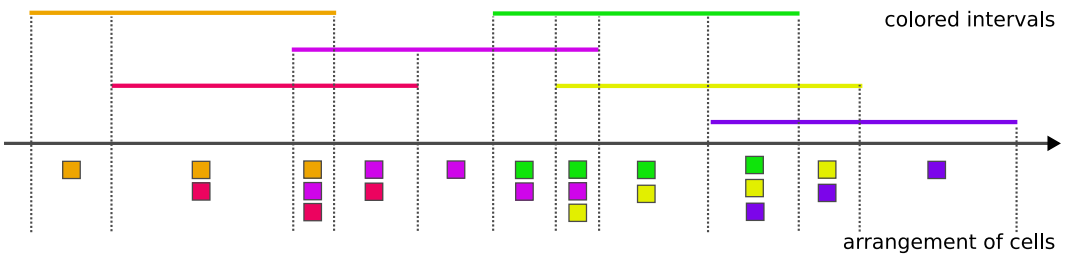


Figure 3.3: The graphics shows how the colored intervals can be transformed into a system of sets of colors. This arrangement system is the foundation of the combinatorial Best-Barbecue-Problem.

maximum cluster of colors contained in each $\mathcal{C}_i$ can be found by maximizing

$$|\bigcap_{i\in[1:K]} C_{i,\nu_i}| \text{ with } \nu_i \in [1:\lambda_i] \tag{3.2}$$

We can apply two different approaches to solve this problem:

Algorithm A1

Enumerate all combinations of the cells in $\mathcal{C}_i$, denoted by the vector of indices $(\nu_1, ..., \nu_K) \in [1:\lambda_1] \times ... \times [1:\lambda_K]$, and compute $|\bigcap_{i\in[1:K]} C_{i,\nu_i}|$. Keep track of the largest cardinality intersection.

Algorithm A2

Enumerate all subsets of $[1:m]$. For each of $A \subseteq [1:m]$ check whether there exists a vector of indices $(\nu_1, ..., \nu_K)$ such that $A \subseteq \bigcap_{i\in[1:K]} C_{i,\nu_i}$. Keep track of the subset A for which such suitable indices were found.

3.3 Complexity & Branch and Bound

While algorithm `A1` optimizes the cardinality of the possible intersections of the cells in the arrangement system and algorithm `A2` enumerates all possible combinations of binding sites, both algorithms rely on a brute force search. Thus the required time for testing all possible solutions increases exponentially with the input size. The worst case time complexity of A1 depends essentially on the number of sequences K:

$$\mathcal{O}(Km\lambda^K), \text{ with } \lambda = \max_i \lambda_i \tag{3.3}$$

whereas A2 depends on the number of binding sites m:

$$\mathcal{O}(2^m \Lambda m), \text{ with } \Lambda := |C_1| + ... + |C_K| \tag{3.4}$$

There is no algorithm known, which finds the best solution to the Best-Barbecue-Problem in polynomial time. Actually the Best-Barbecue-Problem has been proven to be NP-complete[37]. Luckily the input parameters in the exponent are usually small in most instances, so that the running time is still in a practical range. Additionally the mean running time of both algorithms can be decreased significantly by applying appropriate branch-and-bound modifications.

The number of binding site combinations which are tested against the arrangement system can be decreased by cutting the edges of the search tree which cannot yield a best BBQ. If a set of binding sites A is no BBQ, i.e. it is not found in a cell in each sequence,then all supersets of A are no BBQs too. Thus by starting the search with one-element sets and increasing the size, this method prevents unnecessary tests and reduces the running time considerably.

3.4 Implementation

The `bbq` software[1] implements both algorithms A1 and A2. The program accepts genome sequences in the FASTA format and TFBS motifs either as a consensus sequence about the IUPAC alphabet[2] or as PCMs. `bbq` is written in the C++ language and runs under *NIX and Windows boxes. The source code is published under the terms of the GNU General Public License[3].

3.5 Limitations

Both algorithms `A1` and `A2` provide always the best *accurate* results. The *best BBQ* is always contained in **all** input sequences. While it is nice to obtain always correct results, this approach has several drawbacks.
The success of the algorithm strongly depends on a careful selection of the input sequences. If we have only one "bad" sequence, which contains only one or few binding sites, the overall best BBQ is restricted to the sites contained in this sequence. Strictly speaking, if one sequence contains no sites at all, the algorithm returns no best BBQ, although the other sequences might share a common set of binding sites.
Additionally consider this example: Let $S_1 = \{A, B\}$, $S_2 = \{A, C\}$ and $S_3 = \{B, C\}$ be cells of binding sites contains in the sequences 1 to 3. The `bbq` algorithm has no chance to find a best BBQ Z with $Z \subseteq S_1 \wedge A \subseteq S_2 \wedge A \subseteq S_3$,

[1] `http://www.bioinf.uni-leipzig.de/Software/bbq/`
[2] `http://www.chem.qmul.ac.uk/iupac/misc/naabb.html`
[3] `http://www.gnu.org/copyleft/gpl.html`

simply because it does not exist:

$$\bigcap_{i=1..3} S_i = \emptyset$$

Using a lesser strict search algorithm may yield a better result: $Z = \{A, B, C\}$. Although Z is not a subset of any S_i, it is a good representation of the sites found in the three sequences.

Therefore the motivation of this work is to extend `bbq` by a lesser strict search method and scoring system which ranks the found results. We want a scoring function f, which applied to a candidate set Z, gives a similarity score *how good* it matches the given cells in the sequences. This approach will certainly lead to non-accurate results.

In the next chapter we develop such a scoring method for the Best-Barbecue-Problem and apply it the `bbq` algorithm.

Chapter 4

Novel Scoring Schemes

The motivation of this work is to extend the `bbq` algorithm by a scoring method which increases the flexibility of the search algorithm and overcomes the limitations of the original algorithm discussed in section 3.5. For this purpose we implement a similarity score, the so-called *Tanimoto score*, which compares two sets of binding sites and returns a measure on the similarity of both sets.

4.1 Tanimoto Scores

The Tanimoto score is essentially a similarity score between two sets and is defined as follows:

$$\mathrm{tnm}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

with $|X|$ denoting the cardinality of the set X. Following this definition, $\mathrm{tnm}(X, Y)$ describes the relation between the number of elements common in both sets and the number of all distinct elements found in both sets.

4.1.1 Variants

Beside the "classic" Tanimoto scoring function, we can apply slightly modified versions, with different properties.

$$\mathrm{tnm}_1(X, Y) = \frac{|X \cap Y|}{|X \setminus Y \cup Y \setminus X|} = \frac{|X \cap Y|}{|X \triangle Y|} \tag{4.1}$$

with $X \triangle Y := X - Y \cup Y - X$. In equation 4.1 the score denotes the fraction of elements common in both sets X and Y and elements occurring either (exclusively) in X or Y.

$$\mathrm{tnm}_2(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{4.2}$$

Equation 4.2 states the classic Tanimoto coefficient between two sets. Its value is always in the interval $[0 : 1]$, being 1.0 if both sets are equal.

$$\mathrm{tnm}_3(X, Y) = \left(\frac{(|X \cap Y|)^2}{|X \cup Y|} \right)^2 \tag{4.3}$$

Equation 4.3 squares the term for identical elements in both sets, thus increasing the weight of equal elements. Additionally by squaring the whole term, we favor larger sets over smaller ones. Note that the values fall not necessarily in the interval $[0 : 1]$ anymore.

4.2 BBQ with Tanimoto Scores

To integrate the scoring formulas into the `bbq` algorithm, we extend the algorithm A2 which enumerates all combinations of distinct binding sites found in the sequences. Each of these candidate sets is scored against each cell of the arrangement using the Tanimoto scores. The sum of the highest local scores in each sequence makes up the overall score of a candidate set.
For each candidate set $A \subseteq [1 : m]$ calculate:

$$\mathrm{Tnm}^{\Sigma}(A) = \sum_{i=1}^{K} \max_{j=1}^{\lambda_i} \mathrm{tnm}(A, B_{i,j})$$

with K being the number of sequences and λ_i the number of cells in sequence i; $B_{i,j}$ denotes the jth in the ith sequence. Furthermore we can divide the overall score by the number of sequences K. This normalization step guarantees the comparability of the results between data sets of different size.

$$\mathrm{Tnm}^{\Sigma}(A) = \frac{\sum_{i=1}^{K} \max_{j=1}^{\lambda_i} \mathrm{tnm}(A, B_{i,j})}{K} \tag{4.4}$$

4.3 Limitations

While an efficient branch-and-bound extension can be applied to the original A2 algorithm, this is no longer possible with the scoring extension. In algorithm A2 every superset of a set which is no BBQ could be skipped for testing, this bound no longer holds true for the Tanimoto scores. Thus we must enumerate all 2^n different combinations of n different motifs. With increasing n the runtime complexity quickly exeeds the limit for computations in reasonable time. Therefore we can only reduce the complexity by omitting certain candidate sets, which are unlikely to be a best BBQ. Several methods have been implemented to reduce the number of candidate sets.

4.3.1 δ-bounded Candidates

The main purpose of the δ-bounded candidates method is to reduce the runtime complexity by only enumerating certain candidates. In contrast to the conventional A2 algorithm, only candidate sets are enumerated, which are similar to the cells contained in the arrangement. Therefore only candidates are chosen which have at most δ more or less elements compared to each cell in the arrangement. Because this approach will construct many equal candidates, they are collected in a set containing only distinct candidates before testing them. This candidate set is constructed as following: First all distinct TFBS motifs are collected:

$$C = \bigcup_{i=1}^{K} \bigcup_{j=1}^{\lambda_i} \{C_{ij}\} \tag{4.5}$$

In the next step all subsets of C with cardinality $1..\delta$ are enumerated. This is required for adding elements to the cells. For the purpose of enumerating all n-element subsets of C Knuth's "Revolving-doors" algorithm is used [30]. The advantage of this method is that the number of operations on the set C is minimized.
Let $P^\delta(X)$ be the set of all subsets of X with cardinality δ.

$$C^\delta = \bigcup_{d=1}^{\delta} P^d(C) \tag{4.6}$$

After this, the candidate set S^+ is constructed by adding every element of C to each cell in the arrangement.

$$S^+ = \bigcup_{i=1}^{K} \bigcup_{j=1}^{\lambda_i} \bigcup_{e \in C^\delta} \{C_{ij} \cup e\} \tag{4.7}$$

And finally the candidates S^- are constructed by removing every subset up to size δ from each cell.

$$S^- = \bigcup_{i=1}^{K} \bigcup_{j=1}^{\lambda_i} \bigcup_{d=1}^{\delta} \bigcup_{e \in P^d(C_{ij})} \{C_{ij} \backslash e\} \tag{4.8}$$

The final candidate set $S = S^- \cup S^+$ contains all candidates with at most δ many elements more or less then each cell in the arrangement.

4.4 Weighted Scores

Until now we only studied Tanimoto scores between standard sets. On the one hand, we have a candidate set A of binding sites and on the other hand we have a cell $B_{i,j}$ in the sequence. We then calculate the Tanimoto score between both sets and get a score *how good* both sets fit together in terms of common and different elements. But in the BBQ scenario, we have not standard sets of the occurring binding sites in each cell, but we have matches of PWMs to sequence positions. Each match has an associated weight denoting the match score of a binding site and the site's matrix profile. This score is a real number between 0.0 and 1.0. The higher the weight is, the better fits the PWM to the sequence.

4.4.1 Fuzzy Sets

Having a weight for each binding site leads us from standard sets of binding sites to so called *fuzzy sets*. Fuzzy sets are an extension to the standard set theory, where each element belongs with a certain *degree* (or *weight*) to a set.
The usual notation for fuzzy sets is: $S = \{0.85/A, 0.9/B, 0.5/C, ...\}$, meaning that element A is contained in S with a degree of 0.85, B is contained with degree 0.9, and so on. Fuzzy sets need special operations for computing the intersection,

union and the complement of them. For easier notation we just say that $S(A)$ is the degree to which element A belongs to a set S.

Standard fuzzy intersection:

$$(A \cap B)(x) = \min[A(x), B(x)]$$

The intersection of two sets contains *everything that is contained in both sets.* Thus we can not choose a weight higher then the smaller of the two values $A(x)$ and $B(x)$, because only this minimum value is contained in both sets.

Standard fuzzy union:

$$(A \cup B)(x) = \max[A(x), B(x)]$$

The union of two sets contains *the maximum elements we find in both sets.* Therefore we choose the higher of the two values $A(x)$ and $B(x)$.

4.4.2 Weighted Tanimoto Scores

Consider a candidate set $A = \{A, B, C\}$ and one of the cells in the arrangement $B = \{0.84/A, 0.92/C, 0.95/D\}$. We see that element B is contained in the candidate set A, but not in the cell, and element D is contained with grade 0.95 in the cell B but not in the candidate set. If we want to apply the fuzzy set operations with the Tanimoto scores, we must modify the scoring function in some way. Since the elements of a candidate set have no associated weight, we can only consider the weights of the elements in each cell. Therefore we only modify the intersection term $A \cap B$ and sum the weights of the elements in the standard intersection of A and B. The modified tnm_i are defined in a weighted version each, denoted by tnm_i^w:

$$\text{tnm}_1^w(A, B) = \frac{\Sigma_{x \in A \cap B} B(x)}{|A \triangle B|} \tag{4.9}$$

$$\text{tnm}_2^w(A, B) = \frac{\Sigma_{x \in A \cap B} B(x)}{|A \cup B|} \tag{4.10}$$

$$\text{tnm}_3^w(A, B) = \left(\frac{(\Sigma_{x \in A \cap B} B(x))^2}{|A \cup B|} \right)^2 \tag{4.11}$$

4.5 Implementation

The original `bbq` program implements both algorithms A1 and A2 for solving the Best-Barbecue-Problem. We therefore extend the existing source code with the new search algorithm using the Tanimoto scoring functions tnm_i in both unweighted and weighted variants. Additionally several options for limiting the search space, like the δ-bounded candidate sets, are implemented. Following new command line parameters are introduced:

-T x enables Tanimoto scoring. x denotes the variant 1, 2 or 3.

-u enables unweighted Tanimoto scores, default are weighted scores.

-D x δ-bounded candidates are used, where x specifies the bounding distance, see section 4.3.1.

-L x only candidates with size x are enumerated.

-I x y only candidates with sizes from x to y are enumerated.

-h x the x best scoring modules are returned, without modifying the arrangement system.

-H x the x best scoring modules are returned, and after each run the sites occurring in the best module are removed from the arrangement system.

For a more complete overview of `bbq`'s options refer to appendix A.

Chapter 5

Performance Evaluation

To access the performance of the extended `bbq` algorithm, we validated the Tanimoto scores on two biological data sets which have been studied before. Additionally the `bbq` algorithm was tested ont artificial data sets. Figure 5.1 shows

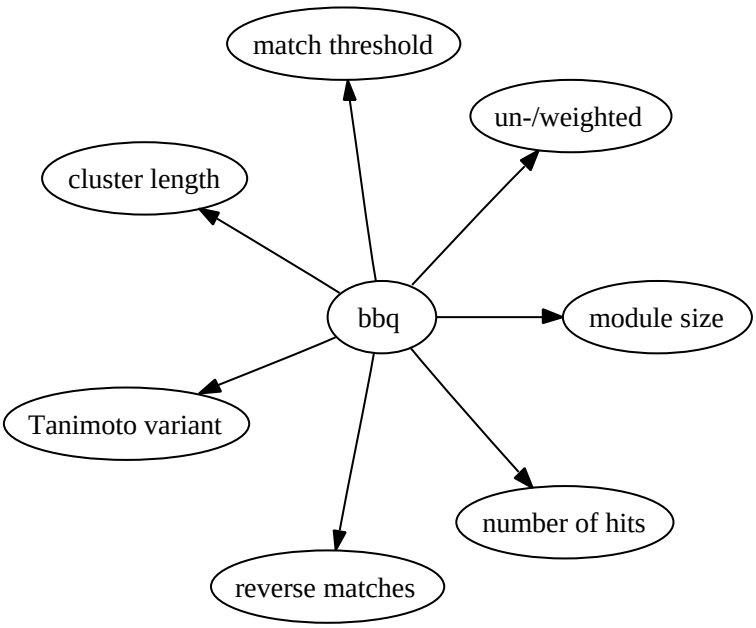


Figure 5.1: `bbq` input parameters.

the different input parameters which can be chosen in each run of `bbq`. Due to the high number of possible parameter combinations, only a reasonable subset was used in each scenario. Whereas all three Tanimoto variants were used, either weighted or unweighted, the other parameters were set to only a few standard

values. For example all motif searches were performed on both strands of the genomic sequence. Furthermore only the three best scoring modules found in each instance are collected. The module size was restricted to contain up to five motifs, due to the computational efforts computing all possible combinations of motifs (see section 4.3). Values for the match threshold and module length were chosen, depending on the properties of the actual data sets used.

5.1 Muscle Genes

The first performance evaluation is carried out on a set of muscle-specific regulatory regions, first introduced by Wasserman et al.[57]. Perco et al. used this dataset for evaluating the genetic algorithm, thus it is used here for a comparison with `bbq`.
The muscle-specific genes are known to be regulated by several TFs which interact cooperatively at multiple sites in the regulatory sequences. Major interacting TFs are: *Myf*, *Mef-2*, *SRF*, *Tef-1* and the general factor *Sp-1*. No single sequence contains the combination of all five factors, but a subset of the five factors occurs frequently in most but not all sequences.

5.1.1 Data Compilation

The data set consists of 46 DNA sequences of promoter and enhancer regions of 39 distinct human genes. The average sequence length is 333nt, with a minimum of 61nt and maximum of 1029nt. Position weight matrices for the five known TFs are taken from the original publication, ensuring a high specificity of the pattern search. Additionally weight matrices from JASPAR CORE belonging to *homo sapiens* with an information content above 11 were added to the motif set. The motif set contains 33 PWMs for distinct TFBSs.

5.1.2 Results

At first `bbq` was run with the standard algorithm A2 and different module lengths of 200, 500 and 1000 nucleotides and a minimum match threshold of 0.8. All runs did not find any L-Occurrences. This is due to the fact, that the sequence set

contains many sequences with only a few detected motifs. With a score threshold of 0.8 we have two sequences containing only one motif, three sequences containing two motifs and one sequence which contains no motif at all. As discussed in section 3.5 these *weak* sequences prevent the A2 algorithm from finding an L-Occurrence in the sequence set.

Next, `bbq` was run with the Tanimoto scores and following parameter combinations:

Parameter	Values
Tanimoto variant `-T`	1,2,3
weighted scores `-u`	on / off
match threshold `-W`	0.8, 0.825, 0.85, 0.875, 0.9
module size `-L`	2, 3, 4, 5
module length	200, 500, 1000

Altogether 360 runs were performed and the three best results (`-H 3`) were collected for each parameter combination. See the web supplements for the outputs of all runs.

We expected to find the motifs of *Myf*, *Mef-2*, *SRF*, *Tef-1* and *Sp-1* in the best BBQs found. Tables 5.1, 5.2 and 5.3 show the top scoring modules using all three Tanimoto variants and a module length of 500 with varying match thresholds 0.85, 0.75 and 0.9. Note that the occurrence (occ.) columns lists the number of occurrences of the respective motif set in the highest scoring cells in each sequence. Thus it can differ for the different Tanimoto variants. For example Tanimoto variant 2 does not favour sets with larger cardinality against other sets with the same ratio of same and different elements. The three combinations of {MYF, SP1}, {MEF2, SP1} and {SP1, SRF} are the top scoring modules of size 2, but only {MEF2, SP1} and {SP1,SRF} occur within all three match thresholds. Tanimoto variant 3 favours {SP1, SRF} over {SP1, MEF2} because the former occurs four times and the latter only three times in a sequence containing an equal cell (with match threshold 0.9). These cells get an unweighted score of 4, which results in a higher overall score when appearing four times instead of three. Generally Tanimoto variant 3 favours candidates with most occurrences in the sequences. Table 5.4 shows the top 3 modules of size 2 with unweighted Tanimoto variant 3. The score distribution shows highest scoring modules at

size	T var.	motifs	score	occ.
2	1	MEF2,SP1	0.893892	7
2	2	MYF,SP1	0.440217	7
2	3	SP1,SRF	0.802905	9
3	1	MEF2,MYF,SP1	0.893116	4
3	2	MEF2,MYF,SP1	0.386594	4
3	3	E2F,MYF,SP1	0.938206	4
4	1	E2F,MEF2,MYF,SP1	0.769565	1
4	2	E2F,MEF2,MYF,SP1	0.364907	1
4	3	HFH-3,MEF2,MYF,SP1	1.23575	2
5	1	E2F,MEF2,MYF,SP1,USF	0.71087	0
5	2	E2F,MEF2,MYF,SP1,SRF	0.3456	1
5	3	E2F,MEF2,MYF,SP1,SRF	1.46242	1

Table 5.1: Highest scoring modules with match threshold of 0.85 and a window length of 500.

size	T var.	motifs	score	occ.
2	1	MEF2,SP1	0.811594	5
2	2	MEF2,SP1	0.403985	5
2	3	SP1,SRF	0.70471	8
3	1	MEF2,SP1,SRF	0.766304	2
3	2	MEF2,MYF,SP1	0.340217	3
3	3	MEF2,MYF,SP1	0.711226	3
4	1	MEF2,MYF,SP1,TEF	0.630435	1
4	2	MEF2,MYF,SP1,SRF	0.305072	1
4	3	MEF2,MYF,SP1,SRF	0.921184	1
5	1	MEF2,MYF,RORalfa-1,SP1,SRF	0.563768	0
5	2	MEF2,MYF,SP1,SRF,TEF	0.282298	1
5	3	MEF2,MYF,SP1,SRF,TEF	1.07934	1

Table 5.2: Highest scoring modules with match threshold of 0.875 and a window length of 500.

size	T var.	motifs	score	occ.
2	1	MEF2,SP1	0.746377	4
2	2	MEF2,SP1	0.367754	4
2	3	SP1,SRF	0.506039	5
3	1	MEF2,SP1,SRF	0.641304	1
3	2	MEF2,SP1,SRF	0.302536	1
3	3	MYF,SP1,SRF	0.5	1
4	1	MEF2,MYF,SP1,SRF	0.51087	0
4	2	MEF2,MYF,SP1,SRF	0.26413	0
4	3	Irf-1,MEF2,SP1,SRF	0.55125	1
5	1	Irf-1,MEF2,MYF,SP1,SRF	0.400362	0
5	2	MEF2,MYF,SP1,SRF,TEF	0.234058	0
5	3	Irf-1,MEF2,MYF,SP1,SRF	0.500194	0

Table 5.3: Highest scoring modules with match threshold of 0.9 and a window length of 500.

lowest match thresholds. When increasing the module length, the score may get lower, because the cells typically contain more elements, thus reducing the similarity to the candidate modules of size 2. Top scoring modules of size 3 are {MEF2, SP1, SRF}, {MYF, SP1, SRF} and {MEF2, MYF, SP1}.
With a match threshold of 0.875 a module containing all five known TFs of size 5 occurring in one sequence. Both Tanimoto variants 2 and 3 find this module as best L-Occurrence whereas variant 1 finds it as third best hit with score 0.0.546739 after modules {MEF2, MYF, RORalfa-1, SP1, SRF} with score 0.563768 and {MEF2, MYF, SP1, SRF, Tal1beta-E47S} with score 0.558333. These scores are very close, indicating no significant order of the top hits.

Comparison to the Genetic Algorithm

The Genetic Algorithm found the modules of size 2 {SP1, MEF2}, {SP1, MYF} and {MYF, MEF2}. Additionally {SP1, MEF2, MYF} as best module of size 3 was found. This module was also found by the Tanimoto `bbq`algorithm, as well as {SP1, MEF2} and {SP1, MYF}. The combination of {MYF, MEF}

W	N	h		
		1.	2.	3.
0.8	200	SP1, USF	MYF, SP1	MEF2, SP1
	500	MYF, SP1	SP1, USF	MEF2, SP1
	1000	MYF, SP1	SP1, USF	MEF2, SP1
0.825	200	SP1, USF	SP1, SRF	MYF, SP1
	500	SP1, USF	SP1, SRF	MYF, SP1
	1000	SP1, USF	SP1, SRF	MYF, SP1
0.85	200	SP1, SRF	MYF, SP1	MEF2, SP1
	500	SP1, SRF	MYF, SP1	MEF2, SP1
	1000	SP1, SRF	MYF, SP1	MEF2, SP1
0.875	200	SP1, SRF	MEF2, SP1	SP1, TEF
	500	SP1, SRF	SP1, TEF	MEF2, SP1
	1000	SP1, SRF	SP1, TEF	MEF2, SP1
0.9	200	SP1, SRF	MEF2, SP1	MYF, SP1
	500	SP1, SRF	MEF2, SP1	MYF, SP1
	1000	SP1, SRF	MEF2, SP1	MYF, SP1

Table 5.4: Top 3 scoring modules with size 2 and unweighted Tanimoto variant 3. Highest score is 0.996069 and lowest score is 0.370773

was not recognised as a high scoring module, because of the dominance of SP1, which occurs in more sequences than the other TFs. Using a match threshold of 0.85, SP1 occurs in 33, MEF2 in 15 and MYF in 18 sequences. When removing SP1 from the set of TFs, the module {MEF2, MYF} is found as highest scoring module of size 2 with all three Tanimoto variants.

5.2 Beta-Actin Genes

The next biological example is performed on a set of promoters of beta-actin (ACTB) genes. A CRM in these promoters was described earlier by Frech et al.[18]. It contains three promoter motifs: a *CAAT* box, a *SRF* motif and a muscle specific TATA box, *mTATA*. Perco et al. also used the beta-actin genes for evaluating their genetic algorithm. Additionally to the above three motifs

they found three more frequently detected motifs: *TATA*, *TCF4* and *NFY*.

5.2.1 Data Compilation

The EPD (see 2.1.3) was searched for beta-actin genes and three entries were found. Additionally the ENSEMBL database was searched for beta-actin genes and the upstream regions of two matching genes were retrieved. Furthermore 6 upstream regions were fetched from Genbank, from which four sequences were used by Perco et al. before. See table 5.5 for an overview and the web supplement for the complete sequence collection. Altogether we have 11 sequences with an average length of 342nt.

Species	Length	Source	Position
Homo sapiens	601	EPD, EP17045	-500 to 100 relative to TSS
Gallus gallus	601	EPD, EP07061	-500 to 100 relative to TSS
Rattus norvegicus	335	EPD, EP07062	-234 to 100 relative to TSS
Bos taurus	601	ENSEMBL v39, Cow	chr3, 54412515:54413115:1
Mus musculus	601	ENSEMBL v39, Mouse	chr5, 143168400:143169000:1
Misgurnus mizolepis	151	Genbank, AF270649	3177 to 3327
Cricetulus griseus	151	Genbank, U20114	41 to 191
Megalobrama amblycephala	151	Genbank, AY170122	1816 to 1966
Oryzias latipes	151	Genbank, S74868	744 to 994
Pan troglodytes	170	Genbank, NM_001009945	1 to 170
Xenopus tropicalis	250	Genbank, NM_001006111	1 to 251

Table 5.5: Overview of the 11 beta-actin gene promoters.

The TRANSFAC database (release 8.2) serves as PCM repository. It contains 531 matrix profiles for motifs belonging to 357 families. For this study, only high quality matrices as defined in [46] were considered in the search. 155 matrices with a Q-balance above 0.95 considering a `GC`-rich background distribution remained. The motifs for *TCF4*, *MTATA* and *CAAT_01* with lower Q-balance values were added, due to their estimated relevance. Altogether 158 matrices were used, see complete list in the web supplement.

5.2.2 Results

At first `bbq` was run with the standard algorithm A2 and different module lengths 100, 200 and 500 and minimum match threshold of 0.8. None of the three runs provided a best BBQ. In the sequence set, one sequence (*Pan troglodytes*) yields only one match: *CIZ*, which is not present in every other sequence. This upstream sequence is possibly no real promoter sequence, otherwise some hits could be expected, or it contains really no similar elements in comparison to the other sequences. Therefore the *Pan troglodytes* sequence was manually removed from the sequence set and the standard A2 algorithm was run again with the same settings. This time it found a module containing the single element *CAAT* which occurs in all 10 sequences. After obtaining these results with the standard approach, the brand new Tanimoto scores were applied. Again `bbq` was run with various parameter combinations and for each combination the best three results were obtained. Following input parameters were chosen:

Parameter	Values
Tanimoto variant `-T`	1,2,3
weighted scores `-u`	on / off
match threshold `-W`	0.85, 0.9
module size `-L`	2, 3, 4, 5
module length	100, 300

Table 5.6 shows the number of motifs found in each promoter. Note that the six known motifs, accompanied by *CIZ*, are the most occurring motifs in the sequence set. Tables 5.7 and 5.8 show the occurrences of motif matches which occur at least in two different sequences with a match threshold of 0.85 and 0.9 respectively.

Considering a match threshold of 0.85 six motifs occur in 7 respectively 8 sequences. Thus the search for a module of size two results in more than one hit, using the different scoring methods. Best scoring modules are {TATA, TCF4}, {CAAT, TATA} and {CAAT, NFY}. With a match threshold 0.9, the most prominent module with size two contains CAAT and TATA and can be found in 7 sequences.

Sequence	W 0.85	W 0.9
Homo sapiens	7	4
Gallus gallus	26	11
Rattus norvegicus	8	5
Bos taurus	6	2
Mus musculus	15	3
Misgurnus mizolepis	7	3
Cricetulus griseus	8	4
Megalobrama amblycephala	6	3
Oryzias latipes	9	2
Pan troglodytes	1	0
Xenopus tropicalis	5	1

Table 5.6: Number of distinct TF motifs found in each sequence with a matrix match threshold of 0.85 and 0.9 respectively.

With module size three and a match threshold of 0.85 no single module is scored best in all Tanimoto variants. Whereas the combination of {CAAT, NFY, SRF} occurs in 7 sequences, it is recognized only by the unweighted Tanimoto variant 3 as highest scoring module. Variant 3 also finds the combinations {MTATA, TATA, TCF4} and {CAAT, CIZ, NFY}, both occurring in 4 sequences. The reason is that some sequences contain cells having nearly exactly this set of sequences, resulting in a high score matching such a cell with the candidate motif set, because of little differences. These cells result from the method how the arrangement of cells is constructed from the interval arrangement. The best scoring module of size 3 and match threshold of 0.9 is found to be {CAAT, TATA, NFY} which occurs in 4 sequences. In this case all Tanimoto variants find this combination as highest scoring.

With module size 4 the runs with match threshold 0.85 again resulted in a divergent set of top scoring modules among the scoring variants. Whereas the combinations of {CAAT, NFY, SRF, TCF4} and {NFY, SRF, TATA, TCF4} nearly score the same (variant 3), the module {MTATA, SRF, TATA, TCF4} ranks first, though it only occurs in 4 sequences in contrast to the others which appear six times. This is again due to the actual cells which are contained in

	MMEF2	HFH3	HFH8	CAAT	LPOLYA	TATA	MTATA	GFI1	FOXJ2	S8	NFY	HNF3B	SRF	NKX22	CIZ	ALPHACP1	TCF4	XFD2
B. taurus										x							x	
C. griseus				x		x	x				x		x		x	x	x	
G. gallus		x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x
H. sapiens				x		x	x				x		x		x		x	
M. amblycephala				x		x		x			x		x				x	
M. musculus	x	x	x	x		x			x	x		x		x	x		x	x
M. mizolepis				x	x	x					x		x					
O. latipes	x			x		x					x		x				x	
P. troglodytes															x			
R. norvegicus				x		x	x				x		x		x	x	x	
X. tropicalis														x	x			
	2	2	2	8	2	8	4	2	2	3	7	2	7	2	7	3	8	2

Table 5.7: Occurrences of TFBS motifs, that occur in more than one genomic sequence. Matching was conducted with a threshold of 0.85.

the arrangement system. We can find three sequences each with a cell yielding this motif combination. With the more strict match threshold of 0.9, the modules {CAAT, CIZ, NFY, TATA} and {CAAT, MTATA, NFY, TATA}, both occurring twice, share the top positions with similar scores. Only with Tanimoto variant 3 the latter is considerably scored better than the former. Additionally there is no difference in the order of the top modules when switching between weighted and unweighted scoring methods.

The most frequent occurring module of size five is {CAAT, NFY, SRF, TATA, TCF4}, which appears in 6 sequences. All three Tanimoto variants found it among the top three candidates, but only variant 3 scores it best both with weighted and unweighted scores. There are two combinations {MTATA, NFY, SRF, TATA, TCF4} and {CAAT,CIZ, NFY, SRF, TCF4} which occur 4 times and have slightly different scores, but score considerably less than the top module. With the strict match threshold, as Table 5.8 indicates, only few motif combinations of size five appear in at least one sequence and no module of this size

	TATA	MTATA	CIZ	S8	NFY	CAAT	TCF4
Bos taurus				x			x
Cricetulus griseus	x	x			x	x	
Gallus gallus	x		x	x	x	x	x
Homo sapiens	x	x	x			x	
Megalobrama amblycephala	x					x	x
Mus musculus	x		x				
Misgurnus mizolepis	x				x	x	
Oryzias latipes	x					x	
Pan troglodytes							
Rattus norvegicus	x	x	x		x	x	
Xenopus tropicalis							
	8	3	4	2	4	7	3

Table 5.8: Occurrences of TFBS motifs occurring in more than one genomic sequence. Matching was conducted with a threshold of 0.9

occurs in two or more sequences. The combination of {CAAT, CIZ, MTATA, NFY, TATA} was found by all Tanimoto variants as the best scoring candidate. *SRF* is found in only one sequence with the strict match threshold, thus it does not appear in table 5.8, while with the lower match threshold it appears in 7 sequences. It is therefore not contained in the best results in either Tanimoto variant.

All six already known motifs *CAAT*, *SRF*, *mTATA*, *TATA*, *TCF4* and *NFY* have been found by the `bbq` variant with Tanimoto scores. The most occurring motifs with the both match thresholds are *CAAT* and *TATA*. Additionally the motif *CIZ* appears in many detected modules, which has not been associated before with beta-actin gene regulation.

size	T var.	motifs	score	occ.
2	1	V$TATA,V$TCF4	1.34713	7
		V$CAAT,V$TATA	1.2878	7
		VCIZ,VTATA	1.2762	5
2	2	V$CAAT,V$CIZ	0.529806	3
		V$CAAT,V$NFY	0.448148	7
		V$TATA,V$TCF4	0.398686	6
2	3	V$MTATA,V$TATA	1.09132	4
		V$CAAT,V$NFY	1.00407	7
		V$CAAT,V$CIZ	0.98698	4
3	1	V$CAAT,V$TATA,V$TCF4	1.90003	6
		VNFY,VTATA,V$TCF4	1.88212	6
		VSRF,VTATA,V$TCF4	1.86008	6
3	2	V$CAAT,V$CIZ,V$NFY	0.519078	4
		V$MTATA,V$TATA,V$TCF4	0.457772	4
		V$CAAT,V$NFY,V$SRF	0.439925	7
3	3	V$MTATA,V$TATA,V$TCF4	2.4172	4
		V$CAAT,V$CIZ,V$NFY	2.25675	4
		V$CAAT,V$NFY,V$SRF	2.15232	7
4	1	V$CAAT,V$NFY,V$TATA,V$TCF4	2.38163	6
		V$CAAT,V$NFY,VSRF,VTATA	2.36703	7
		V$CAAT,V$SRF,V$TATA,V$TCF4	2.35959	6
4	2	V$CAAT,V$CIZ,VNFY,VSRF	0.491379	4
		V$MTATA,V$SRF,V$TATA,V$TCF4	0.475209	4
		V$CAAT,V$NFY,VSRF,VTCF4	0.455037	6
4	3	V$MTATA,V$SRF,V$TATA,V$TCF4	4.28282	4
		VNFY,VSRF,V$TATA,V$TCF4	3.51007	6
		V$CAAT,V$NFY,VSRF,VTCF4	3.50027	6
5	1	V$CAAT,V$NFY,VSRF,VTATA,V$TCF4	2.90353	6
		V$CAAT,V$CIZ,VNFY,VSRF,V$TATA	2.77259	4
		V$CAAT,V$MTATA,VNFY,VSRF,V$TATA	2.70487	4
5	2	V$CAAT,V$CIZ,VNFY,VSRF,V$TCF4	0.508265	4
		V$CAAT,V$NFY,VSRF,VTATA,V$TCF4	0.483163	6
		V$ALPHACP1,V$CAAT,VCIZ,VNFY,V$SRF	0.471765	3
5	3	V$CAAT,V$NFY,VSRF,VTATA,V$TCF4	6.7271	6
		V$MTATA,V$NFY,VSRF,VTATA,V$TCF4	5.68137	4
		V$CAAT,V$CIZ,VNFY,VSRF,V$TCF4	5.42519	4

Table 5.9: Top 3 scoring modules with a window length of 300 and a match threshold of 0.85.

size	T var.	motifs	score	occ.
2	1	V$CAAT,V$TATA	1.2878	7
		VCIZ,VTATA	1.03227	4
		VNFY,VTATA	1.03127	4
2	2	V$CAAT,V$TATA	0.456406	5
		V$MTATA,V$TATA	0.448805	3
		V$TATA,V$TCF4	0.417348	1
2	3	V$CAAT,V$TATA	1.07812	7
		V$MTATA,V$TATA	1.02058	3
		V$CAAT,V$NFY	0.82166	4
3	1	V$CAAT,V$CIZ,V$TATA	1.62894	3
		V$CAAT,V$NFY,V$TATA	1.58575	4
		V$CAAT,V$MTATA,V$TATA	1.50875	3
3	2	V$CAAT,V$NFY,V$TATA	0.476964	4
		V$CAAT,V$CIZ,V$TATA	0.447627	3
		V$CAAT,V$MTATA,V$TATA	0.428653	3
3	3	V$CAAT,V$NFY,V$TATA	2.34692	4
		V$CAAT,V$MTATA,V$TATA	1.8295	3
		V$MTATA,V$NFY,V$TATA	1.71695	2
4	1	V$CAAT,V$CIZ,VNFY,VTATA	1.71592	2
		V$CAAT,V$MTATA,VNFY,VTATA	1.66705	2
		V$CAAT,V$CIZ,V$MTATA,V$TATA	1.55354	2
4	2	V$CAAT,V$CIZ,VNFY,VTATA	0.46185	2
		V$CAAT,V$MTATA,VNFY,VTATA	0.455468	2
		V$CAAT,V$NFY,V$TATA,V$TCF4	0.410375	0
4	3	V$CAAT,V$MTATA,VNFY,VTATA	3.54421	2
		V$CAAT,V$CIZ,VNFY,VTATA	2.95567	2
		V$CAAT,V$CIZ,V$MTATA,V$TATA	2.61104	2
5	1	V$CAAT,V$CIZ,V$MTATA,V$NFY,V$TATA	1.76782	1
		V$CAAT,V$MTATA,VNFY,VS8,V$TATA	1.45467	0
		V$CAAT,V$HFH3,V$MTATA,V$NFY,V$TATA	1.43265	0
5	2	V$CAAT,V$CIZ,V$MTATA,V$NFY,V$TATA	0.426802	1
		V$CAAT,V$CIZ,VNFY,VS8,V$TATA	0.412476	1
		V$CAAT,V$CIZ,VNFY,VTATA,V$TCF4	0.407087	0
5	3	V$CAAT,V$CIZ,V$MTATA,V$NFY,V$TATA	4.16842	1
		V$ALPHACP1,V$CAAT,V$MTATA,V$NFY,V$TATA	2.87665	0
		V$CAAT,V$CIZ,VNFY,VS8,V$TATA	2.82475	1

Table 5.10: Top 3 scoring modules with a window length of 300 and a match threshold of 0.9.

Comparison to the Genetic Algorithm

The genetic algorithm by Perco et al. found a module containing the six motifs {MTATA, SRF, CAAT, TATA, TCF4, NFY}, but no more specific information was provided. This module is similar to the size 5 module found by `bbq` with the strict match threshold. The main advantage of the `bbq` approach over the genetic algorithm is the restriction to clusters contained in a window of a specified length (see section 3). Thus `bbq` can handle long promoter sequences, whereas the genetic algorithm would eventually report binding sites which are very far apart, and hence not much likely to act cooperatively.

5.2.3 Long Range Promoters

To verify this behavior we evaluate `bbq` on larger promoter regions. We updated the promoter set where possible by extending the sequence window. Table 5.11 lists the five sequences that we retrieved from the different databases. Both the*Homo sapiens*and the *Rattus norvegicus*sequences are completely upstream of the ACTB gene, whereas the other sequences cover a range upstream and downstream of the TSS.

Species	Length	Source	Position
Homo sapiens	5001	EPD, EP17045	-5000 to +1 relative to TSS
Rattus norvegicus	5031	ENSEMBL v41 Rat	chr12, 12043070:12048100:1
Cricetulus griseus	4224	Genbank, U20114	full record
Megalobrama amblycephala	6819	Genbank, AY170122	full record
Oryzias latipes	4791	Genbank, S74868	full record

Table 5.11: Five extended promoters. The average length is 5173nt.

Again we used the same set of 158 PCMs obtained from TRANSFAC. `bbq` was run with a window length of 300 and match thresholds of 0.85 and 0.9. Due to the high CPU time demand, only candidates of size three and four were evaluated. Tables 5.12 and 5.13 show the top three scoring modules using all three weighted Tanimoto variants and a match threshold of 0.85 and 0.9 respectively.
With the low match threshold Tanimoto variants 2 and 3 report the module {MTATA,TATA,TCF4} with highest score. The set {CAAT,CIZ,NFY} scores

size	T var.	motifs	score	occ.
3	1	VCIZ,VLPOLYA,V$S8	2.81779	5
		V$CAAT,V$CIZ,V$TATA	2.81732	5
		VCIZ,VHFH3,V$S8	2.79446	5
3	2	V$MTATA,V$TATA,V$TCF4	0.71736	5
		V$CAAT,V$CIZ,V$NFY	0.681549	4
		VCIZ,VHNF3B,V$LPOLYA	0.656019	2
3	3	V$MTATA,V$TATA,V$TCF4	4.6368	5
		V$CAAT,V$CIZ,V$NFY	4.41168	4
		V$ATF6,V$CIZ,V$TCF4	2.95433	4
4	1	V$CAAT,V$CIZ,V$TATA,V$TCF4	3.69788	5
		VCIZ,VFOXJ2,V$HFH3,V$S8	3.6916	5
		V$CAAT,V$NFY,V$TATA,V$TCF4	3.68642	5
4	2	V$MTATA,V$SRF,V$TATA,V$TCF4	0.686721	3
		V$ALPHACP1,V$CAAT,VCIZ,VNFY	0.646016	2
		V$ATF6,V$CIZ,V$MEF3,V$TCF4	0.625241	2
4	3	V$MTATA,V$SRF,V$TATA,V$TCF4	7.22814	3
		V$ALPHACP1,V$CAAT,VCIZ,VNFY	5.85776	2
		V$CAAT,V$CIZ,VNFY,VSRF	5.75192	4

Table 5.12: Top three scoring modules in the long range promoter set with a match treshold of 0.85 and a window length of 300.

similarly. Both of these modules have also been found by `bbq` in the short promoter sequences. Considering modules of size four, we find mostly the same modules as in the standard sequence set. Only Tanimoto variant 1 returns other modules, but with higher scores then in the standard set. This is due to PCM matches occurring in the extended regions. With the strict match threshold the sets of size three both {CAAT, CIZ, TATA} and {CAAT, MTATA, TATA} are among the top modules which have also been reported in the short promoter set. Generally one can observe that the top three modules in each instance of the extended promoter set are more divergent as the top results in the standard set. This effect can only be circumvented by applying a higher match threshold. Thus we cannot increase the promoter range arbitrarily. While `bbq` successfully

size	T var.	motifs	score	occ.
3	1	V$CAAT,V$CIZ,V$TATA	2.6442	4
		VCIZ,VHFH3,V$S8	2.61552	4
		VCIZ,VMTATA,V$TATA	2.48183	3
3	2	V$CAAT,V$CIZ,V$TATA	0.720293	3
		VCIZ,VLPOLYA,V$POU1F1	0.705416	2
		V$CAAT,V$TATA,V$TCF4	0.68937	2
3	3	V$CAAT,V$CIZ,V$TATA	3.78797	4
		V$CAAT,V$MTATA,V$TATA	3.52522	3
		VCIZ,VLPOLYA,V$POU1F1	3.39259	2
4	1	VCIZ,VHFH3,V$LPOLYA,V$S8	3.18343	2
		V$CAAT,V$CIZ,V$MTATA,V$TATA	3.03625	3
		V$CAAT,V$CIZ,V$S8,V$TATA	3.01528	1
4	2	V$CAAT,V$CIZ,VNFY,VTATA	0.705147	2
		V$CAAT,V$CIZ,V$MTATA,V$TATA	0.683278	3
		V$CAAT,V$CIZ,V$TATA,V$TCF4	0.677758	1
4	3	V$CAAT,V$CIZ,V$MTATA,V$TATA	6.59203	3
		V$CAAT,V$CIZ,VNFY,VTATA	6.39733	2
		V$CAAT,V$MTATA,VNFY,VTATA	6.38096	2

Table 5.13: Top three scoring modules in the long range promoter set with a match treshold of 0.9 and a window length of 300.

recognizes most modules in the extended promoter set, it might fail with larger sequences. This is due to many false positive hits during PCM matching. This will lead to false positive hits which score better then the actual present modules.

5.3 Artificial Data Sets

After testing the new algorithm on some real life examples, it is interesting to create artificial datasets to verify the algorithms behavior in a controlled environment. This is done by creating different patterns of TFBS distributions within artificial CRMs and distribute these modules among a set of randomly created DNA sequences, which we just call promoters for convenience. For this purpose

we created a small framework consisting of several Perl scripts which generate such artificial data sets that serve as input for the `bbq` tool.

5.3.1 Generating Random Data

Generating the artificial data sets follows a defined workflow, in which all steps must be executed consecutively, because each step requires the output from the previous one.

Create a random TFBS alignment

The first step is to create short random sequence motifs that serve as TFBSs. As done in [22], first we create a seed sequence of length l. Then we copy the sequence s times. For simulating real TFBSs, the sequences are segmented into a *core region* of length $l/2$ flanked by two border regions of length $l/4$. All positions in a sequence are mutated either with probability r in the border regions or probability $r/4$ in the core region. The construction of the mutated alignment is only necessary if we want to use the data set with `bbq`'s weighted scoring functions. This step is done by `rand_aln.pl`, which takes the three input parameters s, l and r.

Convert Alignment into Position Count Matrix

After creating the - possibly mutated - site alignments, we must convert them into the format of position count matrices, which can be read by `bbq`. This is done by `aln2pcm.pl`.

Create distribution information

In the next step we decide how the artificial TFBSs are distributed among different regulatory modules. This is done by `aln2ini.pl` which takes as input the generated alignment files. Furthermore we can choose whether just the seed sequence or all mutated sequences in the alignment should be taken into account. This decision is only important when using weighted scores. Additionally each site is associated with a number denoting how often it should appear in the created modules. This counter can be chosen to be randomly selected or underlie a

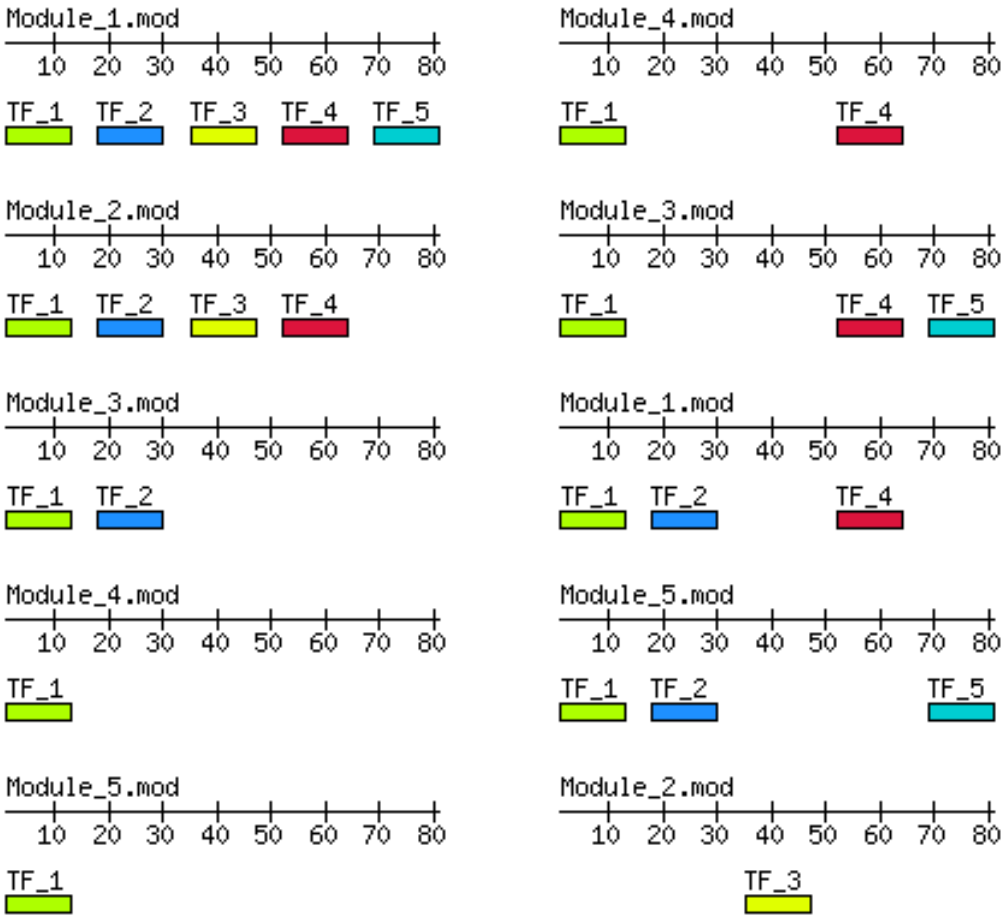


Figure 5.2: Examples for binding site distributions that are either sorted or randomly generated.

given distribution. Eventually `aln2ini.pl` creates a "distribution file" with all required parameters set.

Create set of regulatory modules

Given the distribution file generated by `aln2ini.pl` we create artificial modules containing the TFBSs. Two different programs perform this step. Whereas `distributeRandomSites.pl` iterates through the given TFBSs and chooses randomly a module to which the current motif is added, `distributeSortedSites.pl` iterates through the modules and adds each of the remaining motifs from the distribution file. See figure 5.2 for an example output of both programs. Both

programs output a table containing the positions of the individual sites in the sequences and write the created modules into `.mod` files.

Visualize the resulting arrangement

Given the output of the above two programs, we can create an image of the created module set with `renderDistribution.pl`, which uses the `bioperl`[1] module for drawing.

Creating promoter sequences

The artificial promoter sequences are based on a real life template. The upstream region of the *Homo sapiens* beta-actin (ACTB) gene (EPD accession number EP17045) ranging from -1000 to +1 relative to the TSS, is used as a template for dinucleotide frequencies (overall GC-content is 65.80%). The promoter sequences are created with the `rndgnm` program. It takes a DNA sequence as input, calculates the dinucleotide frequencies and outputs a new random sequence with a specified length and nucleotide frequencies similar to the template DNA. Sequences created with this method are unlikely to contain many false positive hits with a high match score, because of the different nucleotide frequencies between motif and promoter sequences.

5.3.2 Results

Before generating artificial data sets we need to devise the different scenarios which are run against to the `bbq` program. Following assumptions must be considered about the distribution of TFBSs in promoter regions: In the order of 10 different TFs occur in the CRMs of co-regulated genes. TFBSs are more or less randomly distributed within the modules, which essentially means, that not all TFBSs occur in every sequence, the density of TFBSs varies greatly among the promoters and a small set of motifs can be found in the majority of the promoter sequences. These scenarios can easily be played through within our framework.

[1] `http://www.bioperl.org`

Basic scenario

First we create a small set of five modules, with a few binding sites with decreasing occurrence count. The motifs are distributed with `genSortedModule.pl`. See figure 5.3 for the actual distribution of the motifs. In this test scenario we only

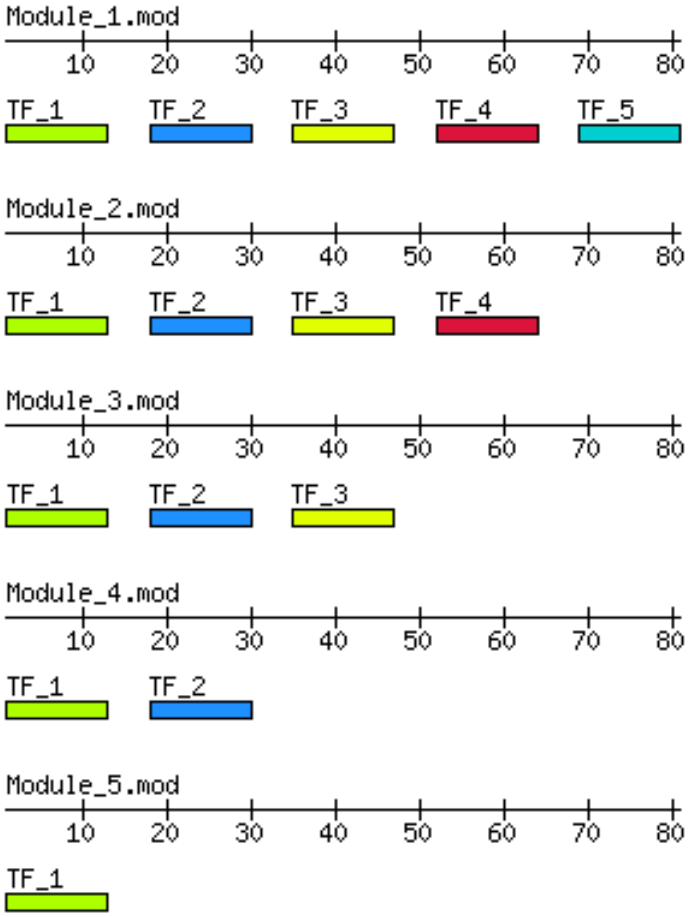


Figure 5.3: The figure shows five promoter modules, each containg a different set of sites. Whereas TF_1 occurs in all modules, TF_2 is contained in only four modules and so forth.

use unweighted scoring functions, because we want to check the basic functionality of the Tanimoto variants first. We search for modules with sizes from 2 up to 5. The window length is successively increased by the size of a single motif. Table 5.14 shows the scores for the top scoring modules of sizes 1 to 5. As expected, only combinations which occur in most sequences have the best scores within all

size	best module	T1	T2	T3
5	{TF_1,...,TF_5}	2.283	0.600	7.832
4	{TF_1,...,TF_4}	2.466	0.700	7.625
3	{TF_1,...,TF_3}	2.300	0.800	5.777
2	{TF_1, TF_2}	1.800	0.900	3.250
1	{TF_1}	1.000	1.000	1.000

Table 5.14: Scores of the top scoring candidate modules of sizes with Tanimoto variants 1, 2 and 3 accordingly.

3 variants. In the first case with single element modules, TF_1 gets the best score by all three Tanimoto variants. Note that each single sequence the `bbq` algorithm allows for cells in the arrangement containing only the outermost motifs. This is due to method of how the colored intervals are constructed and translated into the arrangement of cells. Therefore the cell containing only TF_1 scores best in each sequence resulting in an overall score of 1.0. With Tanimoto variant 2 the overall score decreases with increasing module size. This is due the equal weighting of elements common and uncommon in both compared sets. On the other hand, the overall scores increase with growing module size, when using Tanimoto variant 3.

Full scenario

After verifying the behavior of the different scoring functions, especially the preference for large modules of Tanimoto variant 3, the next scenario contains many different modules which are distributed over several sequences. We create two different sets of modules. One set contains 5 different motifs that occur in clustered in 5 modules. Two factors are designated to be contained in all modules, and additionally 3 other motifs are distributed randomly. The second module set contains 15 different motifs, which are randomly distributed among 16 modules. We call the former set the *positive* set and the latter *negative* set. Next we create 5 promoter sequences, each containing one module from the positive set and two module from the negative set. Additional two promoters with each 3 modules from the negative set are added. The 5 sequences are complemented by two additional sequences containing three negative modules each.

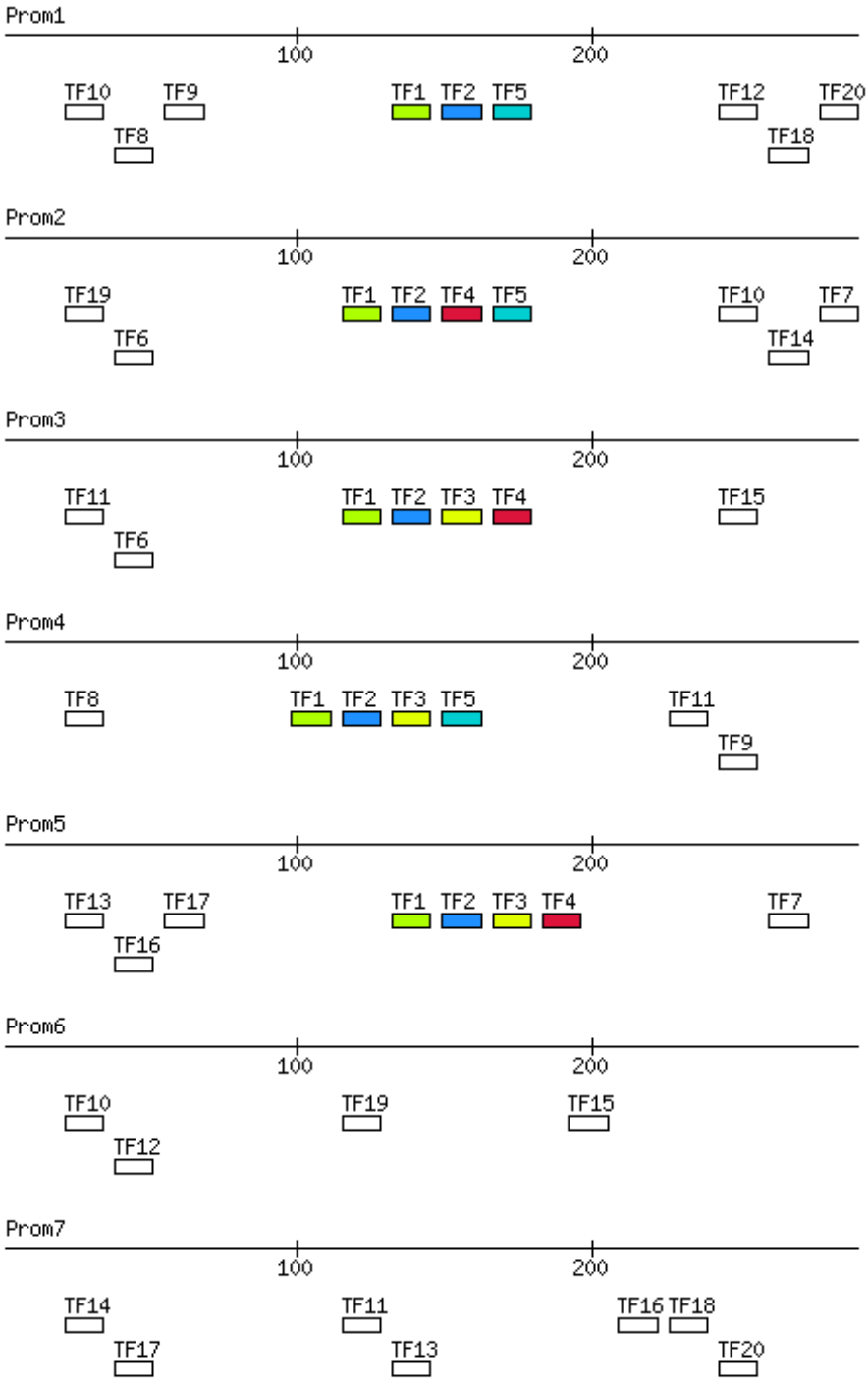


Figure 5.4: Distribution of TFBSs among 6 artificial promoter sequences. The highlighted motifs represent motifs belonging to a *cis*-regulatory module.

`bbq` was run against all 7 artificial promoter sequences and the top scoring modules were evaluated. All three Tanimoto variants, module sizes of 1 to 5 and a window length of 100 were chosen as parameters . Table 5.15 shows the top 3 modules found with each parameter combination.
During generation of the artificial modules, both TF1 and TF2 were placed in each of the 5 modules. They are complemented by TF3, TF4 and TF5, each occurring in three modules. All other factors TF6 to TF20 are randomly distributed in several modules upstream and downstream of the 5 positive modules, serving as noise elements.
All three Tanimoto variants detected the combination {TF1, TF2} as the top scoring candidate module of size two. It is the only module occurring in five of the seven sequences. With modules of size 3, the combination {TF1, TF2, TF3} scores best in variants 2 and 3. In variant 1 its score is equal to the combination of {TF1, TF2} with TF4 and TF5 respectively. Those 3 combinations all occur in three sequences each. Only the combination of {TF1, TF2, TF3, TF4} can be found in two promoters. All three Tanimoto variants find this module as the best scoring candidate. Looking at modules of size five, again the combination of sites from the positive set is returned as top result by all three Tanimoto variants.
These results confirm the expected behavior of the extended `bbq` algorithm. In contrast to the data sets used in the preceding chapters, here all three Tanimoto variants happen to yield the same results, at least in the best scored candidates.

Weighted vs. unweighted scores

All three scoring functions $\mathrm{tnm}_i(A, B)$ come in a weighted version (see section 4.4.2), in which the elements of the then fuzzy set B have an associated degree of membership to the set which denotes the match score of a putative binding site to a matrix profile. The purpose of the weighted scores is to emphasize the importance of elements which are more similar to the matrix profile. We want to favour such matches over matches with lower scores. This approach may find modules which occur less often then others, but have stronger TFBS matches.
Two different mutation rates where applied to respectively two groups of three motifs. Alignments for the *degenerate* motifs TF_1, TF_2 and TF_3 were created using a mutation rate of 0.4. The set of conserved motifs TF_4, TF_5 and TF_6

size	T var.	motifs	score	occ.
2	1	TF1,TF2	1.42857	5
		TF1,TF3	1	3
		TF2,TF3	1	3
2	2	TF1,TF2	0.714286	5
		TF2,TF5	0.428571	3
		TF1,TF6	0.428571	2
2	3	TF1,TF2	2.85714	5
		TF1,TF6	1.19048	2
		TF3,TF4	1.1746	2
3	1	TF1,TF2,TF3	1.85714	3
		TF1,TF2,TF4	1.85714	3
		TF1,TF2,TF5	1.85714	3
3	2	TF1,TF2,TF3	0.619048	3
		TF1,TF2,TF6	0.571429	2
		TF1,TF17,TF2	0.559524	1
3	3	TF1,TF2,TF3	4.36508	3
		TF1,TF2,TF6	3.33333	2
		TF1,TF2,TF5	3.24008	3
4	1	TF1,TF2,TF3,TF4	2.14286	2
		TF1,TF2,TF3,TF5	2.07143	1
		TF2,TF3,TF4,TF5	1.85714	0
4	2	TF1,TF2,TF3,TF4	0.571429	2
		TF1,TF2,TF3,TF5	0.55	1
		TF1,TF2,TF4,TF5	0.507143	1
4	3	TF1,TF2,TF3,TF4	6.16071	2
		TF1,TF2,TF3,TF5	4.91821	1
		TF1,TF2,TF4,TF5	4.3975	1
5	1	TF1,TF2,TF3,TF4,TF5	2.5	0
		TF1,TF2,TF3,TF4,TF9	1.78571	0
		TF1,TF10,TF2,TF3,TF4	1.70238	0
5	2	TF1,TF2,TF3,TF4,TF5	0.542857	0
		TF1,TF10,TF2,TF3,TF4	0.485714	0
		TF1,TF14,TF2,TF3,TF4	0.485714	0
5	3	TF1,TF2,TF3,TF4,TF5	6.31429	0
		TF1,TF2,TF3,TF4,TF9	4.31429	0
		TF1,TF10,TF2,TF3,TF4	3.94857	0

Table 5.15: Top scoring candidate modules of the full scenario with artificial data sets.

contains alignments with a much lower mutation rate of 0.05. Instances of TF_1 to TF_5 were each inserted in three promoter modules. Only TF_6 was put in only two sequences. See figure 5.5 for the distribution of TFBSs and their respective match scores recognized by `bbq`. We expect that TF_6 will be found

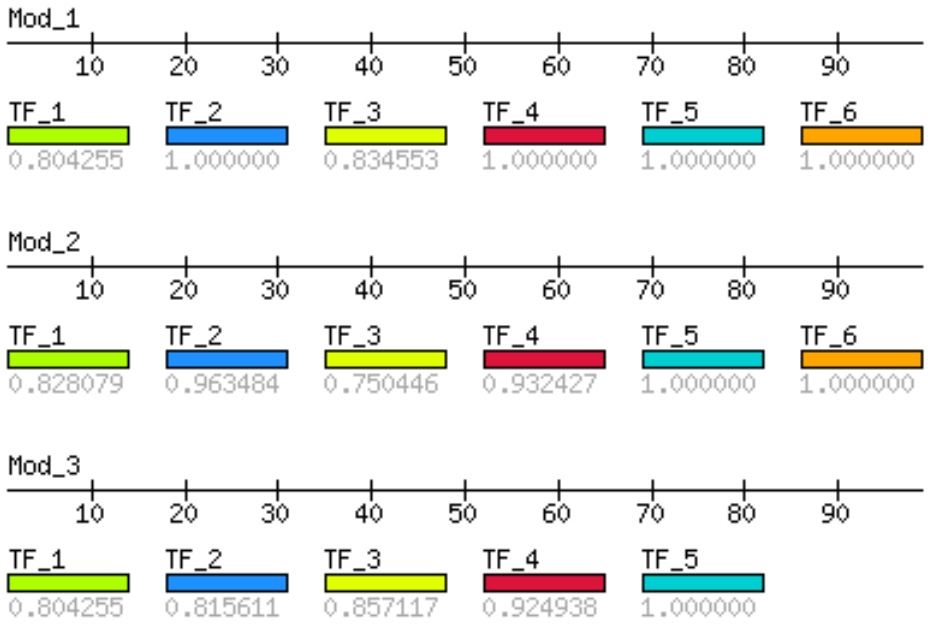


Figure 5.5: Distribution of motif occurrences and respective match scores.

in the best CRMs detected when using weighted scores, despite occurring less often than the other five motifs. At first we run `bbq` with unweighted Tanimoto scores, a match threshold of 0.7, Tanimoto variant 3 and a window length of 15nt. Whereas all modules of size 1 containing each of the TFs occurring in all three sequences have the maximum score of 1.0, the module yielding TF_6 only gets a score of 0.666667. In the weighted variant, the top modules of size one show preference for the sites which match well in all three sequences. The best hit was TF_5 with an overall score of 1 (each sequence score is 1.0 and the sum of all sequence scores is divided by the number of sequences). Runner-ups are TF_4 with 0.829263, TF_2 with 0.768087 already followed by TF_6 with 0.666667. TF_3 and TF_1 have scores below 0.45. In contrast to the unweighted case, where TF_1 to TF_5 shared the first place, now the match scores of each site contribute to the overall ranking. Even TF_6 is the fourth best result. Next, modules of

size two were searched in a window length of 30, again starting with unweighted Tanimoto variant 3. The modules {TF_1, TF_2}, {TF_2, TF_3}, {TF_3, TF_4} and {TF_4, TF_5} have each a score of 4.0, whereas {TF_5, TF_6} has only a score of 2.75. Using the weighted scores, the top module was found to be {TF_4, TF_5} with a score of 3.63956 followed by {TF_5, TF_6} with 2.75. The only 1.0 match scores of all these sites have a high impact on the overall ranking, boosting {TF_5, TF_6} to the second place. In the next run, we look for modules of size 3 and resize the window length to 50. Whereas {TF_1, TF_2, TF_3}, {TF_2, TF_3, TF_4} and {TF_3, TF_4, TF_5} have an top score of 9.0 using the unweighted scoring function, the combination of {TF_4, TF_5, TF_6} only gets a score of 6.59259. Using the weighted variant, {TF_3, TF_4, TF_5} is little better then {TF_4, TF_5, TF_6} with a score of 6.52851 compared to 6.24722.

From these results, it can be concluded, that the weighted Tanimoto variants have a high impact on the result ranking and in some scenarios make a ranking possible in the first place. While the overall score of the top modules varies significantly when the cardinality of the modules is small, the score differences get narrower with increasing cardinality of the modules.

Chapter 6

Discussion

6.1 Results

The elaborate interactions of transcription factors within the promoter of a gene enable an effective regulation of transcription in the first place. Thus search for *cis*-regulatory modules in promoter regions remains an important field of current biological research. The assumptions underlying the Best-Barbecue-Problem resemble the properties of actual promoters observed in studied genomic sequences. Transcription factor binding sites do not necessarily occur in the same order and orientation in regulatory modules of co-regulated genes. Furthermore the combinations of sites must not be exactly the same among similar modules. When searching for regulatory modules consisting of several binding sites which reside among certain interval, these conditions must be considered.
This search problem can be formalized by the means of a combinatorial optimization problem, the Best-Barbecue-Problem. The name follows an illustration of the problem by an analog in the world of BBQ. Unfortunately the Best-Barbecue-Problem has been proven to be *NP*-complete. But the parameters in the exponent are usually small in real life instances of the problem. The `bbq` program solves the Best-Barbecue-Problem with two different algorithms. It takes as input several genomic sequences and a set of transcription factor binding sites, either represented as consensus sequences or in the form of a position count matrix. Then `bbq` searches for putative occurrences of binding sites and organizes them into an arrangement of cells which reflect the window length. Then these cells are

scanned for subsets of factors which occur in a cell in each sequence. This approach always finds "correct" results, i.e. the factors in the found modules are contained in *all* sequences.
It has been shown, that this approach performs unsatisfactory under certain conditions. Most data sets typically contain much divergent sets of binding sites. Under these conditions the `bbq` approach is restricted to the lowest common denominator of all sequences, though some of the promoter sequences might share a larger set of binding sites.
To overcome these limitations, the search algorithm was modified and extended by a scoring function. The Tanimoto distance is a similarity measure for comparing sets. The extended search algorithm enumerates all possible combinations of binding sites and scores each against the cells in the arrangement system. The candidate module with the highest similarity is returned. Three different variants of the Tanimoto scores are implemented each in a weighted and unweighted version.
The performance of the new `bbq` algorithm has been approved successfully on two real life data sets. One set features a well studied collection of promoters of genes expressed in cells in the muscle. In contrast to the standard `bbq` algorithm, the Tanimoto version successfully finds regulatory modules common in many but not all sequences in this set. The second collection contains promoters of beta-actin genes, which were collected from multiple databases. Again, the Tanimoto variant finds clusters of binding sites which are currently associated with regulation of these genes. Additionally three artificial data sets were created for testing various features of the Tanimoto scores.
With the Tanimoto scores, the restrictions on the selection of input sequences are much reduced. It is now possible to scan larger collections of promoters and search for largest possible regulatory modules which are occur in the majority of the promoters.

6.2 Further Work

Still an issue, which is not specific to the `bbq` program, is the matching of PWMs against genomic sequences. While methods exist for accessing the power and

significance of PWMs, the number of these motifs is growing fast. Since `bbq` is designated to search for novel regulatory modules, too much false positive hits decrease the specificity of the search. Therefore for a better BBQ experience, the input PWMs need to be carefully selected. Additionally choosing a high match threshold is advisable. But this approach would probably reduce the sensitivity of the matches. To circumvent these restrictions, one should consider the implementation of novel matching algorithms, for example P-Match, even though it relies not only on PWMs, but also requires an alignment of the TFBS motifs (see 2.2.1).

In some scenarios it might desirable to only consider clusters of TFBSs which do not overlap, e.g. for studying cooperatively acting factors. This can be accomplished by drawing the reversed overlap graph for each cell in the arrangement system. These graphs are then searched for maximum cliques, which is again an NP-complete problem.

Appendix A

Manual - bbq

NAME

bbq – discovering clusters of transcription factor binding sites

SYNOPSIS

bbq [options] <length> <motif-file> <seq-file1> ... <seq-fileK>

DESCRIPTION

bbq is a command-line tool for discovering clusters of transcription factor binding sites that occur simultaneously in several genomes. Finding such clusters – sometimes also referred to as *cis-regulatory modules* – is done in a multiple-alignment-like fashion by solving a certain combinatorial and geometric optimization problem, the so-called Best-Barbecue-Problem (explaining the name bbq). As opposed to classical (typically dynamic programming based) alignment procedures, the order of the binding sites' occurences can be arbitrarily shuffled, so that bbq is the result of developing completely new algorithms.

MOTIF AND GENOME FILE FORMATS

The genome files <seq-file1> ... <seq-fileK> must be in FASTA format. Each file must not contain more than one sequence.
Transcription factor binding sites may either be given in the format of a consensus sequence or as a Position Specific Count Matrix (PCM).

While using consensus sequences `bbq` expects a pattern file containing all motif sequences. The pattern file is required to be in the following format:

```
<motif1-name>  <motif1-sequence>   m1
<motif2-name>  <motif2-sequence>   m2
...
<motifM-name>  <motifM-sequence>   m3
```

The first columns of each line contains the name of the motif (i.e., the corresponding binding site). The second column contains its DNA sequence, while the last column specifies the maximum number of mismatches that is tolerated for finding occurences of the motif.
The motif sequence may contain words over the 16-letter FASTA alphabet for nucleic acids:

A - adenosine	M - A C	B - G T C
C - cytidine	W - A T	D - G A T
G - guanine	S - G C	H - A C T
T - thymidine	R - G A	V - G C A
U - uridine	K - G T	N - A G C T
	Y - T C	

When using position count matrices, each matrix must be stored in a separate file. Additionally a text file containing the path to a directory with the matrix files is required (see example below).

OUTPUT FORMAT

The output is written to stdout (except for runtime information, which is written to stderr). Basic output is not intended to be human readable. Use -S to get more verbose textual output. A graphical representation of the found motif clusters in each genomic sequence can be obtained either in postscript (-P) or encapsulated postscript format (-E).

OPTIONS

–**g** S for grouping, where S is an integer indicating the length of the group prefix for fragment names;

–**h** N computing the best N matches while removing each found result from the input sequences before each step;

–**H** N computing the best N matches without modifying the arrangement system, only together with –**T**;

–**S** more verbose output so that it can be understood by a Stupid user :);

–**m** use multisets (i.e., multiple occurences in 1 cluster make a difference);

–**u** (unweighted) to switch off weighting and optimize for intersection cardinality;

–**w**t select weighting based on mismatches (t=m) or based on p-values (t=p);

–**t** T for thresholding, with T denoting a float threshold. All occurences whose weight (or intersection cardinality, in case -u was specified) exceeds T are printed.;

–**P** <**filename**> for postscript output to <filename>;

–**E** <**filename**> for encapsulated postscript (EPS) output to file <filename>;

–**p** display progress information;

–**r** include reverse complemented fragments so that both 3' and 5' occurences may be part of a cluster;

–**A**x with x=1 or x=2 for choosing Algorithm (A1) (default) or (A2), respectively;

–**W** x specify a treshold while matching position count matrices against the genomic sequences, x must be a real number between 0.0 and 1.0, defaults to 0.8;

–**T** x enables Tanimoto scoring, with integer <x> indicating the Tanimoto variant and must be either 1, 2 or 3;

–**D** x delta-bounded candidates are used, where <x> specifies the maximal difference between the candidate clusters and the cells occuring in the genomic sequences;

–**L** x only candidates with size x are tested, only together with –**T**;

–**I** x y only candidates with sizes from x to y are tested, only together with –**T**;

EXIT STATUS

bbq returns non zero if a failure occured, zero otherwise.

EXAMPLES

The motif file bsites.motifs contains following lines:

```
c414M2 GGCTGCGAA 1
c432M5 GATTTCCCGA 1
c106M1 GGCAGG 1
c196M2 AAGTAATTAGT 1
c196M3 TTCTCCTT 1
c196M4 TTATTGTC 1
```

The command line

```
bbq -wp -r -P bsites.ps 50 bsites.motifs GS1.fa GS2.fa GS3.fa
```

writes graphical output into bsites.ps for best clustered occurences of motifs from bsites.motifs in GS1.fa, GS2.fa and GS3.fa
When using position weight matrices, a text file is required which contains the path to the directory containing the matrix files, e.g. the file motifs.text yielding following line:

```
# matrices /home/user/path/to/matrices
```

In this directory each file must be in the following format:

```
3    0    0    9    0   13   12   13   13    3
3   10    3    0    0    0    0    0    0    1
2    2    0    0    0    0    0    0    0    6
3    0    9    3   13    0    0    0    0    1
```

Each row contains the frequency of a single nucleotide at each position. The rows correspond to nucleotides in the order A, C, G, T beginning at the first row.
The command

```
bbq -r -S -W 0.9 -T 3 -L 3 -H 3 300 motifs.txt *.fa
```

tells bbq to use these matrices and apply a match treshold of 0.9. Furthermore the Tanimoto scoring scheme in variant 3 is applied (–T 3) and only clusters of size three (–L 3) are evaluated.

AUTHORS

Axel Mosig, Bioinformatics Group, University of Leipzig, Germany.
e-mail: axel@bioinf.uni-leipzig.de
Peter Menzel, Bioinformatics Group, University of Leipzig, Germany.
e-mail: peterm@bioinf.uni-leipzig.de

URL

http://www.bioinf.uni-leipzig.de/Software/bbq/

Bibliography

[1] D. N. Arnosti and M. M. Kulkarni. Transcriptional enhancers: Intelligent enhanceosomes or flexible billboards? *Journal of Cellular Biochemistry*, 94(5):890–898, 2005.

[2] T. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology*, pages 28–36. AAAI Press, 1994.

[3] C. M. Bergman, J. W. Carlson, and S. E. Celniker. Drosophila DNase I footprint database: a systematic genome annotation of transcription factor binding sites in the fruitfly, Drosophila melanogaster. *Bioinformatics*, 21(8):1747–1749, 2005.

[4] T. Berners-Lee, R. Cailliau, J.-F. Groff, and B. Pollermann. World-Wide Web: An information infrastructure for high-energy physics. In *Proceedings of the Workshop on Software Engineering, Artificial Intelligence and Expert Systems for High Energy and Nuclear Physics*, 1992.

[5] E. M. Blackwood and J. T. Kadonaga. Going the distance: A current view of enhancer action. *Going the Distance: A Current View of Enhancer ActionScience*, 281(5373):60 – 63, 1998.

[6] M. Blanchette, A. R. Bataille, X. Chen, C. Poitras, J. Laganiere, C. Lefebvre, G. Deblois, V. Giguere, V. Ferretti, D. Bergeron, B. Coulombe, and F. Robert. Genome-wide computational prediction of transcriptional regulatory modules reveals new insights into human gene expression. *Genome Res.*, 16(5):656–668, 2006.

[7] M. Blanchette, W. J. Kent, C. Riemer, L. Elnitski, A. F. Smit, K. M. Roskin, R. Baertsch, K. Rosenbloom, H. Clawson, E. D. Green, D. Haussler, and W. Miller. Aligning Multiple Genomic Sequences With the Threaded Blockset Aligner. *Genome Res.*, 14(4):708–715, 2004.

[8] T. Brown. *Moderne Genetik*. Spektrum Akademischer Verlag, 1999.

[9] M. Capelson and V. G. Corces. Boundary elements and nuclear organization. *Biology of the Cell*, 96(8):617–629, Oct. 2004.

[10] M. G. Caprara and T. W. Nilsen. Rna: Versatility in form and function. *Nat Struct Mol Biol*, 7(10):831–833, Oct. 2000.

[11] R. Cavin Perier, T. Junier, and P. Bucher. The Eukaryotic Promoter Database EPD. *Nucl. Acids Res.*, 26(1):353–357, 1998.

[12] D. S. Chekmenev, C. Haid, and A. E. Kel. P-match: transcription factor binding site search by combining patterns and weight matrices. *Nucl. Acids Res.*, 33(2):W432–437, 2005.

[13] F. Crick. On protein synthesis. In *Symp. Soc. Exp. Biol. XII*, pages 139–163, 1958.

[14] E. Davidson. *Genomic Regulatory Systems: Development and Evolution*. Academic Press, New York, 2001.

[15] R. V. Davuluri, H. Sun, S. K. Palaniswamy, N. Matthews, C. Molina, M. Kurtz, and E. Grotewold. Agris: Arabidopsis gene regulatory information server, an information resource of arabidopsis cis-regulatory elements and transcription factors. *BMC Bioinformatics*, 4:25, 2003.

[16] S. R. Eddy. Non-coding rna genes and the modern rna world. *Nat Rev Genet*, 2(12):919–929, Dec. 2001.

[17] J. Fickett. Quantitative discrimination of MEF2 sites. *Mol. Cell. Biol.*, 16(1):437–441, 1996.

[18] K. Fresh, K. Quandt, and T. Werner. Muscle actin genes: a first step towards computational classification of tissue specific promoters. *In Silico Biol.*, 1(1):29–38, 1998.

[19] M. Y. Galperin. The Molecular Biology Database Collection: 2006 update. *Nucl. Acids Res.*, 34(1):D3–5, 2006.

[20] G. G. Gary and V. Strelets. FlyBase: anatomical data, images and queries. *Nucl. Acids Res.*, 34(1):D484–488, 2006.

[21] D. Greenbaum, R. Jansen, and M. Gerstein. Analysis of mRNA expression and protein abundance data: an approach for the comparison of the enrichment of features in the cellular population of proteins and transcripts . *Bioinformatics*, 18(4):585–596, 2002.

[22] D. GuhaThakurta and G. D. Stormo. Identifying target sites for cooperatively binding factors. *Bioinformatics*, 17(7):608–621, 2001.

[23] G. Hertz and G. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7):563–577, 1999.

[24] J. D. Hughes, P. W. Estep, S. Tavazoie, and G. M. Church. Computational identification of cis-regulatory elements associated with groups of functionally related genes in saccharomyces cerevisiae. *J Mol Biol*, 296(5):1205–14, 2000.

[25] N. Hunter. Prion diseases and the central dogma of molecular biology. *Trends in Microbiology*, 7(7):265–266, 1999.

[26] Jan Kohlmann and Klaus-Heinrich Röhm. *Taschenatlas der Biochemie.* Georg Thieme Verlag, 2003.

[27] C. Kanz, P. Aldebert, N. Althorpe, W. Baker, A. Baldwin, K. Bates, P. Browne, A. van den Broek, M. Castro, G. Cochrane, K. Duggan, R. Eberhardt, N. Faruque, J. Gamble, F. G. Diez, N. Harte, T. Kulikova, Q. Lin, V. Lombard, R. Lopez, R. Mancuso, M. McHale, F. Nardone, V. Silventoinen, S. Sobhany, P. Stoehr, M. A. Tuli, K. Tzouvara, R. Vaughan, D. Wu,

W. Zhu, and R. Apweiler. The EMBL Nucleotide Sequence Database. *Nucl. Acids Res.*, 33(1):D29–33, 2005.

[28] A. Kel, E. Gossling, I. Reuter, E. Cheremushkin, O. Kel-Margoulis, and E. Wingender. Match: a tool for searching transcription factor binding sites in dna sequences. *Nucl. Acids Res.*, 31(13):3576–3579, 2003.

[29] O. V. Kel-Margoulis, A. G. Romashchenko, N. A. Kolchanov, E. Wingender, and A. E. Kel. COMPEL: a database on composite regulatory elements providing combinatorial transcriptional regulation. *Nucl. Acids Res.*, 28(1):311–315, 2000.

[30] D. E. Knuth. *The Art of Computer Programming Voulme 4A Fascicle 3.* Addison-Wesley, 2005.

[31] N. A. Kolchanov, E. V. Ignatieva, E. A. Ananko, O. A. Podkolodnaya, I. L. Stepanenko, T. I. Merkulova, M. A. Pozdnyakov, N. L. Podkolodny, A. N. Naumochkin, and A. G. Romashchenko. Transcription regulatory regions database (trrd): its status in 2002. *Nucl. Acids Res.*, 30(1):312–317, 2002.

[32] B. Lenhard, A. Sandelin, L. Mendoza, P. Engstrom, N. Jareborg, and W. Wasserman. Identification of conserved regulatory elements by comparative genome analysis. *Journal of Biology*, 2(2):13, 2003.

[33] B. Lenhard and W. W. Wasserman. TFBS: Computational framework for transcription factor binding site analysis. *Bioinformatics*, 18(8):1135–1136, 2002.

[34] M. Levine and R. Tjian. Transcription regulation and animal diversity. *Nature*, 424(6945):147–151, July 2003.

[35] G. G. Loots and I. Ovcharenko. rvista 2.0: evolutionary analysis of transcription factor binding sites. *Nucl. Acids Res.*, 32:217–221, 2004.

[36] B. Modrek and C. Lee. A genomic view of alternative splicing. *Nat Genet*, 30(1):13–19, Jan. 2002.

[37] A. Mosig, T. Biyikoglu, S. J. Prohaska, and P. F. Stadler. Detecting phylogenetic footprint clusters by optimizing barbeques. Technical report, Max Plank Institute for Mathematics in Sciences, 2005.

[38] I. Ovcharenko, M. A. Nobrega, G. G. Loots, and L. Stubbs. ECR Browser: a tool for visualizing and accessing data from comparisons of multiple vertebrate genomes. *Nucl. Acids Res.*, 32(2):W280–286, 2004.

[39] S. K. Palaniswamy, V. X. Jin, H. Sun, and R. V. Davuluri. OMGProm: a database of orthologous mammalian gene promoters. *Bioinformatics*, 21(6):835–836, 2005.

[40] P. Perco, A. Kainz, G. Mayer, A. Lukas, R. Oberbauer, and B. Mayer. Detection of coregulation in differential gene expression profiles. *Biosystems*, 82(3):235–247, Dec. 2005.

[41] T. T. Pohar, H. Sun, and R. V. Davuluri. HemoPDB: Hematopoiesis Promoter Database, an information resource of transcriptional regulation in blood cell development. *Nucl. Acids Res.*, 32(1):D86–90, 2004.

[42] D. Pribnow. Nucleotide Sequence of an RNA Polymerase Binding Site at an Early T7 Promoter. *PNAS*, 72(3):784–788, 1975.

[43] Prohaska. Regulatory signals in genomic sequences. Technical report, IZBI, 2005.

[44] R. Pudimat, E. Schukat-Talamazzini, and R. Backofen. A multiple-feature framework for modelling and predicting transcription factor binding sites. *Bioinformatics*, 21(14):3082–8, 2005.

[45] K. Quandt, K. Frech, H. Karas, E. Wingender, and T. Werner. Matlnd and Matlnspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data. *Nucl. Acids Res.*, 23(23):4878–4884, 1995.

[46] S. Rahmann, T. Mueller, and M. Vingron. On the power of profiles for transcription factor binding site detection. *Statistical Applications in Genetics and Molecular Biology*, 2(1):22, 2003.

[47] A. Sandelin, W. Alkema, P. Engstrom, W. W. Wasserman, and B. Lenhard. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucl. Acids Res.*, 32(1):D91–94, 2004.

[48] C. D. Schmid, R. Perier, V. Praz, and P. Bucher. EPD in its twentieth year: towards complete promoter coverage of selected model organisms. *Nucl. Acids Res.*, 34(1):D82–85, 2006.

[49] T. D. Schneider and R. Stephens. Sequence logos: a new way to display consensus sequences. *Nucl. Acids Res.*, 18(20):6097–6100, 1990.

[50] R. Sharan, A. Ben-Hur, G. G. Loots, and I. Ovcharenko. CREME: Cis-Regulatory Module Explorer for the human genome. *Nucl. Acids Res.*, 32(2):W253–256, 2004.

[51] R. Sharan, I. Ovcharenko, A. Ben-Hur, and R. M. Karp. Creme: a framework for identifying cis-regulatory modules in human-mouse conserved segments. *Bioinformatics*, 19(1):283–291, 2003.

[52] S. T. Smale and J. T. Kadonaga. The rna polymerase ii core promoter. *Annual Review of Biochemistry*, 72:449–479, 2003.

[53] G. D. Stormo. Dna binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23, 2000.

[54] B. D. Temin H.M. Rna-directed dna synthesis and rna tumor viruses. *Adv Virus Res.*, 17:129–186, 1972.

[55] M. Tompa, N. Li, T. L. Bailey, G. M. Church, B. De Moor, E. Eskin, A. V. Favorov, M. C. Frith, Y. Fu, W. J. Kent, V. J. Makeev, A. A. Mironov, W. S. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, J. van Helden, M. Vandenbogaert, Z. Weng, C. Workman, C. Ye, and Z. Zhu. Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotech*, 23(1):137–144, Jan. 2005.

[56] D. Vlieghe, A. Sandelin, P. J. De Bleser, K. Vleminckx, W. W. Wasserman, F. van Roy, and B. Lenhard. A new generation of JASPAR, the open-access

repository for transcription factor binding site profiles. *Nucl. Acids Res.*, 34(1):D95–97, 2006.

[57] W. W. Wasserman and J. W. Fickett. Identification of regulatory regions which confer muscle-specific gene expression. *Journal of Molecular Biology*, 278(1):167–181, Apr. 1998.

[58] W. W. Wasserman and A. Sandelin. Applied bioinformatics for the identification of regulatory elements. *Nat Rev Genet*, 5(4):276–287, Apr. 2004.

[59] E. Wingender. Compilation of transcription regulating proteins. *Nucleic Acids Res.*, 16(5):1879–1902, 1988.

[60] E. Wingender. Classification scheme of eukaryotic transcription factors. *Molekularnaya Biologiya*, 31:584–600, 1997.

[61] E. Wingender, X. Chen, E. Fricke, R. Geffers, R. Hehl, I. Liebich, M. Krull, V. Matys, H. Michael, R. Ohnhauser, M. Prubeta, F. Schacherer, S. Thiele, and S. Urbach. The TRANSFAC system on gene expression regulation. *Nucl. Acids Res.*, 29(1):281–283, 2001.

[62] X. Xie, J. Lu, E. J. Kulbokas, T. R. Golub, V. Mootha, K. Lindblad-Toh, E. S. Lander, and M. Kellis. Systematic discovery of regulatory motifs in human promoters and 3[prime] utrs by comparison of several mammals. *Nature*, 434(7031):338–345, Mar. 2005.

[63] J. Zhu and M. Zhang. SCPD: a promoter database of the yeast Saccharomyces cerevisiae. *Bioinformatics*, 15(7):607–611, 1999.

Selbstständigkeitserklärung

Ich versichere hiermit, die vorliegende Diplomarbeit selbstständig und nur unter Verwendung der angegebenen Hilfsmittel und Literatur angefertigt zu haben.

Peter Menzel

Leipzig, den 23. Oktober 2006