Transcriptional Regulatory Elements

Detection and Evolutionary Analysis

Der Fakultät für Mathematik und Informatik
der Universität Leipzig
eingereichte

Dissertation

zur Erlangung des akademischen Grades

Doctor rerum naturalium
(Dr. rer. nat.)

im Fachgebiet

Informatik

vorgelegt

von Diplom Bioinformatiker Wolfgang Otto

geboren am 30. Dezember 1978 in Jena

Die Annahme der Dissertation haben empfohlen:

1. Prof. Dr. Peter F. Stadler (Universität Leipzig, Deutschland)
2. Prof. Dr. Burkhard Morgenstern (Universität Göttingen, Deutschland)

Die Verleihung des akademischen Grades erfolgt mit Bestehen der Verteitigung am 29. November 2011 mit dem Gesamtprädikat *magna cum laude*.

Für Sandra

Contents

Abstract

A major challenge in life sciences is the understanding of mechanisms that regulate the expression of genes. An important step towards this goal is the ability to identify transcriptional regulatory elements like binding sites for transcription factors. In computational biology, a popular approach for this task is comparative sequence analysis using both distantly as well as closely related species. Although this method has successfully identified conserved regulatory regions, the majority of binding sites can change rapidly even between closely related species. This makes it difficult to detect them using DNA sequences alone. In this thesis, we introduce two new approaches for the detection and evolutionary analysis of transcriptional elements that consider the challenges of binding site turnover.

In the first part, we develop a method for detecting homologous motifs in a given set of sequences in order to obtain evidence for evolutionary events and turnover. Based on a detailed theoretical scaffold, we develop a simple, but effective and efficient heuristic for assembling local pairwise sequence alignments into a local multiple sequence alignment. This kind of multiple alignment only contains conserved motifs represented in columns which satisfy the order implied by the underlying sequences. By favoring motifs that are contained in a great range of sequences, our method is additionally able to detect even small conserved motifs. Furthermore, the calculation of the initial local pairwise alignments is generic. This allows the use of fast heuristic methods in case of large data sets while exact alignment programs can be used for small data sets where detailed information is needed. Application to artificial as well as biological data sets demonstrate the capabilities of our algorithm.

In the second part, we propose a conceptually simple, but mathematically non-trivial, phenomenological model for the binding site turnover at a genomic locus. The model is based on the assumption that binding sites have a constant rate of origination and a constant decay rate per binding site. The elementary derivation of the transient probability distribution is affirmed by simulations of sequence evolution as well as biological data. Based on the derived distribution, we develop a phenomenological model of binding site number dynamics in order to detect changes in selective constraints acting on transcription factor binding sites. Using

a maximum likelihood implementation as well as exploratory data analysis, we show the functionality of the model by identifying functionally important changes in the evolutionary turnover rates on biological data.

Each part of this thesis leads to the development of a new program. While `Tracker` allows the computation of conserved homologous motifs and their representation in a local multiple alignment, `Creto` determines the evolutionary turnover rates for arbitrary clades of a phylogenetic tree with given binding site numbers at the final taxa. Both software tools are freely available to the scientific community for further research in this important and exciting field.

Acknowledgments

First of all, I want to thank my supervisor Peter F. Stadler for giving me the great opportunity to work on this interesting research topic. It was a challenging pleasure to develop and improve the algorithms for the footprint detection. He let me develop my own ideas and gave me support whenever needed. I also like to thank Sonja J. Prohaska for all the interesting and helpful discussions about the biological aspects of genes, transcription and regulation.

I am also very grateful for the scientific encouragement of my second adviser Günter P. Wagner. His enthusiasm as well as his valuable support are remarkable. Special thanks go to Charlie whose cheerful character and generosity made my research project in New Haven a real pleasure. I also thank Vinny for all the help and fun. I really enjoyed the uncomplicated and amicable atmosphere at the "wagnerlab". Further, I'm very grateful for the collaborations with Rolf Backofen and Sebastian Will in Freiburg.

Special acknowledgments go to Petra who is not only responsible for the maintenance of the lab but who also always has a smile, words of encouragement and great suggestions for bike tours. I thank Jens for all the technical support of highest quality and the mysterious chocolate at some days. I also will miss all the other people in the lab: Anne, Axel, Berni, Christian, Corinna, David, Dom, Fakteh, Gunnar, Gruber, Guido, Henry, Jan, Jana, Jörg, Kristin, Konstantin, Lydia, Marc, Markus, Maribel, Mario, Steffi, Steve and Sven. Especially, I thank my great roommates Alex and Stephie for making the lab a wonderful home. I really enjoyed the time at the "Bierinformatik". What a nice place to work!

My PhD project was supported by a PhD Fellowship of the Konrad-Adenauer-Foundation and a PhD Fellowship of the Max-Planck-Society for Mathematics in the Sciences. I'm really grateful for all the support, the awesome seminars and the great people I met.

Finally, I would like to thank the most important people in my life. Without the love and support from my family and friends, this work would not exist. Thanks Mom, Dad and Evi. Thanks Andi, Andreas, Eva, Fefa, Heike, Kristin, Liza, Marlene, Peter, Sandra and all the great people I swim, bike and run with. Linda deserves special mention for proofreading the entire thesis. This was extremely valuable - thanks so much! To all of you, I express my deep appreciation.

CHAPTER 1

Introduction

> "Once we were blobs in the sea, and then fishes, and then lizards and rats, and then monkeys, and hundreds of things in between. This hand was once a fin, this hand once had claws! In my human mouth I have the pointy teeth of a wolf and the chisel teeth of a rabbit and the grinding teeth of a cow! Our blood is as salty as the sea we used to live in! When we're frightened, the hair on our skin stands up, just like it did when we had fur. We ARE history! Everything we've ever been on the way to becoming us, we still are. Would you like the rest of the story? I'm made up of the memories of my parents and my grandparents, all my ancestors. They're in the way I look, in the color of my hair. And I'm made up of everyone I've ever met who's changed the way I think.".
>
> *A Hat Full of Sky*
> Terry Pratchett

The diversity of form in life has interested humanity for thousands of years. Since Plato's first idealistic concepts, in which all natural phenomena are imperfect representations of the true essence of an ideal unseen world, and Aristotle's scala naturae, a chain-like series of links in the progress from most imperfect to most perfect, see Figure 1.1, life sciences have made progresses.

By the end of the 1950s, it was clear that causal differences between the body plans of animals like insects, mammals and fishes are somehow encoded in their genomes (Beadle and Tatum, 1941; Avery *et al.*, 1944; Watson and Crick, 1953). But this fact alone does not provide the mechanisms by which the various life forms develop or emerge during evolution. Today we know more about how genomes actually work, but questions of where in the genome the

Figure 1.1: The Scala Naturae of Aristotle by Mark Dion (1993). This ladder of nature starts with the imperfect inanimate matter and ends with the most perfect form, the human.

causal differences responsible for morphological diversity reside and how exactly they function remains.

A large part of the answer lies in the gene control circuitry encoded in the sequence, the structure and the functional organization of the DNA. The regulatory interactions mandated by this circuitry determine whether a gene is expressed in a cell through out developmental space and time and, if so, at what amplitude. Based on this regulation, changes in the DNA over time and geographical space are thought to be responsible for the evolution of species.

In this thesis we introduce new computational approaches for detecting regulatory elements in the genome and characterizing evolutionary changes in regulatory regions. We start in this chapter by considering the biological basics that are necessary to develop and understand these approaches.

1.1 Regulatory Control Structures

The genome of a species contains in its DNA almost all the information that is necessary for the development of this species. Therefore, it is not surprising that for a long time we thought that the amount of DNA in a genome correlates with complexity, i.e. the number of different types of cells, and the degree of cellular organization of an organism. And indeed, species vary

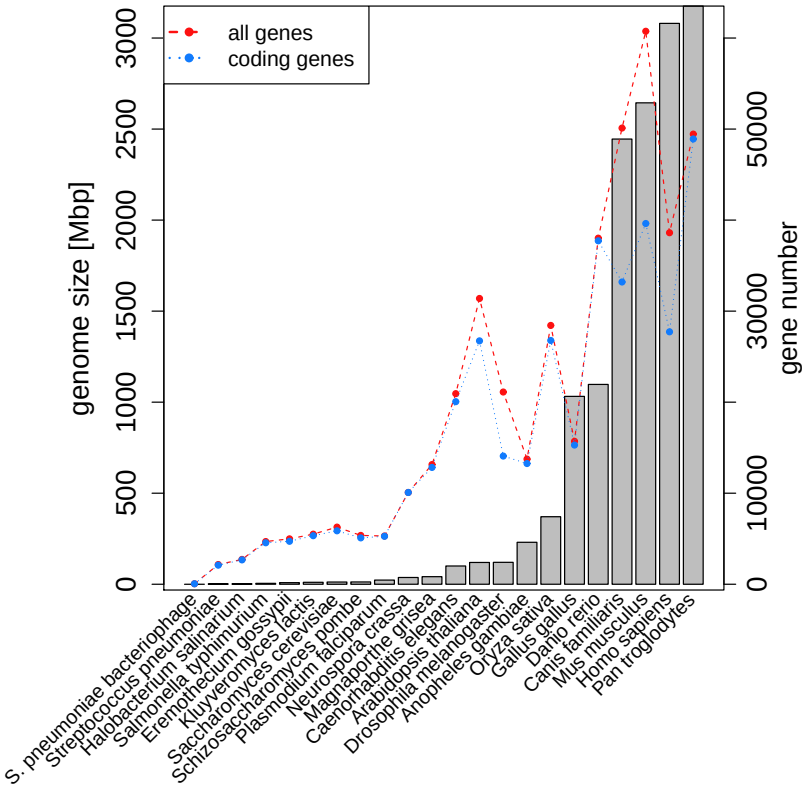


Figure 1.2: Comparison of genome size (gray bars, left axis) and gene number (red: all genes, blue: protein coding genes, right axis) based on data from Hou and Lin (2009).

enormously from one another in the amount of DNA per haploid genome. But then, data about the complexity of genomes revealed that the range of DNA sizes in many groups can vary over several orders of magnitude, even between related species. Furthermore, "simple" species like gymnosperms can have bigger genomes than "complex" species like mammals.

This lack of correlation between genome size and complexity is called the *C-value paradox* where C stands for the haploid genome size. The term was popularized by Benjamin Lewin. In Genes II (Lewin, 1983) he wrote:

> "The C value paradox takes its name from our inability to account for the content of the genome in terms of known function. One puzzling feature is the existence of huge variations in C values between species whose apparent complexity does not vary correspondingly. An extraordinary range of C values is found in amphibians where the smallest genomes are just below 10^9 base pairs (bp) while the largest are almost 10^{11} bp. It is hard to believe that this could reflect a 100-fold variation in the number of genes needed to specify different amphibians."

Consequently, the next theory was that the number of genes is responsible for the complexity.

But neither is the number of genes reflected by the genome size nor are there any correlation between complexity and gene number. For example, humans (*Homo sapiens*), chickens (*Gallus gallus*), the nematode *Caenorhabditis elegans* and thale cress (*Arabidopsis thaliana*) all have about the same number of genes, see Figure 1.2. This *G-value paradox*, where G is the number of genes in a haploid genome, (Hahn and Wray, 2002) is even more problematic since it can not be explained by varying ploidy, i.e. multiple copies of chromosomes.

Today, most scientists think that the informational paradox in complex organisms can be explained by the complexity of the regulatory control structures (Taft *et al.*, 2007). For example, the combinatorial variety of $20,000$ genes allows theoretical up to 10^{6000} different gene expression patterns.

1.1.1 Gene Expression

Life depends on the ability of cells to synthesize the information from a gene into the corresponding product. These products are either proteins, in case of protein coding genes, or functional non-coding RNAs (ncRNAs) , in case of non-coding genes, and they are essential for all metabolic processes. Their synthesis involves multiple steps, summarized in Figure 1.3. In detail, the following happens (Lewin, 2007; Alberts *et al.*, 2002):

Transcription: Within the genome, genes are informational units of DNA. Every molecule of DNA consists of two strands, each of them having asymmetric $5'$ and $3'$ ends oriented in anti-parallel direction. The coding strand contains the genetic information of a gene while the non-coding template strand serves as a blueprint for the production of RNA. The transcription of a gene is the production of RNA copies based on the corresponding DNA. It is performed by RNA polymerases. These complex molecules bind with the assistance of other molecules to the promoter, a region of DNA in front of the gen that facilitates the transcription and move along the template strand from $3'$ to the $5'$ end. Thereby it unwind and unzips the DNA by breaking the hydrogen bonds between complementary nucleotides. The unpaired nucleotides on the template strand are then paired with complementary RNA nucleotides that are connected by forming the sugar-phosphate backbone. The hydrogen bonds of the untwisted RNA-DNA compound break and the growing newly synthesized RNA strand is freed. This transcript is identical to the coding DNA strand with the exception that thymines are replaced by uracils in the RNA.

Transcription in prokaryotes is carried out by a single type of RNA polymerase while in eukaryotes three types of RNA polymerases exists. RNA polymerase I transcribes rRNA genes. RNA polymerase II is responsible for all protein-coding genes and some non-coding RNAs like snRNAs, snoRNAs and long ncRNAs. RNA polymerase III transcribes 5S rRNA and tRNA genes and some small ncRNA genes like 7SK. Each of

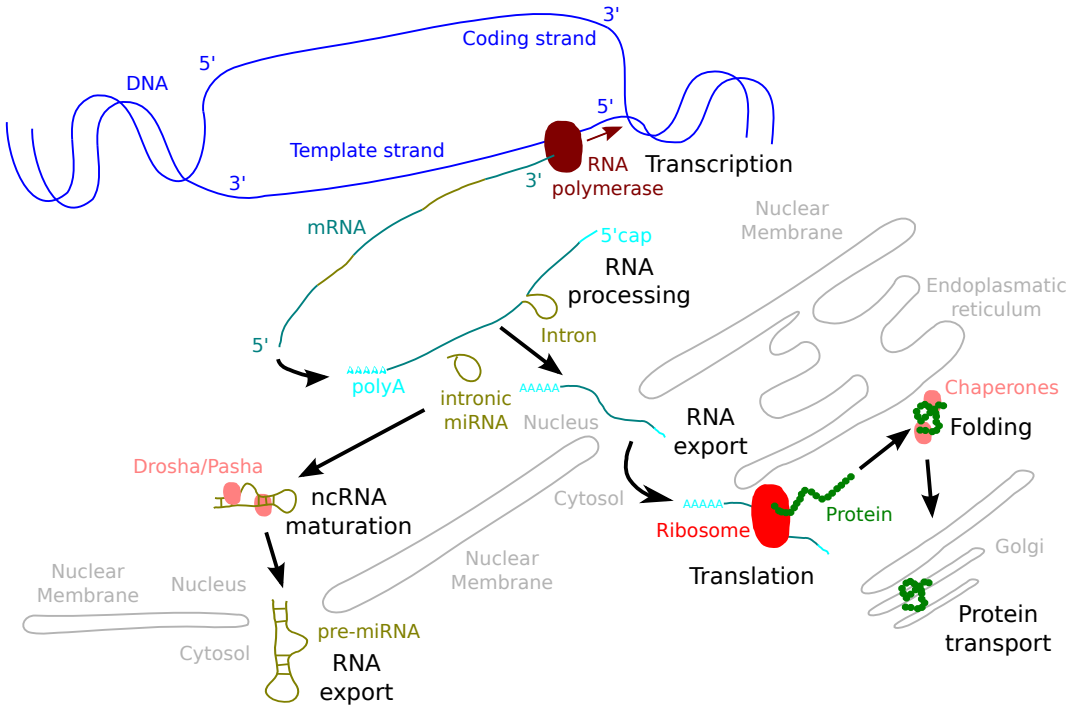


Figure 1.3: Gene expression. The gene is transcribed in mRNA by RNA polymerase. During the subsequent processing, the 5′ cap and the polyA tail are added and the introns are spliced out. In this example, one intron contains a miRNA that is processed to a short stem loop structure in the maturation step by the proteins Drosha and Pasha. The resulting pre-miRNA is then exported to the cytosol. The mRNA is also exported to the cytosol where it is translated in to protein by the ribosome. The structure-less protein chain is then folded at the endoplasmatic reticulum with help of chaperones and finally transported to its destination by the Golgi apparatus.

the eukaryote types has its own kind of promoter and factors to initiate the process. Transcription ends at a special sequence called terminator.

RNA processing: The transcript of prokaryotic protein-coding genes is already messenger RNA (mRNA) that carries the gene information that is needed for the translation. In contrast, the primary transcript of eukaryotic genes, the pre-mRNA has first to undergo a series of modification to become a mature mRNA. These modifications include 5′ capping, 3′ cleavage with polyadenylation and splicing.

The 5′ capping adds 7-methylguanosine to the 5′ end of pre-mRNA. This cap protects the RNA from degradation by exonucleases and aids the mRNA export to cytoplasm. The 3′ cleavage and polyadenylation occur if the polyadenylation signal sequence is present in the pre-mRNA. In this case, about 200 adenines are added after the cleavage. This poly-A tail protects the RNA from degradation and mediates the mRNA export as well as the re-initiation of translation. The RNA splicing removes areas that are not

needed for the final product, the so-called introns, out of the pre-mRNA. The responsible RNA-protein catalytic complex, known as spliceosome, catalyzes two transesterification reactions, which releases the introns in the form of a lariat structure and then splice the neighbors, the exons, together. In certain cases, alternative splicing removes or retains some introns or exons and creates so series of different transcripts originating from a single gene which extends the complexity of eukaryotic gene expression.

ncRNA maturation: The ncRNA of non-coding genes is mostly transcribed as precursors which undergo further processing (Eddy, 2001). For example, ribosomal RNA (rRNA) is often transcribed as a pre-rRNA, containing one or more rRNA. This pre-rRNA is cleaved and modified by snoRNP, small nucleolus-restricted RNA (snoRNA) that is associated with proteins. Within the complex, the RNA part binds at a precise position while the associated proteins catalyse the reaction. One example is the cleaving of the 45S pre-rRNA into the 28S, 5.8S, and 18S rRNA in eukaryotes. Transfer RNA (tRNA) is processed by RNase P which removes the $5'$ end, by tRNase Z which removes the $3'$ end, and by a nucleotidyltransferase which adds the $3'$ CCA tail. Micro RNA (miRNA) is processed to short stem-loop structures known as pre-miRNA by the enzymes Drosha and Pasha. This pre-miRNA is after the export to the cytoplasm processed to mature miRNA by interaction with Dicer. This endonuclease also initiates the formation of the RNA-induced silencing complex with the RNase argonaute. Even snoRNA and small nuclear RNA (snRNA) are processed before they become part of the functional RNP complexes. For ncRNA the mature RNA is the final gene product.

RNA export: Some RNAs function in the nucleus. In all other cases, the mature RNA is transported in the cytoplasm through the nuclear pores. Linker proteins bind to specific sequences on the RNA and mediate the transport by motor proteins in the cytoplasma (Köhler and Hurt, 2007).

Translation: In case of protein coding genes, the mRNA carries the information for the synthesis of the corresponding protein or, common in prokaryotes, for multiple proteins. Flanked by the untranslated regions (UTRs) at the $5'$ and $3'$ end, the open reading is the part of the mRNA containing the information for the protein synthesis. This information is encoded by the genetic code. Each triplet of nucleotides is complementary to an anticodon triplet of an tRNA and each tRNA with the same anticodon always carries the same appropriate amino acid.

The translation is performed by the ribosome, a complex of RNAs and proteins. The small ribosomal subunit binds to the start codon and recruits the large ribosomal subunit. Then tRNAs with the respective amino acids bind to the ribosome, which synthesizes the elongation of the structure-less peptide according to the order of the triplets and releases the free tRNAs. In prokaryotes, the translation is done together with the

transcription in the cytoplasm, often simultaneously. In eukaryotes, translation of membrane proteins or proteins for export from the cell is mainly done on the membrane of the endoplasmic reticulum while soluble cytoplasmic proteins are mainly translated in the cytoplasm.

Folding: In order to perform its function, the structure-less random coil polypeptide has to fold into a characteristic and well-defined three-dimensional structure (Hebert and Molinari, 2007). This is done by the interaction of amino acids and by enzymes, called chaperones, that help proteins and RNAs to attain their functional shapes. In eukaryotes the folding is mainly done in the endoplasmic reticulum.

Protein transport: Proteins that do not act in the cytosol have to be modified, sorted, and packed for the transport to the correct organelle. In eukaryotes these targeting processes are mainly done by the Golgi apparatus (Moreau *et al.*, 2007).

The synthesis of gene products is essential but the regulation of this process it not less important.

1.1.2 Gene Regulation

The regulation of genes allows cells to control the timing, the location, and the amount of gene expression. Hence, it is the basis for differentiation, morphogenesis and for the versatility and adaptability of any organism. The first discovered regulation system was the lac operon in E. Coli (Jacob and Monod, 1961). In this example, the proteins involved in lactose metabolism are expressed only in the presence of lactose and absence of glucose. This prevents the inefficient production of these enzymes when no lactose is available, or if glucose is available, which is a better energy source.

In general, the expression is regulated through changes in the number and type of interactions between molecules that collectively influence transcription of DNA or translation of RNA. Thereby it is advantageous to regulate gene expression as early as possible to prevent wasting of resources. Nevertheless, all other expression steps are also modulated. Based on the corresponding step, one distinguishes the following kinds of regulations (Lewin, 2007; Alberts *et al.*, 2002):

Transcriptional regulation: The binding of the RNA polymerases to the promoter sequence on the DNA is influenced by other molecules. These molecules, also referred to as *trans*-regulatory factors or *transcription factors* (TFs) bind to specific sequence motifs of the DNA, the so-called *cis*-regulatory elements (CREs) or *transcription factor binding sites* (TFBSs) . This binding increases or decreases the probability of RNA polymerase binding and transcription initiation. In prokaryotes, the CREs are usually close to the RNA polymerase start site and either activate or repress transcription. The flexibility

of the DNA helix, however, also allows factors bound at distant sites to affect the RNA polymerase at the promoter by the looping out of intervening DNA. This is extremely common in eukaryotic cells, where factors, bound to sequences thousands of nucleotides away from the promoter, control gene expression.

Whereas the transcription of a typical prokaryotic gene is controlled by only a few factors, the regulation of higher eukaryotic genes is much more complex, corresponding to the larger genome size and the high number of different cell types. The control region of the eve gene in *Drosophila melanogaster*, for example, encompasses 20,000 nucleotide pairs of DNA and has binding sites for over 20 gene regulatory proteins (Reinitz and Sharp, 1995). In general, eukaryotic transcription regulation is composed of two main systems. The first system is epigenetic (Bird, 2007). To turn on gene expression, the molecules responsible for the transcription have to reach the corresponding DNA. In eukaryotes this accessibility depends on chemical modifications of the DNA itself and on the structure of the chromatin. The chemical modification is done by cytosine methylation, mostly at CpG dinucleotide sequences. The resulting 5-methylcytosine performs similar to regular cytosine but tends to be less transcriptionally active. The structural modifications are done by acetylation, methylation, ubiquitylation, phosphorylation or sumoylation of the histon amino acids. Being temporary, like phosphorylation, or more permanent, like methylation, they have significant impacts on the expression of genes in the lightly packed euchromatin and the tightly packed heterochromatin areas (Reik, 2007).

The second system is the interaction of regulatory factors with the transcription machinery. Thereby a wide variety of mechanism exists. The simplest and most straightforward method is direct interaction with the DNA. Genes often have binding sites around the coding region that are recognized by the DNA-binding domains of TFs. There are many classes of TFBSs. Enhancers increase the transcription levels of genes. They are bound by activators, which directly increase the rate of transcription by assisting the formation of the RNA polymerase holoenzyme, or by coactivators, which connect enhancers with activators. Silencers are bound by repressors, proteins that block the attachment of RNA polymerase to the promoter, and thus prevent the transcription of the gene. Insulators are boundary elements that either block enhancers or, more rarely, act as a barrier against condensed chromatin proteins spreading onto active chromatin. They prevent induction and repression mechanisms of independently regulated genes from interfering with one another. Together, enhancers, silencers and insulators act as a cis-regulatory modules and regulate the correct spatial and temporal pattern of gene expression, see Figure 1.4. Thereby, similar transcription responses can be produced by different combinations of TFBSs, differing in both, number and sequence (Hare *et al.*, 2008). Hence, binding sites underlie a turnover during phylogeny, which may or may

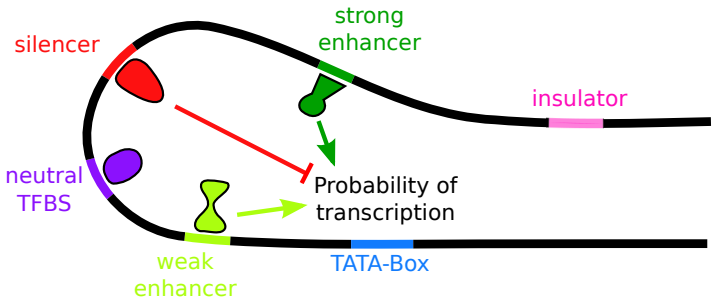


Figure 1.4: Interaction of activating and repressing transcription factors bound to transcription factor binding sites. In this cis-regulatory module multiple sets can work together to influence the probability of transcription initiation.

not affect regulatory function (Ludwig and Kreitman, 1995; Ludwig *et al.*, 2000, 2005).

Besides direct interactions, the activity of TFs is also modulated. Intracellular signals cause protein post-translational modifications like phosphorylations, acetylations or glycosylations. These changes influence the ability of TFs to bind to promoter DNA, to recruit RNA polymerase, or to favor elongation of a newly synthesized RNA molecule. The nuclear membrane in eukaryotes allows further regulation of TFs by the duration of their presence in the nucleus through reversible changes in their structure or by binding to other molecules (Veitia, 2008).

Post-transcriptional regulation: The mRNA created by transcription is the target of post-transcriptional regulation. In eukaryotes the transport out of the nucleus to the translation machinery is controlled by a wide range of import and export proteins. The degradation of mRNA can also be modulated. This is done via the stabilizing post-transcriptional modifications like the $5'$ cap and the poly-adenylated tail. The RNA interfering is another way of regulating the amount of mRNA. In this case, miRNA or small interfering RNA (siRNA) with complementary sequence bind to the mRNA and either increase or decrease their activity.

Translational regulation: Direct regulation of translation is less prevalent than control of transcription or mRNA stability. Nevertheless, the translation is, for example, a major target for antibiotics and toxins. The corresponding mechanisms are primarily based on the control of ribosome recruitment. In most cases, translational regulation involves specific RNA secondary structures on the mRNA (Kozak, 1999).

Protein degradation: Finally, the level of gene expression can be modulated by the degradation of the corresponding protein. There are major protein degradation pathways in all prokaryotes and eukaryotes. For example, unneeded or damaged protein is often labeled for degradation via the proteasome by addition of ubiquitin (Ciechanover, 2005).

A great part of the gene products have again influence on the expression of other genes. These feedback loops create a regulatory network that allow the cells to respond to internal or external signals by up-regulation or down-regulation of the expression of specific genes. This mechanism plays a key role during the development of organisms since a small change in a TF expression level can affect the regulation of a large number of genes and result in a significant phenotypic effect (Seidman and Seidman, 2002). In addition, only a few TFs are required to determine or to change the type of cells (Zhou *et al.*, 2008). Based on this essential meaning of regulatory elements, where small variations can result in profound effects in the development, changes in regulatory elements are a key process for evolution (Wray, 2007).

1.2 Evolution in the Context of Regulation

The replication of organic molecules started approximately four billion years ago. Since this time, evolutionary events have created the diversity of life forms we find on earth today, no matter if we talk about ancient fossils or present organisms. The genetic constancy with which organism transmit the genetic information to their offspring is crucial to maintain life. Nevertheless, evolution cannot occur without genetic variability.

1.2.1 Genetic Constancy and Diversity

Changed or unchanged, the genetic information is transmitted from one generation to the next by cell division. In prokaryotes this happens by binary fission while eukaryotes use somatic cell divisions (mitosis) or gamete-producing division (meiosis). While the daughter cells created by binary fission and mitosis are supposed to be genetically equivalent, meiosis provides the first mechanism for variability. By recombination among homologous chromosomes and by random assortment of these chromosomes into the gametes, an exponential number of different chromosome configurations are possible. The second mechanism is provided by mutations. They are caused by physical influences like energetic radiation including UV and X-rays, mutagenic chemicals or biological reasons like viruses, transposons and errors during meiosis or DNA replication. Mutations can also be induced by the organism itself like in the case of hypermutation where deamination of cytosine to uracil in immunoglobulin genes causes high mutation rates that provide a fast adaption of the immune system to foreign elements (Teng and Papavasiliou, 2007).

Based on the structural effect of the mutation, one can distinguish between large and small changes (Dayhoff *et al.*, 1983):

Small-scale mutations are minor and local changes which affect only one or at most a few consecutive nucleotides. These include:

Point mutations: A single nucleotide is exchanged by another. If a purine is exchanged by a purine ($A \leftrightarrow G$) or if a pyrimidine is exchanged by a pyrimidine ($C \leftrightarrow T$) the mutation is a transition. The contrary transversions ($A/G \leftrightarrow C/T$) are less common.

Insertions: One or more extra nucleotides are added into the DNA. This is usually caused by transposable elements and hence, not entirely random.

Deletions: One or more nucleotides are removed randomly.

Large-scale Mutations are changes in the chromosomal structure, mostly caused by unequal crossing-over of the chromosomes. They result in changes of the amount of genetic material and include:

Amplifications: Multiple copies of chromosomal regions are inserted.

Deletions: Large chromosomal regions are removed.

Translocations: Parts of non-homologous chromosomes are interchanged.

Inversions: The orientation of a chromosomal segment is reversed.

Transposition: Parts from the same chromosomes are rearranged.

In contrast to recombination and mutation, natural selection is a source for genetic constancy. Based on the fitness of an organism, i.e. the ability to survive and reproduce, changes in the genome can be advantageous, neutral or disadvantageous. Individuals with lower fitness are more likely to die early or fail to reproduce. Therefore, changes which on average result in greater fitness become more abundant in the next generation, while changes which generally reduce fitness become rarer. If groups of organisms of the same species are isolated, this process can lead to the emergence of new species. For genetic regions with essential meaning for the organism, changes are mostly disadvantageous. This leads to conserved sequence or structure patterns inside the genome.

1.2.2 The Meaning of Regulatory Elements

Discrete changes in cis-regulatory sequences can alter gene expression and thus generate potential for novel species-specific traits to arise (Carroll, 2008). For example, recent studies have suggested that regulatory changes could play key roles in primate evolution (Wray, 2007). This is supported by computational analyzes showing that changes in the genomic region surrounding orthologous human and chimpanzee genes are correlated with increased expression divergence (De *et al.*, 2009).

While gene regulation can be altered quickly during evolution, substantial changes to gene expression may only occur at late stages of the speciation process. The comparison of expression patterns of different tissues between related vertebrates including pufferfish, frog, chicken, mouse, and human shows that gene expression profiles are more similar among homologous

tissues in different species than among tissues of the same species (Chan *et al.*, 2009). However, the conservation of gene expression during evolution did not correlate with the amount of nearby conserved non-exonic DNA (Schmidt *et al.*, 2010). Hence, tissue-specific expression patterns can be maintained despite extensive sequence divergence in regulatory DNA (Fisher *et al.*, 2006; Tsong *et al.*, 2006; Crocker and Erives, 2008).

Binding Preferences of Transcription Factors

Tissue-specific gene expression is driven by the non-covalent protein-DNA interactions where trans-regulatory TFs bind to cis-regulatory TFBSs. During vertebrate evolution the repertoire of TFs has expanded. For example, the human organism contains over 400 cell types. Each of them is defined by a specific set of expressed genes that are regulated by over 1300 TFs annotated so far. Most of these TF are either tissue-specifically expressed (2 to 3 tissues) or generally expressed (more then 30 tissues) (Vickaryous and Hall, 2006). Unfortunately the understanding of transcriptional control is restricted since binding preferences of TFs are complex (Wilson and Odom, 2009).

In a novel approach, Badis *et al.* (2009) used microarrays containing all possible 10 base pair sequences to determine binding preferences of over 100 TF of mouse. An unexpected observation was that half of the binding domains analyzed had strong, mutually exclusive secondary binding motif preferences. *In vivo*, these secondary motifs are often bound with equally high affinity as the primary motif.

The authors divide the secondary binding motifs in four categories:

- Motifs with *variable spacer lengths* are motifs where a variable number of nucleotides separating the recognized parts of a motif. An example is the leucine zipper transcriptional regulator Jundm2 that binds `TGACGTCA` or `TGAGTCA`.
- Motifs with *position interdependence* are motifs where binding depends on the mutual presence of certain nucleotides at certain positions. An example is the estrogen related receptor alpha that binds `CAAGGTCA` or `AGGGGTCA`, but not `CAGGGTCA` or `CGGGGTCA`.
- Motifs with *multiple effects* display a combination of position interdependence and variable spacer lengths.
- Motifs with *alternate recognition interfaces* are not readily explainable by variable spacer length or position interdependence. This category is the most intriguing, since it suggests that some TFs recognize their DNA binding sites through multiple, completely different interaction modes. This could be via alternate structural features or by switching between alternate conformations.

The fact that many TFs possess similar binding affinities for divergent sequences is an important and unexpected observation with direct relevance for cis-regulatory evolution.

Abundance of Binding Sites

In a relatively short time, new binding sites can evolve. This process can be influenced by pre-sites, which are regions of DNA predisposed to evolving into new regulatory sites (MacArthur and Brookfield, 2004). Rapid expansion of regulatory sequences can also be driven by genomic duplications (Wapinski *et al.*, 2007) or by certain families of transposons that serve as an abundant source of pre-sites for specific TFs (Bourque *et al.*, 2008). Binding sites are also lost due to disruption (Moses *et al.*, 2006). Genome-wide comparisons between multiple Drosophila species suggest that the fraction of shared TFBS decreases as the divergence time increases (Kim *et al.*, 2009). This is consistent with the molecular clock hypothesis (Kumar, 2005).

The resulting divergence in potential regulatory regions between closely related species seems to be the general rule for many species (Dermitzakis and Clark, 2002; Schmidt *et al.*, 2010). One explanation is that transcriptional control often involves multiple TFs that act together as cis-regulatory modules. Therefore, the birth or death of single binding sites has only small influence on the transcription (Odom *et al.*, 2007) which relaxes the selective pressure onto single sites. This is especially the case when regulation is only determined by the quantity of TF binding to target sites (MacArthur *et al.*, 2009).

Differences in transcriptional regulation are caused by cis-regulatory sequence variations and changes in trans-regulatory TFs. An example for the importance of cis-regulation is that mouse encoded TFs bind an additional human chromosome in a mouse cell almost as precisely as human-genome-encoded TFs (Wilson *et al.*, 2008), i.e. TFs of mouse work also with human cis-regulatory sequences. Nevertheless, under changing environmental conditions, trans regulation becomes increasingly prevalent. One example are sensory changes like mutations in cell surface receptors (Tirosh *et al.*, 2009).

While regulatory cis-acting DNA in specific organisms and pathways can be highly conserved both in sequence and function, the combination of plasticity, regulatory potential, and rapid turnover provides an excellent basis for rapid evolutionary changes.

1.3 Detection of Regulatory Elements

In addition to knowing “how” and “what” DNA sequences TFs bind, the question of “where” TFs bind in the genome is important for understanding motif preference, tissue-specific gene regulation and evolution. For the detection of regulatory TFBSs, experimental as well as computer-based approaches are used. If the binding site is known, motif search algorithms are used to determine locations in the genome.

1.3.1 Experimental Binding Site Mapping

Mapping binding sites of TFs can be done using chromatin immunoprecipitation, or short ChIP (Collas, 2010). The first step of ChIP is the crosslinking of proteins to chromatin in a cell lysate. The DNA-protein complexes are then sheared and DNA fragments associated with the proteins of interest are selectively immunoprecipitated. This is done by specific antibodies to the proteins, commonly coupled to agarose, sepharose or magnetic beads. The relative amount and genomic location of enriched DNA fragments can then be analyzed using microarrays, short ChIP-chip (Reimer and Turck, 2010), or DNA sequencing, short ChIP-seq (Johnson *et al.*, 2007). ChIP-seq experiments are amenable to any species with a genome assembly and can detect TF binding at high resolution in all but the most degenerate repeat regions within a mammalian genome. Importantly, TF binding events often occur in repetitive regions, which by design are absent from most microarrays (Bourque *et al.*, 2008).

1.3.2 Computer-Based Binding Site Prediction

The experimental detection of binding sites certainly provides results of highest quality, but caused by the experimental effort it is limited to a small number of cases. For analyzes beyond single TFs, computational approaches are the first choice. The problem here is that binding site motifs can be very short, down to only a few nucleotides. Located in the regulatory regions, which can cover several thousand nucleotides (Dieterich *et al.*, 2002), they are not very significant. They can become outweighed by random similarities and it is very likely that similar patterns exist by chance. Prediction algorithms try to overcome this problem using two strategies. The first is based on sequence conservation between orthologous sequences, the so-called *phylogenetic footprinting*, while the second is based on nucleotide composition and detects overrepresented motifs (Wasserman and Sandelin, 2004; Ureta-Vidal *et al.*, 2003).

Phylogenetic Footprinting

Phylogenetic footprinting is a technique used to identify TFBSs within intergenic regions of DNA through cross-species comparison. The term is used in analogy with DNAase footprinting (Tagle *et al.*, 1988). In case of closely related species, it is called phylogenetic shadowing (Boffelli *et al.*, 2003).

It is based on the assumption that due to selective pressure during evolution, mutations within functional regions of genes will accumulate more slowly than mutations in regions without sequence-specific function (Frazer *et al.*, 2003). The resulting sequence similarity can be used to indicate segments that might direct transcription by comparison of orthologous genes. Indeed, putative transcription-factor binding sites are enriched in conserved non-coding genomic sequences (Wasserman *et al.*, 2000; Levy *et al.*, 2001; Fickett and Wasserman, 2000) and evolutionarily conserved regions can be linked to experimentally determined regulatory

elements (Aparicio *et al.*, 1995).

Although originally used for the detection of conserved regions between orthologous sequences, phylogenetic footprinting can be used for all kinds of homologous sequences or even sequences that have no common ancestor but that share similar regulatory characteristics. Nevertheless, the regulation of homologous genes is only for moderate evolutionary distances subject to the same regulatory mechanisms. Comparison of promoters from closely related species, such as inside the primates, generally provide little benefit, as the sequences closely resemble each other. In contrast, promoters of widely divergent species, e.g. within the vertebrates, can show no detectable similarity (Lenhard *et al.*, 2003). Furthermore, the rate of evolutionary events in promoters is different for individual genes within the same organism. For instance, regulatory elements in the vertebrate Hox gene clusters have a high selective pressure. This is linked to chromatin structure or unknown mechanisms (Santini *et al.*, 2003) and results in evolutionary distances as extreme as $450-500$ million years necessary for useful comparisons (Aparicio *et al.*, 1995).

In general, phylogenetic footprinting algorithms consist of three components: defining suitable homologous gene sequences, comparison of the promoter sequences of homologous genes and visualizing or identifying segments of significant conservation.

For the definition of homologues, the assumption is made that homologous genes are under common evolutionary pressures. This can be problematic in some cases since retained function is not inherent to the definition of homology. In addition, duplication as well as deletion of genes during evolution makes it sometimes difficult to select reliable sets of sequences. Databases that provide homologues between species include `HomoloGene` (Sayers *et al.*, 2011), `COG` (Tatusov *et al.*, 2003) and `HOPS` (Storm and Sonnhammer, 2003).

Once suitable sequences are obtained, they must be compared. The standard method for the detection of conserved regions is the computation of alignments. An alignment is an arrangement of biological sequences in order to identify regions of similarity that could be interpreted as the consequence of functional, structural or evolutionary relationships between sequences. In computational biology, this arrangement is based on the optimization of a scoring scheme that rewards matches, i.e. arrangements of similar sequence areas, and penalizes mismatches, i.e. arrangements of dissimilar areas. Gaps between arranged areas correspond to insertions (respective deletions) of areas and they are also penalized.

There are two widely used strategies: local alignments that target short segments of similarity and global alignments that determine a description of similarity across the entire sequences. An example for the local approach is the `LASTZ` algorithm, a drop-in replacement for `BLASTZ` (Schwartz *et al.*, 2003). It identifies short segments of exact identity, the so-called seeds, and constructs local pairwise alignments by extending the seeds in both directions. The global approach is, for example, used by `LAGAN` (Brudno *et al.*, 2003). This program generates short local alignments to identify related sub-segments. This segments are used as anchors for the

computation of the intermediate alignments with a standard Needleman-Wunsch algorithm (Needleman and Wunsch, 1970). The program `mLAGAN` uses the pairwise alignments of the `LAGAN` algorithm to determine multiple alignment in a progressive manner (Thompson *et al.*, 1994).

Global alignment tools generally have difficulties with large-scale mutations. For example, duplications will lead to results that indicate that one of the copies is not conserved. While local approaches circumvent such problems, the failure to consider collinearity, i.e. the upkeep of order and orientation of functional elements along the genome, might result in decreased ability of identifying subtle but important similarities in weakly conserved segments between well-conserved blocks or an increased detection of random similarities that are not based on phylogenetic relationships.

An alternative approach is used by the `Footprinter` algorithm (Blanchette and Tompa, 2003). Instead of the computation of alignments in order to detect conserved regions, the algorithm determines for a given phylogenetic tree the motifs of an explicit length that minimize the parsimony score. The parsimony score is the total number of substitutions over the tree needed to explain the observed motifs. In order to determine the minimal parsimony score, `Footprinter` enumerates over all possible motifs at all nodes of the tree whereas the score of terminal taxa is 0 if the motif is part of the sequence or infinite else. The final motifs are determined by backtracking based on the minimal score.

For visualization or identification of significant segments various tools are available. The program `rVista` (Loots *et al.*, 2002) discovers CREs, combining the prediction of TFBSs by searching for motifs given in the `TRANSFAC` database (Matys *et al.*, 2003), the clustering of this predictions and the analysis of interspecies sequence conservation. Elnitski *et al.* (2003) used the distribution of mutations to distinguish coding regions from regulatory sequences. For example, coding regions tend to vary at the third codon and have insertion or deletion lengths that are multiples of codon sizes. In contrast, regulatory regions tend to have more frequent mutations inside distinct blocks that are separated by segments of high similarity. For the purpose of visualization, the `VISTA` browser (Loots *et al.*, 2002) presents a graph of nucleotide identity within a sliding window along an alignment. Similarly, `PipMaker` (Elnitski *et al.*, 2002) displays `BLASTZ` results in an intuitive presentation.

Motif Overrepresentation

This technique tries to find binding sites by searching for motifs that are overrepresented in a cluster of genes. It is also based on the assumption that meaningful sequences evolve much slower than adjacent non-functional DNA. Given a set of sequences that is expect to contain this conserved motif it is likely that the motif is contained in a significantly higher number than random sequences. There are various approaches (Tompa *et al.*, 2005; Wei and Yu, 2007) where the exact operating principle depends on the algorithm. For example, the

program `MEME` (Bailey *et al.*, 2006) optimizes the expectation value of a statistic related to the information content of the motif while `AlignAce` (Roth *et al.*, 1998) is a Gibbs sampling algorithm.

An important drawback of finding motifs by overrepresentation is the relatively bad performance on a small set of input sequences. With increasing amounts of sequence, the distinction between the conserved motifs and the diverged background becomes clearer. Also, this approach works less well with large sequences. Therefore, the usefulness is limited. Nevertheless this approaches can find motifs that independently satisfy the initial parameters of the surrounding sequence identity. This is not the case with global alignments, in which the noise of the diverged non-functional background can overcome the short conserved signal.

1.3.3 Motif Modelling and Search

Once a binding site of a TF is determined, one is also interested in further potential binding sites of this factor in the genome. This motif finding problem is complicated by the high variability of regulatory DNA motifs. Therefore, it is not sufficient to find an exact substring of some length. Instead, the known binding sites correspond to a multiple alignment and the motif finding problem consists of finding high scoring local alignments of the motif alignment and the genome (Frith *et al.*, 2004). Despite the variability on the nucleotide level, regulatory sites have in most cases a constant size, based on the conserved DNA binding site domains of TFs. The resulting insignificance of gaps allows linear search, or sublinear search by using index based algorithms, for suitable motif representations.

Modelling of Sequence Specific Binding Sites

For the representation of binding site patterns multiple possibilities exist (cf. Figure 1.5). The basis for all the models is the multiple alignment of the known binding sites. The simplest description is the consensus sequences. In this case, a consensus nucleotide letter is assigned to represent the nucleotide composition in each column. Although the use of consensus sequences provides better representation than a single sequence, this model tends to result in an information loss of the original data. Since binding bias towards certain nucleotides is not reflected, consensus sequences fail to represent the quantitative characteristics of TF binding.

This problem is circumvented by position frequency matrices (PFMs) . They contain for each column i in the multiple alignment the number $f_{b,i}$ of observations of each nucleotide b (Hertz and Stormo, 1999). However, for efficient computational analysis this representation has to be transformed. In a first step, the numbers are transform into the probabilities $p_{b,i}$ to have nucleotide b at position i by

$$p_{b,i} = \frac{f_{b,i} + c_b}{n + \sum_{b'} c(b')}$$

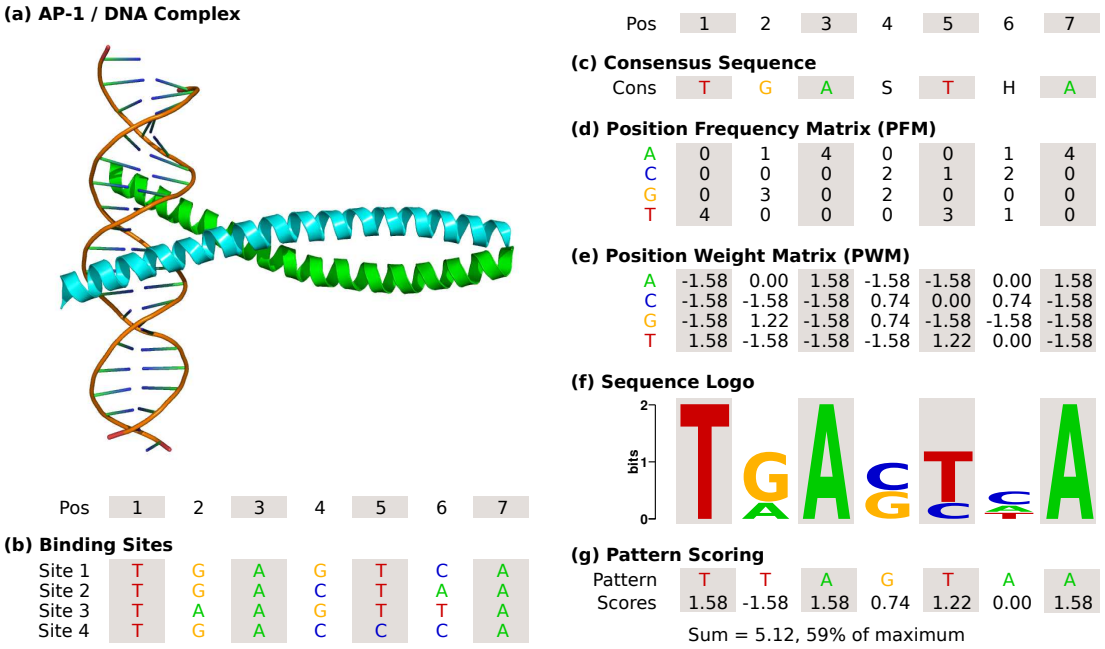


Figure 1.5: Possible descriptions for TFBSs. Here, the binding site of the TF AP-1 is used as example. AP-1 is a basic zipper that binds as heterodimer to DNA (a) and controls a number of cellular processes including differentiation, proliferation, and apoptosis (Glover and Harrison, 1995; Ameyar *et al.*, 2003). Given a data collection of aligned binding sites (b), the consensus sequence (c) is defined by the degeneracy nucleotide symbol, in this case the IUPAC nucleotide code, for each column. The position frequency matrix (d) gives the number of observed nucleotides for each position. The position weight matrix (e) contains the normalized frequency values, here with pseudocount function $c_b = p_b\sqrt{n}$, in a log-scale and is the most commonly used model. The sequence logo (f) scales the information content multiplied by the relative occurrence of the nucleotides at each position. It is used for a fast and intuitive assessment of characteristics. Given an arbitrary DNA pattern (g), the scores of the position weight matrix can be used to determine a quantitative score that reflects the similarity between the pattern and the binding site. This score is also proportional to the binding energy (Stormo, 2000).

where n is the number of binding site sequences in the multiple alignment and c_b is the pseudocount for nucleotide b. The pseudocount is a sampling correction used to eliminate null values before log-conversion and to correct for small samples of binding sites (King and Roth, 2003). The exact value for c_b varies widely.

The probabilities $p_{b,i}$ are then normalized by the background probability p_b of base b and converted to a binary log scale weight

$$w_{b,i} = \log_2 \left(\frac{p_{b,i}}{p_b} \right).$$

The resulting position weight matrix (PWM), also known as position specific scoring ma-

trix (PSSM, pronounced possum), is the most common model. Although PWMs capture a broad spectrum of the variability of regulatory regions, effects like variable spacer lengths and position interdependence (Badis *et al.*, 2009) must be described by multiple matrices. Alternatively, hidden Markov models that are able to account for the neighborhood of each nucleotide can be used.

For a fast and intuitive visualization of the regulatory motif, sequence logos are used. They display the information content $D_i = 2 + \sum_b p_{b,i} \log_2(p_{b,i})$ for each column i and mark the contribution of each nucleotide.

Furthermore, databases such as `TRANSFAC` (Matys *et al.*, 2003) and `JASPAR` (Sandelin *et al.*, 2004b) exist which contain known binding site models for TFs.

Prediction of Binding Sites in Genomic Sequences

Using position weight matrices, a quantitative score s for a potential site consisting out of basis b_i is produced by summing the relevant nucleotide PWM values over all columns i

$$s = \sum_i w_{b_i,i} \,.$$

This score is directly related to the binding energy of the DNA-protein interaction (Berg and von Hippel, 1987; Stormo, 2000) so the representation by PWM can be viewed both as a statistical and as an energy-based model. By determining all subsequences in a genomic sequence with a score above a certain percent value of the maximal score, for example with the `MATCH` program (Kel *et al.*, 2003), potential binding sites can be determined in linear time.

Although not perfect, models for representing the specificity of the factors are generally reliable and can be used to search genomic DNA to predict new potential binding sites. The largest problem is the tendency for many false positives in such searches. Based on the short length of binding sites, applications will report binding sites every 500 to 5000 bp although only a very small fraction of reports is functional in the organism. For example, in the case of myoD, a muscle-specific TF, 10^6 predictions of binding sites are accompanied by 10^3 sites that are likely to be functional (Fickett, 1996).

The high number of false predictions is not simply a result of inadequate model frameworks. They are bound readily by factors *in vitro* (Tronche *et al.*, 1997) and would also be bound *in vivo* if they were available in epigenetic terms. Hence, the pattern recognition methods detect potential binding sites, albeit not necessarily those of functional importance. Nevertheless, the difference between true and false predictions is intolerable. Algorithms that are biologically motivated and that consider the highly quantitative nature of DNA binding need to be developed.

1.4 Structure of the Thesis

In this thesis, I present new methods for the detection and evolutionary analysis of transcriptional regulatory elements. In the first part, the detection of regulatory elements, we develop a new algorithm for phylogenetic footprinting based on local alignments. As pointed out in the introduction, it is important to maintain the order and the orientation of functional elements along the genome. Otherwise, one might miss important similarities in weakly conserved segments between well-conserved blocks or to detect random similarities that are not based on homology.

In order to avoid these effects, we develop and analyze in Chapter 2 a new algorithm. For a given set of arbitrary local alignments, it determines maximal subsets that are consistent with respect to order and orientation. The chapter is based on the following publications:

- Otto W, Stadler PF, Prohaska SJ (2011). **Local, Multiple Alignments Based on Consistent Subsets of Pairwise Alignment Collections.** in prep. for *Theoretical Computer Science*

- Otto W, Stadler PF, Prohaska SJ (2011). **Phylogenetic Footprinting and Consistent Sets of Local Alignments.** In R. Giancarlo and G. Manzini, editors, *CPM 2011*, *Lecture Notes in Computer Science*, volume 6661, pages 118–131, Heidelberg, Germany. Springer-Verlag. in press.

- Otto W*, Will S*, Backofen R (2008). **Structural Local Multiple Alignment of RNA.** In A. Beyer and M. Schroeder, editors, *German Conference on Bioinformatics*, volume 136 of LNI, pages 178–177. GI. (* equal authorship)

In Chapter 3, we use the algorithm for consistent alignment subsets as basis for the development of `Tracker`, a program for the detection of phylogenetic footprints.

For the second part, the evolutionary analysis of regulatory elements, we focus on the abundance of specific TFBS motifs and its change during evolution. This is a completely new approach, based on the recent insides of transcriptional regulation, as described in this introduction, that the specific location of a binding site seems to be less important and binding sites underlie a turnover during phylogeny which may or may not affect their regulatory function.

In Chapter 4, we present and analyze a stochastic model for TFBS abundance evolution under the assumption of a constant rate of binding site origination and a constant per site decay rate. The chapter is based on following publication:

- Wagner GP, Otto W, Lynch V, Stadler PF (2007). **A Stochastic Model for the Evolution of Transcription Factor Binding Site Abundance.** *Journal of Theoretical Biology* 2007 Aug 7; volume 247, issue 3, pages 544–53.

This model is subsequently used in Chapter 5 for the development of `Creto`. For a given phylogenetic tree with known binding site numbers for the terminal taxons, this program determines the origination and decay rates that explain the tree with the maximal likelihoods. The chapter is based on following publication:

- Otto W, Stadler PF, López-Giraldéz F, Townsend JP, Lynch VJ, Wagner GP (2009). **Measuring Transcription Factor-Binding Site Turnover: A Maximum Likelihood Approach Using Phylogenies.** *Genome Biology and Evolution.* 2009 May 25; volume 1, pages 85–98.

CHAPTER 2

Alignment Consistency

> Unfortunately, Ponder was a clear logical thinker who, in times of mental confusion, fell back on reason and honesty, which, when dealing with an angry Archchancellor, were, to use the proper academic term, unhelpful. And he neglected to think strategically, always a mistake when talking to fellow academics, and as a result made the the mistake of employing, as at this point, common sense.
>
> *Unseen Academicals*
> Terry Pratchett

Regulatory sequence elements are highly important for the development of organisms. Therefore, they are likely to be subject to stabilizing selection and hence to evolve much slower than adjacent non-functional DNA. This makes these phylogenetic footprints detectable by comparative sequence analysis. In cases where large intergenic regions are under investigation and pattern discovery approaches fail to detect overrepresented sequence motifs, comparative sequence analysis is the only way to detect regulatory elements. The standard method thereby is to identify these conserved footprints using multiple alignments. Since the general alignment problem is NP-complete (Elias, 2006), the computation of multiple alignments is mainly done by heuristic approaches which reduce the problem to the computation of pairwise alignments for all sequence pairs under investigation. Based on well-defined optimization functions, these alignments are determined by summing up substitution scores and penalties for insertions or deletions (Needleman and Wunsch, 1970). Then, they are combined to a final alignment of all sequences. In case of global alignments, good heuristic approaches already exist, like e.g. progressive alignment construction (Thompson *et al.*, 1994; Notredame *et al.*, 2000), iterative methods (Morgenstern *et al.*, 1998; Edgar, 2004) or hidden

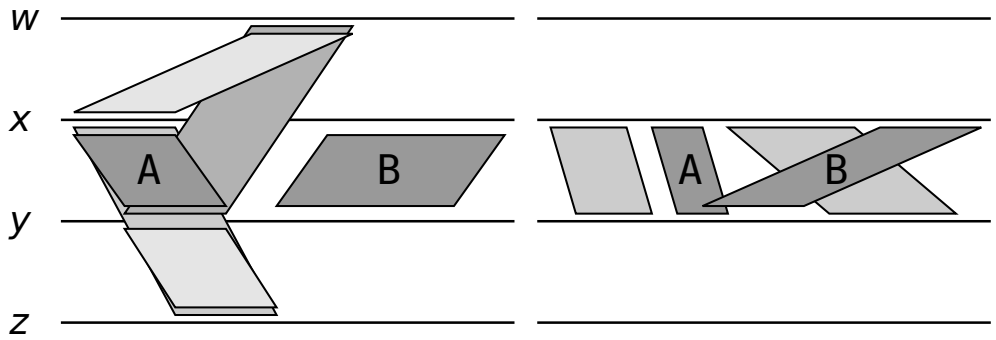


Figure 2.1: Alignment support and consistence as indicators for homologous regions. (a) Alignments between motifs that occur in many sequences are likely to be based on homology. Therefore, alignment A is more trustworthy than alignment B, even if the similarity of the sequences in B is higher. (b) The consistency with other alignments is also an indicator for homology. Therefore, alignment B that creates an inconsistency by crossing a second alignment is more likely to assign heterologous sequences than alignment A that is consistent with the remaining alignments.

Markov models (Hughey and Krogh, 1996).

However, regulatory sequence elements can be very short down to only a few nucleotides and they are surrounded by large areas of unconserved DNA. Additionally they can undergo rapid changes, caused by evolutionary events, that do not necessarily conform with the general phylogenetic relationships of the surrounding sequences (Chiu *et al.*, 2002). These are serious problems for a comparative sequence analysis based on pairwise alignments. Small regions can easily be overseen since they may not appear statistically significant or they become outweighed by random similarities in the surrounding areas.

One way to overcome this problem is to use a low stringency for the alignment computation but this creates another challenge: low significance. In addition to alignments between evolutionary conserved sequences we also get a vast number of false positive alignments between random similarities. Nevertheless, alignments between long motifs and motifs that occur in many sequences are likely to be evolutionary related. In contrast, alignments based on random similarities are arbitrarily distributed and create inconsistencies with other alignments. This can be used to detect alignments between homologous sequence parts, see Figure 2.1. In summary, given a set of pairwise alignments over different sequences, we are interested in consistent subsets of these alignments where the aligned areas occur in a great range of sequences in order to detect homologous areas. This chapter is dedicated to this problem.

First, we set up the theoretical scaffold by defining the mathematical models and formalizing the problem. This is followed by an overview of existing approaches with analyses of advantages and disadvantages. Subsequently we describe our new method and analyze its complexity. Finally, we show some results of the correctness of our method and its runtime behavior.

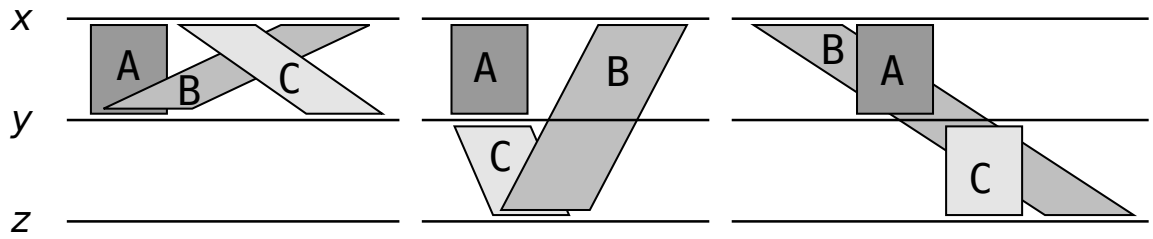


Figure 2.2: Illustration of possible inconsistencies of alignments over the same sequences. In the first subfigure, alignment A and B create a contradiction since they align the same area of sequence y with different areas of sequence x. Furthermore alignment B and C cause a crossing since the order of the aligned regions is not identical on both sequences. These inconsistencies can also be indirect over third party alignments as shown in the second and third subfigure for contradictions and crossings respectively.

2.1 Basics and Definitions

The composition of sequences define a linear order between their sites. Since alignments are assignments of sequence sites, this order applies also for parts of alignments over the same sequence. Thereby, different sequences can imply contrary orders. For example, two alignments between the same sequences x and y can have different orders at x and y. In this case, both alignments create a conflict in respect to the order implied by the sequences and we say that the alignments are inconsistent or that they create an inconsistency.

We distinguish between two kinds of inconsistencies, see Figure 2.2. In cases where alignments assign the same area at one sequence to different areas at one other sequence, we denote the inconsistency as a contradiction. Inconsistencies based on contrary orders are denoted as a crossing. Since the order relation is transitive, such inconsistencies can also be caused indirectly. This is, for example, the case in the middle and the right illustration of Figure 2.2, where sequence x implies $A < B$ while sequence y and z imply $A = C = B$ (middle illustration), or where sequence x implies $B < A$ while sequence y and z imply $A < C < B$ (right illustration). A set of alignments is therefore consistent, if there is no direct or indirect inconsistency.

This interpretation of consistency is similar to the constraints that define a valid alignment (Needleman and Wunsch, 1970), where alignment columns have at most one entry for each sequence and different alignment columns are not allowed to cross each other. If we split all alignments in single columns and if we join columns when they contain equal sequence sites, columns with multiple entries at one sequence correspond to contradictions while crossing columns correspond to alignment crossings.

Based on this correlation, we will later detect consistent alignment sets by joining single alignments to a multiple alignment that satisfies these constraints. In contrast to the standard definition of alignments, not all sequences sites will be part of this multiple alignment. If sites are not aligned by any consistent pairwise alignment in the set, these sites will be missing.

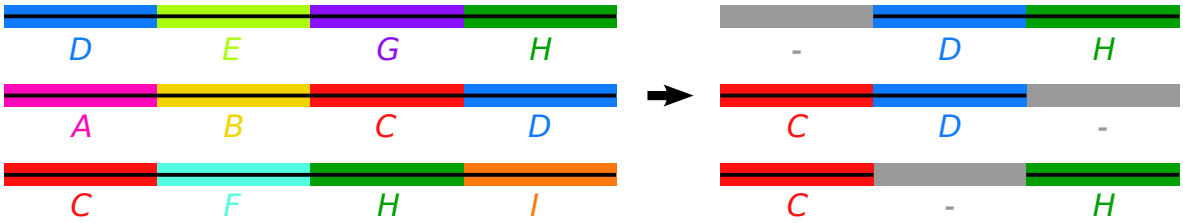


Figure 2.3: Column based local multiple alignment. The left part of the figure shows the initial situation of the sequences where corresponding motifs have the same color. The determination of pairwise alignments will result in three alignments C (red areas), D (blue areas) and H (green areas). The assembly of these alignments creates the multiple alignment showen in the right part. The sequence areas of the pairwise alignments are aligned in columns where the order is determined by the order of the motifs on the sequences. The gray parts are gaps. Note that the alignment is local since sequence areas that are not aligned by pairwise alignments are not included.

Hence, the resulting multiple alignment is a local alignment that corresponds to an ordered list of conserved but not necessarily consecutive motifs, see Figure 2.3.

In theory, the method of joining pairwise alignments to a multiple alignment corresponds to the standard alignment problem (Needleman and Wunsch, 1970) with a scoring scheme where gaps, i.e. indels, are neutral, mismatches are negative and matches positive. This scoring in theory allows the detection of conserved motifs in respect to the underlying sequence order even if the distance between motifs differs dramatically between the sequences. Practically, gap neutral scoring will end up in highly fragmented alignments that are insignificant and unlikely to be based on homology.

In contrast, conventional scoring schemes, used by algorithms like `ClustalW2` (Thompson *et al.*, 1994), penalize gaps. On the one hand, this enables them to detect significant motifs. On the other hand, it makes them inappropriate for the detection of sequence elements where the exact position, and hence the distance between them, is less important than the general presence.

Joining local alignment between significant motifs to a multiple alignment without penalizing different distances between these motifs is therefore a possibility to use the advantages of both scoring schemes while avoiding the disadvantages. Furthermore, this approach preserves the order and orientation of functional elements along the genome. This is a further advantage since the additional information supports the detection of homologous similarities. In contrast, programs like `Footprinter` (Blanchette and Tompa, 2003) miss this additional information since they enumerate over all possible motifs of a certain length without checking the location, see Section 1.3.2 (p.14).

In summary, this kind of multiple alignment assembly is well suited for the determination of consistent alignment subsets in order to detect conserved, regulatory sequences based on homology. For the formalization of the problem, we start with necessary basics.

2.1.1 Biological Sequences and Alignments

A biological sequence is a generic term for DNA, RNA, or proteins. It is a single, continuous molecule consisting of nucleic acids in the cases of DNA and RNA or amino acids in the case of proteins. In computer science it can be thought of as a multiple inheritance class hierarchy. One hierarchy is of the underlying molecule type: DNA, RNA, or protein. The other hierarchy is how the underlying biological sequence is represented by the data structure. It could be an actual sequence of amino acids or nucleic acids, a physical or genetic map, or a some more complicated data structure building a composite view from other entries.

Sequences: In this thesis we represent biological sequences as words of a formal language over a finite set Σ of characters, called the *alphabet*. Thus, Σ consists of letters that correspond to the different nucleic or amino acids. In this theoretical consideration, however, we do not restrict Σ in any way in order to allow a broad range of applications. The Kleene-closure

$$\Sigma^* = \bigcup_{i \in \mathbb{N}} \Sigma^i$$

defines the *sequence space* over Σ and each word $\omega \in \Sigma^*$ is called a *sequence*. The *length* $|\omega|$ of sequence ω is defined as the unique number $l \in \mathbb{N}$ with $\omega \in \Sigma^l$. Furthermore, $\omega[i]$ with $1 \leq i \leq |\omega|$ refers to the i-th letter of ω while $\omega[i:j]$ with $1 \leq i \leq j \leq |\omega|$ stands for the consecutive *subsequence* $\omega[i] \circ \ldots \circ \omega[j]$. Any function $S : \{1, \ldots, m\} \rightarrow \Sigma^*$ is a *sequence family* of m sequences over Σ and we refer to the x-th sequence $S(x)$ by S_x. The *site space* $\mathcal{S}$ of a sequence family $S = \{S_1, \ldots, S_m\}$ is defined as the set

$$\mathcal{S}(S) = \{[x, i] : 1 \leq x \leq m, 1 \leq i \leq |S_x|\}.$$

If the context is clear, we denote the site space $\mathcal{S}(S)$ only by $\mathcal{S}$.

Alignments: An alignment A of a sequence family S is a bijective assignment of elements of the site space $\mathcal{S}$ that is consistent with the linear order of the sites in the individual sequences. Formally this can be represented as a relation $A \subseteq \mathcal{S} \times \mathcal{S}$ (Morgenstern *et al.*, 1999), as a graph $A = (V, E)$ with $V \subseteq \mathcal{S}$ and $E \subseteq \mathcal{S} \times \mathcal{S}$ (Notredame *et al.*, 2000; Morgenstern *et al.*, 1998), or as higher forms of representations like matrices (Needleman and Wunsch, 1970). Here, however, we use the basic formulation in form of graphs.

Definition 2.1 (Alignment)**.** Given a sequence family S of m sequences $S_1, \ldots S_m$ with the corresponding site space $\mathcal{S}$, an *alignment* $A(S)$ of S is an undirected graph $A(S) = (V, E)$ with nodes $V = V(A) \subseteq \mathcal{S}$, vertex labels $\sigma : V \rightarrow \Sigma$ with $\sigma([x, i]) = S_x[i]$, and edge set $E = E(A) \subseteq \{[x, i], [y, j] : [x, i], [y, j] \in V\}$ satisfying the following three conditions:

1. The connected components of A are complete graphs K_p with $p \leq m$ (i.e. graphs with

p vertices and $p(p-1)/2$ edges). These complete graphs correspond to the alignment columns. Vertices without adjacent edges are unaligned positions.

2. For each connected component c of A we have $[x,i] \in V(c) \wedge [x,j] \in V(c) \Rightarrow i = j$. Thus, every alignment column contains at most one position from each sequence.

3. All connected components conform to the partial order $\preceq$ which reflects the linear order of the sites in the individual sequences and which is defined for two connected components c and c' by $c \preceq c' \Leftrightarrow \exists [x,i] \in V(c) \wedge \exists [x,j] \in V(c') \wedge i \leq j$. This ensures that the columns of the alignment never cross each other.

Note that alignments can be stored and manipulated more efficiently, for example as a partially ordered sets of alignment columns. This point of view is used later. The graph structure introduced here, however, appears more convenient for theoretical analysis, in particular when starting from collections of pairwise alignments whose union in general does not form an alignment. Additionally, this definition is very flexible. On the one hand the alignment can be local, i.e. not all elements of the site space $\mathcal{S}$ have to be part of A. On the other hand, the alignment can contain gaps, i.e. the elements of the site space of one sequence in A have not to be consecutive. Furthermore, for $m = 2$ we call A a pairwise alignment while for $m > 2$ we say A is a multiple alignment.

As mentioned above, alignments should arrange regions together which are evolutionary related. In order to describe and compare the level of the relatedness of the sequences in different alignments we assume a *scoring function* β that assigns each alignment A a *weight* $\beta(A)$. Without loosing generality we assume that the weight correlates with the relatedness of the sequences in the alignments, i.e. the higher the weight of an alignment the higher the relatedness. Further assumptions like additivity are not necessary for this consideration. In fact, the weight of each input alignment A can be assigned arbitrarily in our setting.

Each alignment A of a sequence family S includes subgraphs for each subset $\mathcal{R} \subseteq \mathcal{S}$ of the site space of S. The restriction $A(S)[\mathcal{R}]$, or short $A[\mathcal{R}]$ of A to this subset $\mathcal{R}$ is the subgraph of A induced by $\mathcal{R}$, i.e. $A[R] = (V, E)$ with $V = \mathcal{R}$ and $([x,i],[y,j]) \in E \Leftrightarrow [x,i],[y,j] \in V \wedge ([x,i],[y,j]) \in E(A)$. Note that $\mathcal{R}$ not only represents the site space of single subsequences of a subset of the sequence family S but also of an arbitrary number of subsequences for each sequence in S and that this subsequences do not have to be consecutive. We can show that these subgraphs induced by $\mathcal{R}$ in return are alignments:

Lemma 2.1. *Let A be an alignment of a sequence family S and $\mathcal{R} \subseteq \mathcal{S}$ a subset of the corresponding site space. Then the subgraph $A[\mathcal{R}]$ is an alignment.*

Proof. It suffices to show that all three conditions of Definition 2.1 (p.27) hold for $A[\mathcal{R}]$. This is trivial for the second condition since the vertex set of $A[\mathcal{R}]$ is a subset of $V(A)$. The restriction of A to $\mathcal{R}$ is the union of the disjoint sets in A restricted to $\mathcal{R}$. These disjoint

sets are the alignment columns and hence complete graphs. Since every induced subgraph of a complete graph is again complete, the first condition holds for the subgraph as well. Furthermore, the induced columns remain disjoint sets and are therefore still conform to the partial order $\preceq$ of the last condition. □

2.1.2 Consistency of Alignment Collections

A collection of alignments can be inconsistent as seen in Figure 2.2. Nevertheless, a single alignment is consistent by definition. We therefore use Lemma 2.1 (p.28) and define the consistency of an alignment with the help of a superior single alignment.

Definition 2.2 (Consistency)**.** Given a sequence family S and a collection $\mathcal{A} = \{A_1, \dots A_n\}$ of n alignments over subsets of the site space of S, i.e. $V(A_i) = \mathcal{R}_i$ with $\mathcal{R}_i \subseteq \mathcal{S}$ for $1 \leq i \leq n$. The collection $\mathcal{A}$ is *consistent* if and only if there is an alignment M of S so that for all $1 \leq i \leq n$ holds $M[V(A_i)] = A_i$, i.e. the given alignments A_i are the subalignments of M restricted to the subsequences aligned by A_i.

E contrario, a given alignment collection $\mathcal{A}$ is consistent if and only if it is possible to join these alignments to a single alignment M.

Lemma 2.2. *Let S be a sequence family with site space $\mathcal{S}$ and let $\mathcal{A}$ be an alignment collection over subsets of $\mathcal{S}$. $\mathcal{A}$ is consistent if and only if the transitive closure $(\mathcal{S}, E')^+ = (\mathcal{S}, E'^+)$ of the graph $(\mathcal{S}, E')$ with $E' = \bigcup_{A_i \in \mathcal{A}} E(A_i)$ is an alignment.*

Proof. In the following we proof both directions:

$\mathcal{A}$ is consistent $\Rightarrow$ $(\mathcal{S}, E')^+$ is an alignment: Since $\mathcal{A}$ is consistent, it exists an alignment M and by construction the graph $(\mathcal{S}, E')$ is a subgraph of M. In particular each connected component is a subgraph of a connected component in M. Furthermore, the transitive closure of a graph is the union of the transitive closures of its connected components. Thus, the transitive closure of $(\mathcal{S}, E')$ is a transitive subgraph and therefore itself an alignment by Lemma 2.1 (p.28).

$(\mathcal{S}, E')^+$ is an alignment $\Rightarrow$ $\mathcal{A}$ is consistent: By construction of $(\mathcal{S}, E')^+$ the subgraph induced by the site space $V(A_i)$ is the alignment A_i itself. Hence, $(\mathcal{S}, E')^+$ corresponds to alignment M in Definition 2.2 (p.29) and therefore $\mathcal{A}$ is consistent.

□

Based on this Lemma, we can finally formalize the combinatorial optimization problem.

Definition 2.3 (Maximal Consistent Alignment Subset Problem)**.** Given a sequence family S and a collection $\mathcal{A} = \{A_1, \dots A_n\}$ of n alignments over subsets of $\mathcal{S}$, i.e. $A_i = (V, E)$ with $V = \mathcal{R}_i$ and $\mathcal{R}_i \subseteq \mathcal{S}$ for $1 \leq i \leq n$. The *Maximal Consistent Alignment Subset Problem* (MCASP) is to find a maximal subset $\mathcal{A}'$ of $\mathcal{A}$ that is consistent.

Note that the term *maximal* is not qualified and can be arbitrarily defined. For example, one can maximize the cardinality $|\mathcal{A}'|$ of the consistent subsets, or the sum of the alignment scores $\sum_{A_i \in \mathcal{A}'} \beta(A_i)$, or one can maximize the score of the multiple alignment M formed by the alignments in $\mathcal{A}'$.

Also, we do not restrict $\mathcal{A}$ to contain only pairwise alignments. However, the heuristic algorithm introduced later works on pairwise alignments since comparative sequence analysis are based on pairwise alignments. Nevertheless the heuristic could also be used on multiple alignments by decomposing them into all pairwise subalignments based on Lemma 2.1 (p.28).

2.1.3 Alignment Operations and Intervals

Evolutionary relationships are transitive. This also applies to alignments for which the alignment columns are complete graphs that correspond to the transitive closure. In case of the union of pairwise alignments this means that if two alignments assign two different regions to the same third region, the first two regions are also aligned indirectly. We already used this fact when constructing the transitive closure in Lemma 2.2 (p.29). For two alignments, however, we define an extra concatenation operation. Before doing so, we first adapt some basic operations for sets:

Union: The *union* of alignments is defined as the union of the vertex sets and the union of the edge sets, i.e. $A \cup B = (V, E)$ with $V = V(A) \cup V(B)$ and $E = E(A) \cup E(B)$.

Intersection: Analog to the union the *intersection* of alignments is defined as the intersection of the vertex sets and the intersection of the edge sets, i.e. $A \cap B = (V, E)$ with $V = V(A) \cap V(B)$ and $E = E(A) \cap E(B)$.

Difference: The *difference* of alignments is also well-defined in terms of their graphs. Given two consistent alignments A and B over subsets of the same site space, the difference $A \setminus B$ is the alignment (V, E) with $E = E(A) \setminus E(B)$ and $V = \{[x, i] : \exists [y, j]\{[x, i], [y, j]\} \in E\}$.

Concatenation: The *concatenation* of two pairwise alignments is the alignment that is implied by the composition of their alignment edges. Given two pairwise alignments A and B over subsets of the same site space $\mathcal{S}$, the concatenation $A \bullet B$ is defined as follows:

1. If A and B are inconsistent or disjoint, i.e., there is no vertex $[x, i]$ with $[x, i] \in V(A) \wedge [x, i] \in V(B)$, we set $A \bullet B = \emptyset$.

2. If A and B have exactly one sequence in common, $A \bullet B$ is defined as the relational composition of the edge sets $E(A) \circ E(B)$. In other words, $\{[x, i], [z, k]\}$ is an edge in the concatenated alignment $A \bullet B$ if and only if there is a vertex $[y, j] \in \mathcal{S}$ such

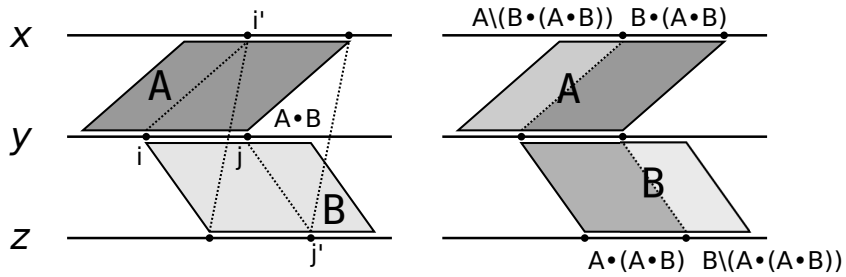


Figure 2.4: Concatenations and differences of pairwise alignments. The first part shows the alignment $A \bullet B$ implied by A and B. The second part describes the subparts of A and B induced by $A \bullet B$ in terms of the alignment operations concatenation and difference.

that $\{[x,i],[y,j]\}$ is an alignment edge in A and $\{[y,j],[z,k]\}$ is an alignment edge in B, and $x \neq z$. The vertex set consists of all vertices belonging to these edges, i.e. $V(A \bullet B) = \{[x,i] : \exists [y,j]\{[x,i],[y,j]\} \in E(A \bullet B)\}$.

3. If A and B are two pairwise alignments of the same two sequences, $A \bullet B$ is defined as the intersection $A \cap B$.

By construction $A \bullet B$ is consistent with A and B. Additionally, the concatenation operation is commutative but not associative. For a graphical representation see the first part of Figure 2.4.

For a collection of pairwise alignments $\mathcal{A}$, we define the transitive closure with respect to the $\bullet$ operation, formal $\mathcal{A}^\bullet$, as the set of all alignments that can be generated by the repeated application of the concatenation operator $\bullet$ to the alignments in $\mathcal{A}$. Note that this set is finite. By definition of the operator the number of alignment columns in $A \bullet B$ is less or equal to the number of columns in A and in B. Furthermore, $A \bullet B$ is in the same site space as A and B. Thus, the repeated application of $\bullet$ does at some point not generate any new alignments. Furthermore, the union $\bigcup \mathcal{A}^\bullet$ of the alignments in $\mathcal{A}^\bullet$ is equivalent to the transitive closure $(\mathcal{S}, E')^+$ in Lemma 2.2 (p.29). The set of pairwise alignments $\mathcal{A}$ is therefore consistent if and only if $\mathcal{A}^\bullet$ is consistent, i.e. if $M = \bigcup \mathcal{A}^\bullet$ is an alignment.

Based on the concatenation of two pairwise alignments we can furthermore define the concatenation $A \bullet B$ of a multiple alignment A with a pairwise alignment B over the same site space $\mathcal{S}$. If A and B are inconsistent, $A \bullet B = \emptyset$. Otherwise we reduce the problem to the concatenation of pairwise alignments by defining $A \bullet B$ as the union $\bigcup_{i,j}(A[\mathcal{R}_{i,j}], E_{i,j}) \bullet B$, where $\mathcal{R}_{i,j} \subseteq V(A)$ is the site space based on sequences S_i and S_j and where $E_{i,j} \subseteq E(A)$ is the corresponding edge set. The concatenation of two multiple alignments can now also be reduced to the concatenation of the first multiple alignment with the pairwise alignments corresponding to the second multiple alignment.

By this operations, it is possible to refer to specific parts of the transitive closure, see the second part in Figure 2.4.

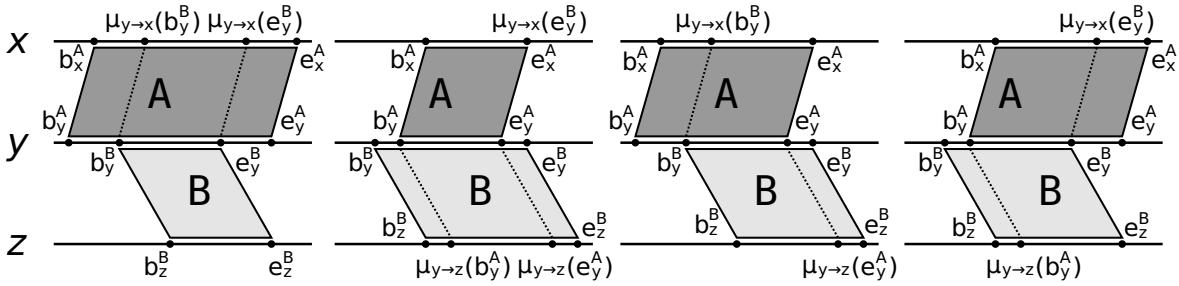


Figure 2.5: Alignment concatenation based on mapping. This picture shows all four possible cases of the relative location of two alignments that overlap at one sequence with the corresponding mappings of the overlap positions.

Alignment Intervals: In contrast to exact alignments, it is also possible to treat local pairwise alignments as matches between two sequence intervals, disregarding the exact position of the individual alignment edges within these intervals (Prohaska *et al.*, 2004b). This approximation can be implemented by representing only the delimiting edges, i.e. the left- and rightmost edge of the alignment with respect to the sequence positions. More formally, a pairwise alignment $A = (V, E)$ is defined by the two edges $\{[x, b_x], [y, b_y]\}$ and $\{[x, e_x], [y, e_y]\}$, where $b_x = \min\{i_x : \{[x, i_x], [y, i_y]\} \in E\}$, $e_x = \max\{i_x : \{[x, i_x], [y, i_y]\} \in E\}$, and b_y and e_y are defined analog. It will be convenient in to following to specify an interval on sequence x as a triple $[x, b, e]$, where b and e are the begin and end coordinates. A pairwise alignment is then described by the unordered pair of intervals $A = \{[x, b_x, e_x], [y, b_y, e_y]\}$.

Mapping: For the construction of the concatenation $A \bullet B$ of two alignments A and B two additional interval boundaries that delimit the overlap of A and B have to be known (see for example $([S_1, i'], [S_2, i])$ and $([S_2, j], [S_3, j'])$ in Figure 2.4). In principle, these boundaries could be derived from edges in the original alignment. Here, however, we use a linear interpolation scheme. Given an alignment $A = \{[x, b_x, e_x], [y, b_y, e_y]\}$, the mapping $\mu_{s\to t}(p_s)$ of the position p_s on the source sequence $s \in \{x, y\}$ at the target sequence $t \in \{x, y\} \setminus \{s\}$ is defined as

$$\mu_{s\to t}(p_s) = p_t = \frac{e_t - b_t}{e_s - b_s}(p_s - b_s) + b_t \tag{2.1}$$

See also Figure 2.5 for a graphical representation of the possible mappings.

For alignments with a small number of gaps this method is almost exact. Nevertheless, in this approximate model it is reasonable to relax the requirements by allowing a small tolerance ε for alignments. Therefore, we adapt the conditions of Definition 2.1 (p.27) as follows:

- The second condition is relaxed by allowing more than one vertex of the same sequence as long as the distance between them does not exceed a given error tolerance ε. More formal: For each connected component c of an alignment A we have $[x, i] \in V(c) \wedge [x, j] \in V(c) \Rightarrow |i - j| \leq \varepsilon$.

- The connected components of the first condition do not have to be complete graphs anymore. Instead it suffices if the sequence graph of a connected component c, which is defined as the graph (V, E) with $x \in V \Leftrightarrow \exists[x, i] \in V(c)$ and $(x, y) \in E \Leftrightarrow \exists\{[x, i], [y, j]\} \in E(c)$, is a complete graph.
- The partial order of the last condition is now defined for two connected components c and c' by $c \preceq c' \Leftrightarrow \exists[x, i] \in V(c) \wedge \exists[x, j] \in V(c') \wedge i \leq j + \varepsilon$.

Consistency, as defined originally, is recovered for the case $\varepsilon = 0$.

Column Intervals: Analog to the representation of pairwise alignments by intervals we also simplify the representation of multiple alignments. Since the connected components, i.e. the columns, in the multiple alignment are not necessary consecutive for all sequences, we can not describe the whole alignment by intervals without loosing this important information. Instead we combine the columns that are consecutive and involve the same sequences to *thick columns* and represent a thick column c as a set of intervals of the form $[z, b_z^c, e_z^c]$, one for each involved sequence z. Since these thick columns are alignments, we can use the previously defined operations. Also we use linear mapping for the determination of internal assignments as described in Equation 2.1 (p.32).

2.1.4 Complexity Classifications

Clearly, the empty set is consistent and subsets of consistent alignment sets are also consistent, i.e. consistency is hereditary. Hence, consistent subsets $\mathcal{A}'$ of $\mathcal{A}$ form an independence system (Euler, 1983) what suggests to explore greedy-like heuristics. However, the union of two consistent subsets can be inconsistent. Thus, a consistent subset is not a matroid or greedoid and distinct maximal consistent subsets may have different cardinalities. A canonical greedy algorithm will therefore in general fail to find maximal consistent subsets (Helman *et al.*, 1993).

In fact, one can expect that the optimal solution of the general Maximal Consistent Alignment Subset Problem takes exponential time.

Lemma 2.3. *The general Maximal Consistent Alignment Subset Problem is NP-complete.*

Proof. For the proof we assume a given collection $\mathcal{A} = \{A_1, \dots A_n\}$ of n alignments of subsequences of a sequence family S with m sequences $S_1, \dots S_m$ whose length is bounded by l. To show that the MCASP is NP-complete, we first have to show that the MCASP is NP-hard and secondly, that we can solve the MCASP in polynomial time with a non-deterministic algorithm.

The problem is NP-hard: In order to show that the MCASP is NP-hard, we reduce the general multiple alignment problem which is NP-complete (Elias, 2006) in deterministic

polynomial time to the MCASP. This is done by representing each possible alignment edge by a pairwise alignment. Then we solve the MCASP to determine the consistent subset of all possible alignment edges that lead to the highest sum-of-pairs score of the corresponding multiple alignment. Since the site space of S is in $\mathcal{O}(lm)$, there are only $\mathcal{O}(l^2m^2)$ possible alignment edges and hence the reduction can be done in polynomial time.

The problem is in NP: We solve the MCASP by checking for each subset $\mathcal{A}'$ of $\mathcal{A}$ if the transitive closure $M = (\mathcal{S}, E')^+$ of the graph $(\mathcal{S}, E')$ with $E' = \bigcup_{A_i \in \mathcal{A}'} E(A_i)$ is an alignment. This can be done in polynomial time:

The construction of the graph $M = (\mathcal{S}, \bigcup_{A_i \in \mathcal{A}'} E(A_i))$ can be done in $\mathcal{O}(l^2m^2)$ which corresponds to the number of possible alignment edges. This graph has at most $\mathcal{O}(lm)$ connected components which can be found in $\mathcal{O}(l^2m^2)$ (Tarjan, 1972). Each component has maximally m vertices so that the determination of the transitive closure of one component can be done in $\mathcal{O}(m^3)$ (Floyd, 1962; Warshall, 1962). This results in a total construction time of $\mathcal{O}(l^2m^2 + lm^4)$.

The decision whether a graph $M = (V, E)$ is an alignment or not can be done in $\mathcal{O}(l^2m^2)$ time. We first insert directed edges $([x, i], [y, j])$ for all nodes $[x, i]$ and $x[x, j]$ with $i < j$, corresponding to the order of letters in the sequences in $\mathcal{O}(l^2m^2)$ time. Then we determine the strongly connected components of M (Tarjan, 1972) in linear time to the number of edges in M which is in $\mathcal{O}(lm^2)$. If these components are complete graphs and have at most one position from each sequence, which can be checked in $\mathcal{O}(lm^2)$, the partial order $\preceq$ is also well defined for all pairs of components.

□

The time statements used in the proof are just of theoretical nature. By checking the alignment prerequisites while constructing M, as described later, it is possible to achieve better running times.

The complexity of the operations on pairwise alignments depends on the size of the site space which is for $m = 2$ in $\mathcal{O}(l)$. Given a vertex out of the site space of the first alignment, we can determine in constant time if this vertex also exists in the second alignment and whether there is an adjacent edge. Therefore, the concatenation, the intersection, and the difference of two pairwise alignments can be determined in $\mathcal{O}(l)$. Using intervals, these operations can be done in constant time on pairwise alignments.

The arbitrary complexity of the operation with $m > 2$ is $\mathcal{O}(lm^3)$ since each of the $\mathcal{O}(lm)$ vertices can be adjacent to $\mathcal{O}(m)$ edges which gives $\mathcal{O}(m^2)$ possible edge pairs ending in one vertex. Using intervals, the operations can be done in $\mathcal{O}(m^2)$ time on arbitrary alignments.

2.2 Related Approaches

For an overview of related approaches, we have to divide the MCASP into two settings. In the first case we examining solutions for the general problem, i.e. the determination of consistent subsets $\mathcal{A}'$ out of a given set of alignments $\mathcal{A}$. Since there are no other known solutions for the arbitrary problem with whole alignments, we are concentrating here on the optimal solution and the first `Tracker` approach (Prohaska *et al.*, 2004b). The second setting is a special case of the MCASP where the alignments in $\mathcal{A}$ are individual alignment edges. This version is the problem faced by consistency-based alignment procedures and hence is subject of most of the research on alignment consistency.

2.2.1 Consistent Alignment Subsets

In Lemma 2.3 (p.33) we have shown that the MCASP is NP-complete. Therefore, any attempt of developing a fast algorithm to compute an optimal solution of MCASP in polynomial time is expected to fail. However, using branch and bound methods, it is possible to determine exact solutions for a limited number of alignments in $\mathcal{A}$.

Optimal Solution

Given a set $\mathcal{A} = \{A_1, \ldots, A_n\}$ of n alignments over the sequence space $\mathcal{S}$ of the sequence family $S = \{S_1, \ldots, S_m\}$ of m sequence whose length is bounded by l. In order to determine if a subset $\mathcal{A}'$ of $\mathcal{A}$ with $|\mathcal{A}'| = n'$ is consistent, we can use our heuristic algorithm, introduced in the next section which needs $\mathcal{O}(n'lm)$ time.

Furthermore, we have 2^n possible subsets of $\mathcal{A}$ that have to be checked. These subsets have between 1 and n elements where the number of subsets with exactly i elements is $\binom{n}{i}$. Based on the binomial theorem

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^{n-i} y^i$$

we get for $x, y = 1$ the final time estimation

$$\mathcal{O}\left(\sum_{1 \le i \le n} \binom{n}{i} lm\right) = \mathcal{O}\left(2^n lm\right).$$

If a set of alignments $\mathcal{A}'$ is consistent, each subset $\mathcal{A}'' \subseteq \mathcal{A}'$ is consistent as well and we can omit the consistency test for all these subsets. The final bunch and bound algorithm is therefore recursive and works like this: The procedure starts with all alignments $\mathcal{A}' = \{A_1, \ldots, A_n\}$ and with the alignment switch position $p = 1$. Solutions are saved in $\mathcal{A}_{opt}$ and the score of the solution is saved as β_{opt}.

In each procedure cycle we first use our heuristic approach to check if $\mathcal{A}'$ with $|\mathcal{A}'| =$

n' is consistent in $\mathcal{O}(n'lm)$ time. In this case, we further determine the score $\beta(\mathcal{A}') = \sum_{A_i \in \mathcal{A}'} \beta(A_i)$. If $\beta(\mathcal{A}') > \beta_{opt}$, $\mathcal{A}'$ is a better solution and we set $\mathcal{A}_{opt} = \{\mathcal{A}'\}$ and $\beta_{opt} = \beta(\mathcal{A}')$. If $\beta(\mathcal{A}') = \beta_{opt}$, $\mathcal{A}'$ is an additional solution and we add $\mathcal{A}'$ to $\mathcal{A}_{opt}$. If $\mathcal{A}'$ is not consistent, we continue the cycle by checking the next subsets $\mathcal{A}'' \subseteq \mathcal{A}'$. For each i with $p \leq i \leq n$ we iteratively set $\mathcal{A}'' = \mathcal{A}' \setminus \{A_i\}$ and repeat recursively the procedure with $\mathcal{A}' = \mathcal{A}''$ and $p = i + 1$.

This procedure has at most 2^n recursive calls. Each alignment set $\mathcal{A}''$ that is produced by the recursive calls in a certain step with alignment set $\mathcal{A}'$ is a subset of $\mathcal{A}'$. Therefore we stop the recursive calls if $\mathcal{A}'$ is consistent.

An alternative method would be to save all consistent alignment sets and to perform an additional check in each procedure cycle if the new $\mathcal{A}''$ is a subset of an existing solution. In this case, we would not have to perform the next recursion step. Unfortunately, there are $\binom{n}{\lceil n/2 \rceil}$ possible solutions that are not subsets of each other. If we save the solutions in a binary tree, we could perform the subset check in $\mathcal{O}(n)$ time but the drawback would be the exponential need for time and memory to save the solutions. If all alignments have the same score, our method would in the worst case face the same problem but this is not very likely for biological data.

Nevertheless, we use this method if we want to determine all alternative solutions, i.e. all consistent subsets that are not itself subsets of other consistent subsets. Of course the number of such subsets is in the worst case in $\mathcal{O}(\binom{n}{\lceil n/2 \rceil})$ and hence exponential.

The First `Tracker` Approach

The `Tracker` program (Prohaska *et al.*, 2004b) has been developed to determine phylogenetic footprints by comparative sequence analysis. For a given set of sequences the approach determines a set $\mathcal{A} = \{A_1, \ldots, A_n\}$ of local, pairwise alignments, see Figure 2.6 (a). In order to find conserved motifs that are consistent to each other, `Tracker`determines a heuristic solution for the MCASP in two steps. First, for each pair $(A_i, A_j) : A_i, A_j \in \mathcal{A}$, `Tracker` checks if A_i and A_j are consistent.

This is done by determining all pairs of alignments that overlap with each other, see Figure 2.6 (b). These unordered pairs $\{A_i, A_j\}$ of overlapping alignments form then the vertex set of the inconsistency search graph. In this undirected graph, two vertices $\{A_i, A_j\}$ and $\{A_j, A_k\}$ are connected by an edge if and only if A_i and A_k are transitive connected by A_j. This is the case if A_i aligns one sequence of A_j and if A_k aligns the other sequence of A_j, see Figure 2.6 (c).

For the detection of inconsistencies, a depth-first search on the inconsistency search graph is performed. Starting for each node of overlapping alignments, the algorithm calculates a path alignment $\bar{A}$ that is the concatenation of all alignments in the path from the start to the present one. Thereby the concatenation and the presentation of the alignments as intervals

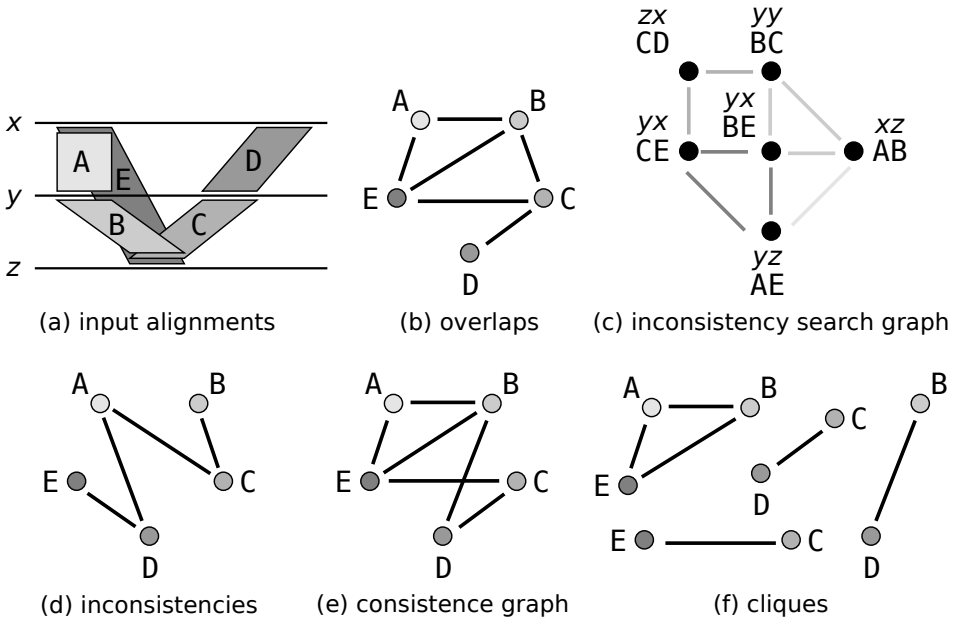


Figure 2.6: Example for the functionality of the first `Tracker` approach. For a given alignment set (a), the program first determines the overlaps (b). Based on this data, `Tracker` creates the inconsistency search graph (c). The nodes of this graph consists of overlapping alignment pairs, labeled with the corresponding alignments and the sequences that are connected by this pair. Two nodes are connected by an edge iff they have a common alignment and if the other two alignments are connected by this alignment. The program then determines the inconsistencies by a deep first search for each node. Two alignments are inconsistent if a path between them exists that is connected through other alignments and that begins and ends at the same sequence but at different areas. Based on the inconsistencies (d), the consistency graph (e) is created. The cliques of this graph are the maximal consistent subsets (f).

is similar to our definitions. At the start, $\bar{A}$ is set to the concatenations of both alignments. In each elongation step from node $\{A_i, A_j\}$ to note $\{A_j, A_k\}$, the algorithm first checks if the path alignment $\bar{A}$ aligns the same two sequences as the new alignment A_k. If this is not the case, it continues with the concatenation of them, i.e. $\bar{A} = \bar{A} \bullet A_k$. Otherwise the concatenation would result in a path that ends at the same sequence where it has started. Therefore, the concatenation is not performed. Instead the algorithm checks if $\bar{A}$ and A_k are consistent. If this is the case, i.e. $\bar{A} \bullet A_k \neq \emptyset$, the present search path is stopped since each inconsistency with the start alignment will be detected by the search path starting at A_k. If $\bar{A}$ and A_k are inconsistent, i.e. $\bar{A} \bullet A_k = \emptyset$, the search branch is also abandoned while the start alignment and A_k are marked as mutual incompatible, see Figure 2.6 (d).

The complexity of the deep first search depends on the number and edges in the search graph. Given n alignments, the number of possible pairwise overlaps, and hence the number of vertices in the search graph is in $\mathcal{O}(n^2)$. Based on the $\mathcal{O}(n^4)$ possible edges between these vertices, the determination of inconsistencies for all nodes is in $\mathcal{O}(n^6)$.

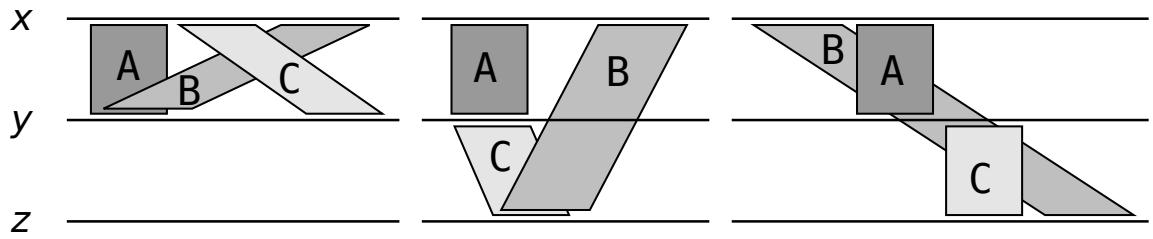


Figure 2.7: Inconsistency problems of the first `Tracker` approach. In the first subfigure showing direct inconsistencies, `Tracker` will find the A-B-contradiction while not regarding the B-C-crossing. In the second subfigure showing an indirect contradiction, `Tracker` will detect the pairwise A-B-inconsistency. Therefore, the consistent solutions are $\{A, C\}$ and $\{C, B\}$. The third consistent solution $\{A, B\}$ is not considered. The indirect crossing in the last subfigure is also not regarded.

Based on these results, the algorithm creates the consistence graph, see Figure 2.6 (e). The vertices consist out of the n alignments and two alignments are connected by an edge if there is no inconsistency between them. The final step is the determination of all maximal cliques in this graph, i.e. subgraphs where each vertex is connected with all other vertices. To this end, `Tracker` uses the Bron-Kerbosch algorithm (Bron and Kerbosch, 1973). The complexity of this algorithm corresponds to the worst case number of $\mathcal{O}(3^{n/3})$ maximal cliques in a graph with n nodes (Moon and Moser, 1965). This complexity is justified by the assumption, that the number of inconsistencies is usually low. This makes the consistency graph almost complete and the number of cliques as well as the time to determine them small. The so calculated cliques correspond to the maximal subsets of $\mathcal{A}$ that are consistent, see Figure 2.6 (f).

Besides the exponential worst case time and space demands, `Tracker` has two further drawbacks, see Figure 2.7. First, the algorithm is based on pairwise inconsistencies. Two pairwise inconsistent alignments can not be part of the same solution. This is also the case if both alignments have no overlap and the solution did not contain any alignments that connect them, i.e. if both alignments are consistent under the given circumstances. Therefore, not all possible solutions are considered. Second, the algorithm disregards inconsistencies based on crossings. Based on this limitations, the algorithm have to be classified as a heuristic, although it determines the exact solution of the clique problem.

2.2.2 Edge Consistency

This setting of the assembly of individual alignment edges to a multiple alignment is subject of intensive research about consistency-based alignment procedures. Based on joining alignment edges from pairwise alignments to a multiple alignment, i.e. to a consistent set of alignment edges, the approaches differ in the form of the assembly.

The T-Coffee algorithm

T-Coffee Notredame *et al.* (2000) stands for Tree-based Consistency Objective Function For alignmEnt Evaluation. This function creates a multiple alignment based on a library of pairwise alignments data from heterogeneous data sources. In the standard case of the first version of T-Coffee, these data came from global pairwise alignments calculated with ClustalW (Thompson *et al.*, 1994) and not necessarily consistent local pairwise alignments determined with Lalign(Huang and Miller, 1991). All edges of the calculated alignments are combined into a library. Each edge achieve a weight that corresponds to the pairwise sequence identity of the corresponding alignment. Equal edges are joined whereas the weight is summed up.

This library is then extended by the information of the other edges in the library. The weight of an edge is thereby enhanced if two other edges support it. The subsequent assembly of the multiple alignment is done by a progressive alignment strategy (Thompson *et al.*, 1994) that derives a distance matrix for all pairs of input sequences based on the global alignments. This matrix is used to create a phylogenetic guide tree by the neighbor-joining method (Saitou and Nei, 1987) that determines the order of the grouping during the multiple alignment process. For each alignment in a grouping step, the weights of the extended library are used. For an overview see Figure 2.8.

The multiple alignment produced by T-Coffee is a global alignment where conserved local motifs are supported by the higher scores in the extended library. The complexity of the procedure for m sequences where the length of each sequence is bounded by l is $\mathcal{O}(m^2l^2 + m^3l)$. This value is made up by the computation of the library in $\mathcal{O}(m^2l^2)$, the extension of the library in $\mathcal{O}(m^3l)$, the computation of the neighbor-joining-tree in in $\mathcal{O}(m^3)$ and the progressive determination of the multiple alignment in $\mathcal{O}(ml^2)$.

The DIALIGN algorithm

Unlike traditional alignment approaches that sum up substitution scores for aligned residues and subtract penalties for gaps, DIALIGN performs a DIagonal ALIGNment that focuses on comparing complete segments of sequences. The original version of the algorithm (Morgenstern *et al.*, 1998) first calculates all optimal pairwise alignments and extracts all gap-free segments, the so called diagonals. These diagonals are sorted to their overlap-weight scores that reflect the weight as well as the degree of overlap with other diagonals (Morgenstern *et al.*, 1996) in order to favor motifs occurring in more than two sequences. The resulting list of diagonals is then used to assemble a multiple alignment in a greedy manner. All diagonals are in the order their score checked for consistency and added to the alignment if consistent. Once a diagonal is added, it becomes part of the alignment and cannot be removed at any later stage. The final step is the introduction of gaps until all diagonals contained in the alignment are matched.

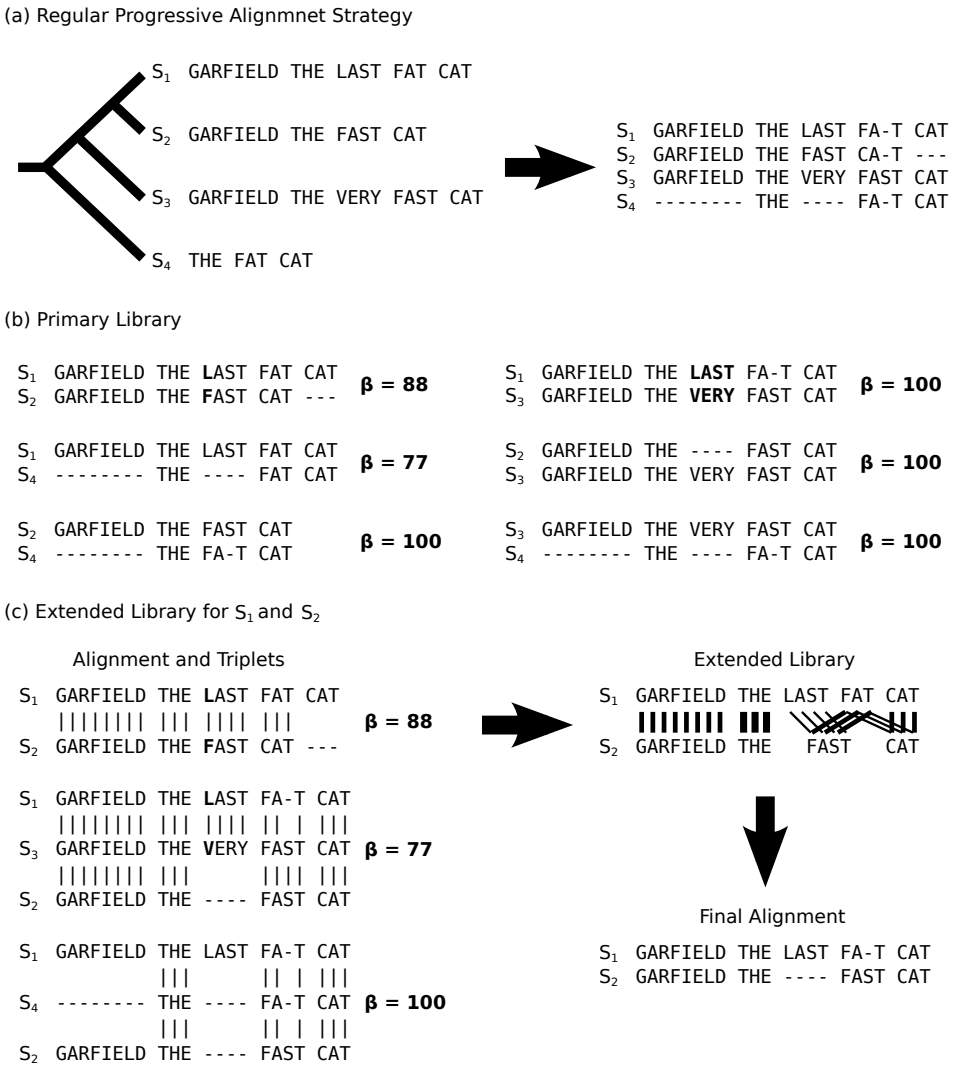


Figure 2.8: Example for library extension (Notredame *et al.*, 2000). (a) Progressive alignment. Four sequences have been designed. The tree indicates the order in which the sequences are aligned when using a progressive method such as `ClustalW`. The resulting alignment is shown, with the word CAT misaligned. (b) Primary library. Each pair of sequences is aligned using `ClustalW`. In these alignments, each pair of aligned residues is associated with a weight equal to the average identity among matched residues within the complete alignment (mismatches are indicated in bold type). (c) Library extension for a pair of sequences. The three possible alignments of sequence S_1 and S_2 are shown (S_1 and S_2, S_1 and S_2 through S_3, S_1 and S_2 through S_4). These alignments are combined to produce the extended library (the thickness of the lines indicates the weight). This library is resolved by dynamic programming to give the correct alignment.

The determination of a multiple alignment of m sequences with lengths smaller then l by `DIALIGN` can be done in $\mathcal{O}(m^4l^2)$. This is composed of $\mathcal{O}(m^2l^2)$ for the computation of the pairwise alignments, $\mathcal{O}(n^4l^2)$ for the calculation of the overlap weights for the $\mathcal{O}(m^2l)$

diagonals and $\mathcal{O}(n^2l^2)$ for the insertion of gaps.

Since the first version of DIALIGN, a variety of extensions and variations has been described. The successor DIALIGN2 (Morgenstern, 1999) uses an improved weight function for diagonals while DIALIGN-T (Subramanian *et al.*, 2005) uses a more context-sensitive approach that takes the overall significance of pairwise alignments into account. The most recent version DIALIGN-TX (Subramanian *et al.*, 2008) combines the greedy assembly with a progressive approach. In addition to this official DIALIGN versions, several related algorithms on segment based alignments have been developed. Lenhof *et al.* (1999) introduced an exact solution for the NP-complete segment-to-segment multiple alignment problem in terms of a branch-and-cut algorithm. Sammeth *et al.* (2003) combined the local segment based approach with a global divide-and-conquer strategy. The algorithm developed by Corel *et al.* (2010) determines local similarities shared by more than two sequences by using a min-cut, max-flow algorithm to identify highly connected positions in the sequence space in $\mathcal{O}(n^6l^4)$ for the worst case scenario.

Other Approaches

Besides the introduced algorithms that especially consider local motifs, a wide range of other approaches exist. For example, Gotoh (1990) introduced a method that determines consistent regions out of a set of single pairwise alignments over m sequences with maximal length l in $\mathcal{O}(m^3l)$ time. For the case that multiple alternative pairwise alignments are given for one sequence pair he uses this method to determine the combination of pairwise alignments that gives the greatest consistency in an exponential worst case time. Vingron and Argos (1991) identify common patterns in dot-matrices by a procedure based on matrix multiplication in $\mathcal{O}(m^3l^3)$ time for m sequences whose length is bound by l. Abdeddaïm (1997) developed an algorithm that maintains the transitive closure of an alignment graph over m sequences with maximal length l in $\mathcal{O}(m^2l + l^2)$ time. This procedure can be used by any greedy alignment algorithm to know in constant time if two characters are alignable or not.

2.3 Heuristic Algorithm

Since the structure of consistent alignment collections is an independence system we apply a greedy-like heuristic to find solutions for the Maximal Consistent Alignment Subset Problem. However, the union of two consistent collections can be inconsistent. Thus, a consistent alignment collection is not a matroid or greedoid and distinct maximal consistent subsets may have different cardinalities. A canonical greedy algorithm will therefore in general fail to find maximal consistent subsets (Helman *et al.*, 1993). We thus have to find a criteria for the greedy determination of consistent subsets which minimizes this effect.

Given a collection $\mathcal{A} = \{A_1, \dots A_n\}$ of n pairwise alignments of subsequences of a sequence

family S, we determine the heuristic solution iteratively trying to add one pairwise alignment $A \in \mathcal{A}$ after the other to the consistent subset $\mathcal{A}' \subseteq \mathcal{A}$. This is done by inserting the alignment A into the multiple alignment $M = \bigcup \mathcal{A}'^{\bullet}$ corresponding to the transitive closure of the union over all alignments in $\mathcal{A}'$. If the transitive closure of $M \cup A$ is still an alignment, A is consistent with the alignments in $\mathcal{A}'$ and we insert A into $\mathcal{A}'$. Otherwise A is inconsistent with $\mathcal{A}'$ and we refuse it.

In this strategy, an alignment once inserted is fixed and cannot be removed at a later stage. Since each inserted alignment influences the compatibility of later alignments, the order of insertion is determinative for the final result and hence crucial.

Intuitively, we are looking for an order where the final alignment collection $\mathcal{A}'$ implies a multiple alignment M that is "biologically correct". Of course, in real life we do not know if this is the case for a calculated M. Therefore, we use in our heuristic the score $\beta(M)$ of M that is defined as the sum $\sum_{A_i \in \mathcal{A}'} \beta(A_i)$ of the scores of the alignments in $\mathcal{A}'$ as an indicator of the "biological correctness".

Hence, we have to insert the alignments in such an order that $\beta(M)$ becomes maximal. Thereby, it is problematic to insert the alignments in the order of their scores. For example, the insertion of a high scoring alignment can prevents the insertion of a group of alignments that have smaller individual scores but a higher group score. This would result in a suboptimal solution. Instead, we have to prefer the insertion of alignments that allow other alignments to be inserted as well. Therefore, we combine the score of an alignment with the support by other alignments to an extended score. This idea is similar to the extended library in `T-Coffee` (Notredame *et al.*, 2000). By adding the pairwise alignments in the order of the extended score we try to minimize the cases where the greedy heuristic determines only suboptimal local maxima instead of the optimal solution of the MCASP.

2.3.1 Extended Scores

An alignment A is supported by a set of other alignments $\mathcal{A}'$ if and only if parts of the sequences aligned by A are also aligned directly or indirectly by the alignments in $\mathcal{A}'$. This is the case if the multiple alignment $M = \bigcup \mathcal{A}'^{\bullet}$ restricted to the subsequences aligned by A is itself a subalignment of A, i.e. if $M[V(A)] = A[V(M)]$. The *extended score* $\gamma(A)$ of alignment A is then defined as the basic score $\beta(A)$ enlarged by the scores $\sum_M \beta(A[V(M)])$ of all parts of A that are supported by sets of alignments with corresponding multiple alignment M.

Obviously the Maximal Consistent Alignment Subset Problem is part of the determination of extended scores. Thus, the determination of the extended scores is also NP-hard. However, it is biologically reasonable to assume that the evolutionary distances between conserved sequences are an ultrametric and hence satisfy the strong triangle inequality $\forall x, y, z : d(x, z) \leq max(d(x, y), d(y, z))$. In other words, if we have three sequences x, y and z and we have alignments between x and y and between y and z we should also have an alignment between

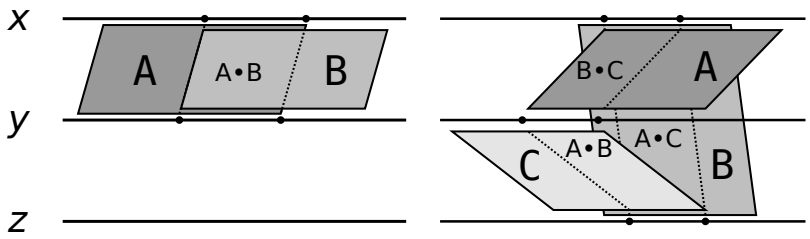


Figure 2.9: Direct and indirect support of alignments. In the direct case on the left picture the score of A and B is extended by the score of the overlapping part $\beta(A \bullet B)$. The same holds for for the indirect case on the right picture where A is extended by $\beta(B \bullet C)$, B by $\beta(A \bullet C)$ and C is extended by $\beta(A \bullet C)$.

x and z. This is because of the triangle inequality that tells us that the distance between x and z is smaller or equal than the distance between x and y and between y and z respectively. Otherwise it is likely that both given alignments are based on random similarities. Based on this assumption, it is sufficient to look only for supporting alignment sets with at most two alignments.

Therefore, we determine the extended scores for each alignment $A \in \mathcal{A}$ by determining the intersection $A \cap B$ for each alignment $B \in \mathcal{A}$ with $B \neq A$ and the intersection $A \cap (B \bullet C)$ for each pair of alignments $B, C \in \mathcal{A}$ with $B, C \neq A$. If the first intersection $A \cap B$ is not empty, we have a direct support and extend the score of A with the score of the overlapping part $\beta(A \cap B)$. This is done using linear interpolation of the basic score $\beta(A)$:

$$\beta_A(B) = \frac{|A \cap B|}{|A|} \times \beta(A)$$

Analog we have an indirect support if the second intersection $A \cap (B \bullet C)$ is not empty and extend the score of A by:

$$\beta_A(B \bullet C) = \frac{|A \cap (B \bullet C)|}{|A|} \times \beta(A).$$

For a graphical representation of the extension cases see Figure 2.9. All together we calculate the extended score by

$$\gamma(A) = \beta(A) + \sum_{B \in \mathcal{A}} \beta_A(B) + \sum_{B,C \in \mathcal{A}} \beta_A(B \bullet C)$$

2.3.2 Greedy Assembly

The first step for the assembly of the multiple alignment M that represents the (local) maximal consistent subset $\mathcal{A}' \subseteq \mathcal{A}$ is therefore, the determination of the extended scores γ for all alignments in $\mathcal{A}$. The alignments are then ordered by these scores and inserted into a queue

$\mathcal{Q}$. If alignments have the same extended score γ, we sort them by the basic score β. If this score is also equal, we sort by the input order.

Starting with the highest scoring alignment we check for all alignments A in $\mathcal{Q}$ whether A is consistent with the set of the already determined consistent alignments $\mathcal{A}'$. If this is the case, we update M as well as the consistent alignment set $\mathcal{A}'$ by A. If not, we refuse A and insert A into the set $\mathcal{I}$ of alignments that are inconsistent with the alignments in $\mathcal{A}'$.

Due to the iterative structure of the greedy approach, we describe the exact algorithm by induction over the alignments $A \in \mathcal{Q}$. In each step, we assure that M is a multiple alignment and that M corresponds to the consistent subset $\mathcal{A}' \subseteq \mathcal{A}$. We represent the alignments as intervals and the columns in M as thick columns which are again intervals.

Base case: In the beginning $\mathcal{Q} = \mathcal{A}$ and $\mathcal{A}'$ as well as $\mathcal{I}$ are empty. The multiple alignment M is the graph $(\mathcal{S}, \emptyset)$ and therefore a valid alignment.

Induction step: In the induction step, M consists of alignment columns where each column $c = \{[x, b_x^c, e_x^c] : 1 \leq x \leq m \wedge 1 \leq b_x^c \leq e_x^c \leq |S_x|\}$ is a set of intervals on the sequences. The columns satisfy the conditions of Definition 2.1 (p.27), i.e. each column is a transitive closure, each column has at most one entry per sequence and all columns underlie the partial order $\preceq$. In addition, we assume that the columns are ordered relative to $\preceq$.

We then take the highest-scoring alignment $A = \{[x, b_x^A, e_x^A], [y, b_y^A, e_y^A]\}$ from the beginning of the queue, update $\mathcal{Q}$ by $\mathcal{Q} = \mathcal{Q} \setminus \{A\}$ and check whether $(M \cup A_x)^\bullet$ is an alignment. If this is the case, we add A to $\mathcal{A}'$ and update M by $M = (M \cup A_x)^\bullet$. Else we update $\mathcal{I}$ by $\mathcal{I} = \mathcal{I} \cup \{A\}$ and leave M unchanged.

In practice, the insertion of A and the test of consistency are performed column-wise. We do this iteratively over all columns in M relative to the partial order $\preceq$. For each column c we check if A and c overlap. In this case, we insert the area of A or c in front of the overlap as a new column before c. Then, the overlapping parts of c, A and possible other columns d that A connects with c are joined. Finally, we insert the remaining part of the column as a new column behind c or continue with the remaining part of the alignment and the next column. This is done until A is empty, all columns are checked or we found an inconsistency. While in the first two cases A is consistent with M and we thus keep all changes, we undo all updates in the last case, remove the newly inserted columns in M and refuse A as being inconsistent. Furthermore, it is possible that A contains additional information about the relative location of the columns based on the partial order $\preceq$. In this case it necessary to adapt the order of the columns in M.

Since the exact actions depend on the location of the present column c relative to the alignment A, the first step for each column update is the determination of the positions of c for each of the two intervals of A. For each sequence interval $[i, b_i^A, e_i^A] \in A$ with $i \in \{x, y\}$ four cases can be distinguished (cf. Figure 2.10):

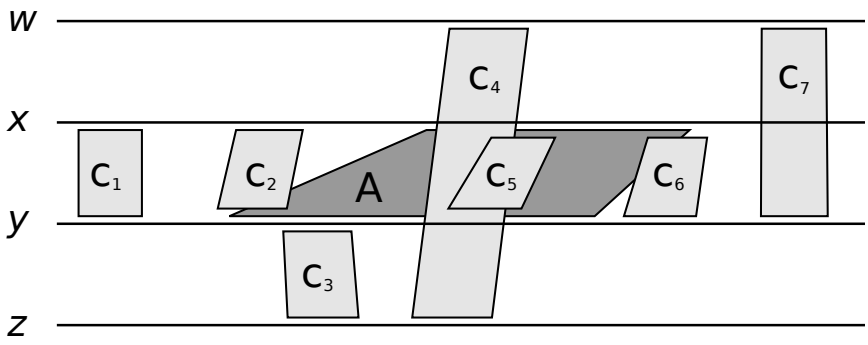


Figure 2.10: Possible locations of columns relative to the entries in alignment A. Column c_1 is a prefix, c_4 is independent, c_5 overlaps and c_7 is a suffix in both sequences x and y. The remaining columns c_2 (prefix in x, overlap in y), c_3 (independent in x, overlap in y) and c_6 (overlap in x, suffix in y) have different locations for the sequences x and y.

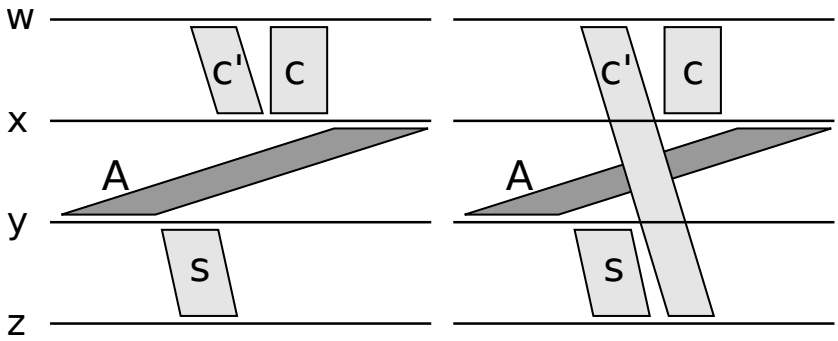


Figure 2.11: Switching columns. In the first case c and s are independent and we insert an alignment A with $c \preceq A$ and $A \preceq s$. Therefore, we know that $c \preceq s$ and we move all columns $c' \preceq c$ in front of s. In the second case the column c' additionally tells us that $s \preceq c$ so that an insertion of A would results in $c \preceq s$ and $s \preceq c$. This is a contradiction in form of a crossing and hence we refuse A.

independence: c contains no entry $[i, b_i^c, e_i^c]$. In this case we do not have any information about the $\preceq$-order of A and c at sequence i.

overlap: c contains an entry $[i, b_i^c, e_i^c]$ and we have $[i, b_i^A, e_i^A] \cap [i, b_i^c, e_i^c] \neq \emptyset$. Thus, c can be extended based on the information of A over the other sequence $\{x, y\} \setminus \{i\}$.

prefix: c contains an entry $[i, b_i^c, e_i^c]$ and we have $e_i^c < b_i^A$. In this case, c is in front of A for sequence i. This information is important to maintain the partial order $\preceq$. We therefore remember the column as the closest prefix s_i of A for sequence i detected so far.

suffix: c contains an entry $[i, b_i^c, e_i^c]$ and we have $b_i^c > e_i^A$. In this case, c is behind A for sequence i. If c is the first column behind A at i we remember c as the closest suffix s_i of A at i, corresponding to the prefix case.

An update of c is only necessary if the column overlaps with at least one sequence. All other

cases are not meaningful and we can continue with the examination of the next column, except for the case that one of the two following situations occurs: If c is a prefix at one sequence but there was already an earlier column which is the closest suffix s at the other sequence, A contains additional information $c \preceq s$ about the order of s and c. Thus, we have to move c and all smaller columns c' with $c' \preceq c$ between s and c to the front of s. If, additionally, s is smaller than c or if c is a prefix and a suffix at the same time, the insertion of A would create an inconsistency in form of a crossing and we therefore reject A, see Figure 2.11. The second special case occurs if c is the first suffix for both sequences or just at one sequence but there was already an earlier column that was the closest suffix for the other sequence. In this case, we can stop the column check since, according to the induction hypothesis, all following columns area suffix or independent.

If A and c overlap we assume, without loss of generality, that the overlap is at sequence x and denote the overlapping interval by $[x, \bar{b}_x, \bar{e}_x]$. In order to update c and A, we additional need to determine the corresponding overlap $[y, \bar{b}_y, \bar{e}_y]$ at sequence y of A and the corresponding overlap $[w, \bar{b}_w, \bar{e}_w]$ for all entries $[w, b_w, e_w] \in c$. We do this for y by mapping based on A, i.e. $\bar{b}_y = \mu^A_{x \to y}(\bar{b}_x)$ and $\bar{e}_y = \mu^A_{x \to y}(\bar{e}_x)$ and for w by mapping based on c, i.e. $\bar{b}_w = \mu^c_{x \to w}(\bar{b}_x)$ and $\bar{e}_w = \mu^c_{x \to w}(\bar{e}_x)$.

In general, the update itself consists of three steps: First, we separate the part in front of the overlap from the alignment or the column and insert it as a new column c' in front of c. More exactly, if A starts in front of c, i.e. if $b^A_x < \bar{b}_x = b^c_x$, the new column c' is the prefix $\{[x, b^A_x, \bar{b}_x - 1], [y, b^A_y, \bar{b}_y - 1]\}$ of A and A is set to the remaining part $\{[x, \bar{b}_x, e^A_x], [y, \bar{b}_y, e^A_y]\}$. Otherwise if c starts in front of A, we insert the prefix part $c' = \{[w, b^c_w, \bar{b}_w - 1] : [w, b^c_w, e^c_w] \in c\}$ of the column and update c by the remaining part $\{[w, \bar{b}_w, e^c_w] : [w, b^c_w, e^c_w] \in c\}$. The next step is the separation of the area behind the alignment from A or c. If A ends behind c, i.e. if $e^A_x > \bar{e}_x = e^c_x$ we shorten A to $\{[x, \bar{b}_x, \bar{e}_x], [y, \bar{b}_y, \bar{e}_y]\}$ and memorize the remaining part $\{[x, \bar{e}_x + 1, e^A_x], [y, \bar{e}_y + 1, e^A_y]\}$ for the check of the next columns as A'. If c ends behind A we append the prefix part $c' = \{[w, \bar{e}_w + 1, e^c_w] : [w, b^c_w, e^c_w] \in c\}$ as a new column behind c and set c to $\{[w, \bar{b}_w, \bar{e}_w] : [w, b_w, e_w] \in c\}$. Finally, if A contains additional information about column c, i.e. if c has no entry for sequence y, we insert $[y, \bar{b}_y, \bar{e}_y]$ as a new entry into c. See Figure 2.12 for a graphical representation of the possible cases.

Since it is possible that multiple columns have an intersection with A and since we allow a small error tolerance ε for alignments, our concept of overlap does not correspond to the intersection between the alignment and the column. Given the begin $\hat{b}_x$ and the end $\hat{e}_x$ of the intersection of A and c, we have to consider three cases for the determination of the begin $\bar{b}_x$ and the end $\bar{e}_x$ of the overlap:

1. First of all, it is possible that c already has an entry $[y, b_y, e_y]$ for the second sequence y of A. In this case the mapping positions at y can not only be calculated by mapping based on A but also by mapping based on c. We therefore denote the mapping values

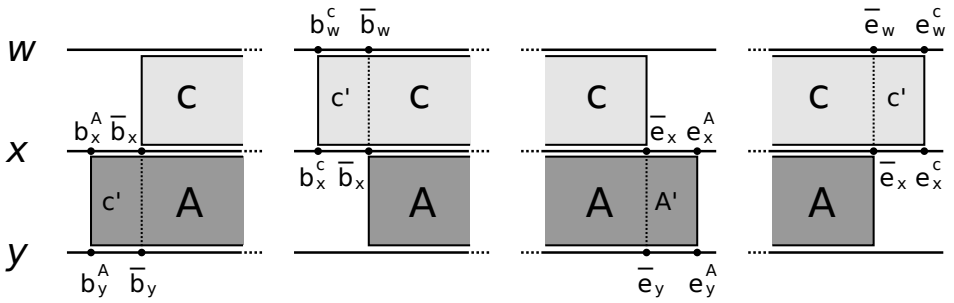


Figure 2.12: Update of a column c by an alignment A. Given the overlap at sequence x beginning in $\bar{b}_x$ and ending in $\bar{e}_x$ and the corresponding overlaps at w and y determined by mapping (labeled with a bar) together with the begin and end values of A and c, the first two figures represent the splitting of the alignment or the column in a prefix part c', which is inserted in front of c, and the remaining overlap part of A and c. The last two figures show the splitting at the overlap end where the suffix part A' of A is used for the update of the next column or the suffix part c' of c is inserted behind c.

based on A by $\bar{b}_y^A$ and $\bar{e}_y^A$ and the mapping values based on c by $\bar{b}_y^c$ and $\bar{e}_y^c$. If A is consistent with c the values of both mapping methods are the same and correspond to the real intersection $\hat{b}_y$ and $\hat{e}_y$. Otherwise A contradicts c and the maximum of the values $\delta_b = |\bar{b}_y^A - \bar{b}_y^c|$ and $\delta_e = |\bar{e}_y^A - \bar{e}_y^c|$ reflects the rate δ of this contradiction, see Figure 2.13 (a). Since the mapping values depend on the size of the intervals of c at both of the sequences, we assume, without loss of generality, that the x interval is smaller than the y interval. This is more stringent since the mapping values have a higher distance.

If δ is above a given error tolerance ε we regard A as incompatible and reject it. Otherwise, i.e. $\delta \leq \varepsilon$, the contradiction is small enough to be ignored and we update c. In this case, the overlap is defined as the intersecting area of A and c extended by all alignment positions the can be reached by mapping of column positions based on A. Formally for $i \in \{x, y\}$, $j \in \{x, y\} \setminus \{i\}$, $\bar{b}_i^A = \mu_{j \to i}^A(\hat{b}_j)$ and $\bar{e}_i^A = \mu_{j \to i}^A(\hat{e}_j)$ we have $\bar{b}_i = \min\{\hat{b}_i, \bar{b}_i^A\}$ and $\bar{e}_i = \max\{\hat{e}_i, \bar{e}_i^A\}$, see Figure 2.13 (b). The entry of c at sequence y is created by an earlier alignment. Since this alignment has a higher extended score and therefore is more trustworthy, we do not update this entry. Hence, we only insert the prefix part c' of A in front of the overlap and the remaining part A' of A behind the overlap, as described in the general case above.

2. The second case occurs if A overlaps with c at sequence x and is independent at sequence y but an earlier column is already the closest suffix s_y at y. Then A contains the additional information $c \preceq s_y$ about the order of s_y and c. Thus, we have to move c and all columns c' with $c' \preceq c$ between s and c to the front of s. If $s \preceq c$ as well, the insertion of A would create an inconsistency in form of a crossing and we would therefore reject

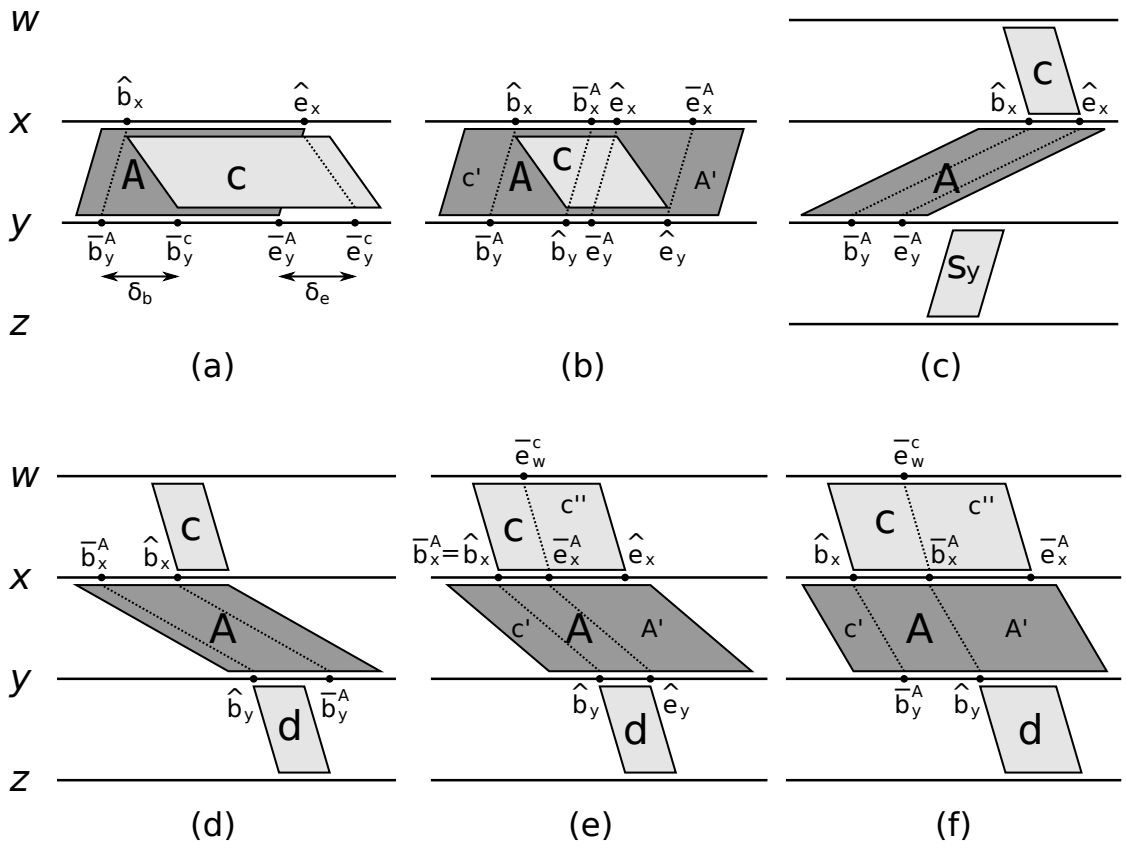


Figure 2.13: Determination of the alignment overlap. The figures illustrate the special cases that affect the assumed overlap: (a) determination of the contradiction rate, (b) overlap in case of contradictions, (c) order information by former column, (d) order information by later column, (e) merge of columns and (f) overlap end correction.

A. The subsequent update of c is done like in the general case with $\bar{b}_x = \hat{b}_x$ and $\bar{e}_x = \hat{e}_x$. See Figure 2.13 (c).

3. Finally, it is possible that A overlaps with c at sequence x, is independent at sequence y and we have no suffix for y so far. We then check the columns following c for the first column d that overlaps A in y. If no such column exists, c is updated as in the general case with $\bar{b}_x = \hat{b}_x$ and $\bar{e}_x = \hat{e}_x$. Otherwise we have three subcases depending on the position of $\hat{b}_x$ and the mapping $\bar{b}_x^A = \mu_{y \to x}^A(\hat{b}_y)$ of the intersection start $\hat{b}_y$ of A and d at sequence y onto sequence x:

   a) If $\bar{b}_x^A < \hat{b}_x$, A tells us that $d \preceq c$, see Figure 2.13 (d). If even $c \preceq d$ the insertion of A creates an inconsistency in form of a crossing and we reject A. Otherwise, we have to move d and all smaller columns d' with $d' \preceq d$ between c and d to the front of c. After the transposition we stop the update of c and continue with the update of d that is now in front of c.

b) If $\bar{b}_x^A = \hat{b}_x$, A connects the begin of column c with the begin of column d, see Figure 2.13 (e). In order to merge both columns, we first adapt the order by moving d and all smaller columns d' with $d' \preceq d$ between c and d to the front of c. Beginning at $\bar{b}_x^A = \hat{b}_x$, the overlap ends at $\bar{e}_x^A = \min\{\hat{e}_x, \bar{e}_x^A\}$ with $\bar{e}_x^A = \mu_{y\to x}^A(\hat{e}_y)$. After merging c and d along the overlap, the prefix and suffix parts of A, c and d are treated as in the general case.

c) If $\bar{b}_x^A > \hat{b}_x$, A extents the begin of column c, see Figure 2.13 (f). If column d starts in front of the end of column c, the columns have to be merged at the corresponding area. In order to perform the merging during the following column update, we set the end of the overlap to $\bar{e}_x^A = \min\{\hat{e}_x, \bar{b}_x^A - 1\}$ and update A and c as described for the general case.

After the column is updated, we set A to the remaining part A' of the alignment and continue with the next column in M until A' is empty. If $A' = \emptyset$, the insertion is complete and we stop the iteration over the columns in M. Otherwise, if A' is not empty after we have checked and updated all columns in M, we insert the remaining A' as a new column. The position of the insertion is behind the last closest prefix or, if no closest prefix exists, in front of the first closest suffix. If non of them exists, we insert A' as the last column in M.

After the successful insertion of a single alignment A, the columns in M are still transitive closures, have at most on entry per sequence and underlie the partial order $\preceq$. In other words, M satisfies the conditions imposed by Definition 2.1 (p.27) and is therefore still a multiple alignment. Therefore the alignments in $\mathcal{A}'$, which now also includes A, are consistent based on Definition 2.2 (p.29). If the insertion of A was not successful, M and $\mathcal{A}'$ remain unchanged. In both cases the induction hypothesis is satisfied.

Assembly end: The assembly is finished when all alignments in the queue $\mathcal{Q}$ are divided into the set $\mathcal{A}'$ of consistent alignments with corresponding multiple alignment M and the set $\mathcal{I}$ of alignments that are not consistent with $\mathcal{A}'$. Note that M is not a multiple alignment in the usual sense. According to Definition 2.1 (p.27) and the applied simplifications, M consists out of thick columns. Each thick column aligns consecutive sequence areas but not all parts of the sequences have to be part of a column. If a sequence area is not aligned by any of the consistent alignments in $\mathcal{A}'$ it is not part of M.

2.3.3 Alternative Solutions

The greedy assembly above determines a local optimal solution of the MCASP that maximizes the sum of the extended scores. In addition to this result, one could also be interested in alternative solutions. Obviously, the number of possible solutions is exponential to the number of alignments and we have to restrict the search. Also, we are interested in solutions that differ from the optimal solution by more then only a few alignments.

We combine this requirements for determining alternative solutions by performing the greedy assembly multiple times, starting each assembly with the insertion of the alignments that have been incompatible during all previous assemblies. More detailed, we perform the first assembly as described above. All following assemblies are then performed using two alignment queues $\mathcal{Q}_1$ and $\mathcal{Q}_2$ instead of just one. The first queue $Q_1 = \mathcal{I}$ contains the alignments that have been incompatible in all previous assemblies while the second queue $\mathcal{Q}_2 = \mathcal{A} \setminus \mathcal{Q}_1$ contains the remaining alignments. Starting each alternative assembly with Q_1 we divide the alignments into a new consistent set $\mathcal{A}'$ and a new inconsistent set $\mathcal{I}$. Then we continue the assembly with $\mathcal{Q}_2$ whereas consistent alignments are added to $\mathcal{A}'$ while inconsistent alignments are not inserted into $\mathcal{I}$.

Using this procedure, at most $|\mathcal{A}|$ different solutions are possible. By inserting first all compatible alignments that are not part of any previous solution, the results differ in a meaningful way.

2.3.4 Runtime and Memory Requirements

The amount of time and space that is needed to solve the MCASP by the heuristic approach depends on the number m of sequences $S_1, \ldots, S_m$, of the upper bound l for the length of these sequences and on the number n of alignments in the alignment collection $\mathcal{A} = \{A_1, \ldots, A_n\}$.

The calculation of the extended scores is done by checking all pairs and triple of alignments in $\mathcal{A}$ in $\mathcal{O}(n^3)$ time. The subsequent assembly with the construction of the multiple alignment M is done by inserting and updating alignment columns based on the alignments in $\mathcal{A}$. The number of these columns is obviously restricted by the number of elements in the site space and hence in $\mathcal{O}(lm)$. The update of a single (thick) column with m_i entries takes a constant number of operations for each entry and hence is in $\mathcal{O}(m_i)$ for the entire column. However, we have also $\sum_i m_i \leq lm$. Therefore, the effort of inserting a complete (local) alignment $A \in \mathcal{A}$ is bounded by $\mathcal{O}(lm)$.

The greedy heuristic thus produces a solution in $\mathcal{O}(n^3+nlm)$ time. In practice, however, the number of columns of $\mathcal{M}$ intersected by an alignment A is much smaller than the theoretical upper bound of $\mathcal{O}(lm)$. Also the calculation of the extended scores will be below $\mathcal{O}(n^3)$ since the check for support of the third alignment must only be performed if the first and second alignments overlap. For the determination of all alternative solutions we need $\mathcal{O}(n^4 + n^2lm)$ time. The amount of memory is determined by the size of the multiple alignment M and hence is restricted by the site space that is in $\mathcal{O}(lm)$.

2.4 Empirical Validation of the Algorithm

In order to evaluate the performance of our algorithm, we follow two different approaches. First, we use random alignment sets to demonstrate the correctness and the good runtime

behavior. Then we use biological data to determine consistent subsets and compare these results with the expected solution and other alignment programs.

2.4.1 Quality of Solutions

For the verification of our heuristic approach we compare our method with the NP-complete model that determines the optimal solution by analyzing each subset of alignments, See Section 2.2.1 (p.35). Because of the exponential running time of the optimal algorithm this comparisons have to be done with small alignment collections $\mathcal{A}$. Also, these small collections need to contain a high number of conflicts in order to induce a high number of possible solutions. Otherwise it would be easy for the heuristic to find the optimal solution and no significant conclusions about the correctness could be made.

Based on these requirements, we decided to generate artificial data sets $\mathcal{A}$. The idea is to use a set of m sequences S_1 to S_m with the same number l of motifs that are randomly distributed in order to create alignments for $\mathcal{A}$. We code each of the l motifs by a single number. If two different sequences, S_x and S_y, have the same number at position i and j, we create an alignment A between the two numbers and insert A into our initial alignment set $\mathcal{A}^+ = \{A = \{[x,i,i],[y,j,j]\} : S_x[i] = S_y[j]\}$. For the simulation of evolutionary differences between the motifs we do not use all of the initial alignments. Instead we insert the alignments in $\mathcal{A}^+$ with the probability $e/(m-1)$ into the final test set $\mathcal{A}$. The variable e with $1 \leq e \leq m-1$ corresponds to the expected number of adjacent edges for each element of the site space. Hence, high values for e describe evolutionary closely related sequences while small values correspond to distantly related sequences.

The crucial part for a high number of conflicts is the choice and the order of the l motifs in each sequence. As seen above, there are two different kinds of conflicts: contradictions and crossings. Contradictions arise if the same motif occurs multiple times in at least one sequence while crossings arise if the order of the motifs is different between the sequences. In order to create meaningful data sets, the sequences should consist of different motifs where some of them are conserved over some sequences. To combine all this, we randomly choose for each sequence l motifs based on the Poisson distribution. We set the parameter λ of the distribution, that corresponds to the expected value, in dependence of the motif that is determined. The probability p that the ith motif with $1 \leq i \leq l$ has the value k is given by

$$p_{i-1}(k) = \frac{(i-1)^k}{k!} e^{-(i-1)} \ .$$

This way the motifs described by small numbers are contained in most sequences while the motifs described by big numbers are likely to exist multiple times in one sequence. The last step is the concatenation of the motifs to the sequences, i.e. the determination of the order. This is done randomly in order to create crossings.

Model (m/l/e)	$\|\mathcal{A}\|$	$\|\mathcal{A}'\|$	Ex.	Hr.	Optimal (in %)	Direct (in %)	Error (in %)
3/8/1	10.65	5.36	21.62	3.64	86.80	60.40	2.61
4/3/1	5.96	4.16	6.74	2.08	96.40	88.00	0.67
4/3/2	11.60	6.84	29.11	3.06	87.20	70.40	2.05
4/3/3	15.72	8.66	51.40	3.91	96.40	78.80	0.42
4/4/1	7.58	5.01	12.21	2.36	96.00	82.80	0.80
4/6/1	10.90	6.54	32.65	2.95	87.60	68.00	2.08
4/8/1	14.09	8.02	63.80	3.38	72.80	47.60	3.59
5/8/1	16.22	10.00	140.58	3.05	65.20	47.60	4.48

Table 2.1: Results for the correctness analysis. The fist column gives the options that where used to generate the alignment collection. Here, m is the number of sequences, l the number of motifs per sequence and e the expected number of alignment edges per sequence and motif. The other columns gives the average values for the number of input alignments ($|\mathcal{A}|$), the number of alignment in the optimal solution ($|\mathcal{A}'|$), the number of exact solutions (*Ex.*), the number of heuristic solutions (*Hr.*), the percentage of how often the heuristic found the optimal solution (*Optimal*), the percentage of how often the first solution of the heuristic has been the optimal (*Direct*) and the rate of missing alignments in the best heuristic solution compared to the optimal solution in percent (*Error*).

In summary, we create the alignment collection $\mathcal{A}$ based on m sequences. Each sequence has l motifs, whereas the probability for the ith motif with $1 \leq i \leq l$ to be k is determined by the Poisson distribution $p_{i-1}(k)$ with the expectation $i-1$. The order of the determined motifs is set randomly where each position has the same probability. Alignments are created between all equal motifs and each alignment is inserted into $\mathcal{A}$ with the probability $e/(m-1)$.

The alignment collections $\mathcal{A}$ determined this way are used to compare our heuristic to the optimal solutions. We first determine all consistent subsets $\mathcal{A}'$ of $\mathcal{A}$ that are not a subset of another consistent subset, i.e. that are maximal, with the exact algorithm described in Section 2.2.1 (p.35). Then we use our heuristic to determine all alternative solutions. For the estimation of the quality of the heuristic solutions, we determine and compare following properties:

- $|\mathcal{A}|$: The number of alignments in the input alignment collection $\mathcal{A}$.
- $|\mathcal{A}'|$: The number of alignments in the optimal consistent subset of $\mathcal{A}$ determined with the exact algorithm.
- *Ex.*: The number of consistent maximal subsets determined by the exact algorithm.
- *Hr.*: The number of consistent alignment subsets determined by our heuristic.
- *Optimal*: This value is 1 if the heuristic has found an optimal consistent subset and 0 otherwise.

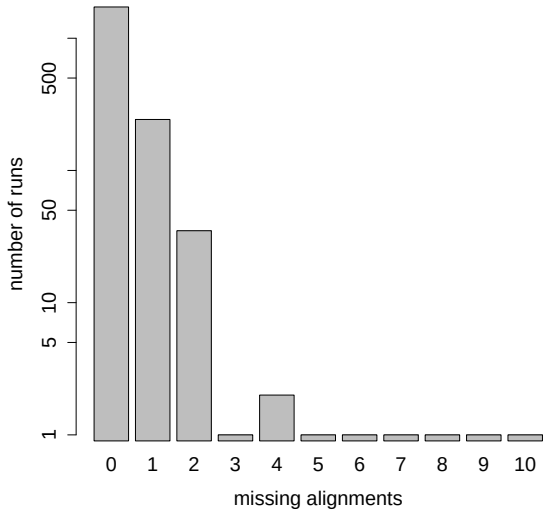


Figure 2.14: Correctness analysis. The bars indicate the number of runs (log scale) with the corresponding number of missing alignments the best heuristic solution.

- *Direct*: This value is 1 if the heuristic found the optimal consistent subset in the first place, i.e. not as an alternative solution, and 0 otherwise.

- *Error*: This value gives the rate of missing alignments in the best heuristic solution relative to the optimal solution.

The results for the different sequence numbers m, the motif numbers l and the expected alignment edges e per motif are summarized in Table 2.1. For each set of parameters we performed 250 runs. Based on the greedy approach, our heuristic determines in every case a correct consistent subset. In most cases, this subset is also maximal, i.e. the heuristic detects the optimal solution. This optimal solution is furthermore mainly found in first place. With a rising number of input alignments, i.e. higher numbers for m and l, the number of possible solutions grows as well. In these cases, the heuristic becomes worse. In contrast, for high values of e, which correspond to closely related sequences, we get an improvement of the correctness.

In summary, from the $2,000$ calculated test sets, the heuristic has one error in 242 cases, two errors in 34 cases and four errors in one case. In all other cases, the algorithm determined optimal results, see Figure 2.14. All optimal solutions of the test sets together have $13,659$ alignments while the errors sum up to 314 alignments. The corresponding error rate is 2.30 percent.

The worst result with the parameters $m = 5$, $l = 8$ and $e = 1$ has on average an optimal solution with 10 alignments where 4.48 percent of these alignments have been missing in the

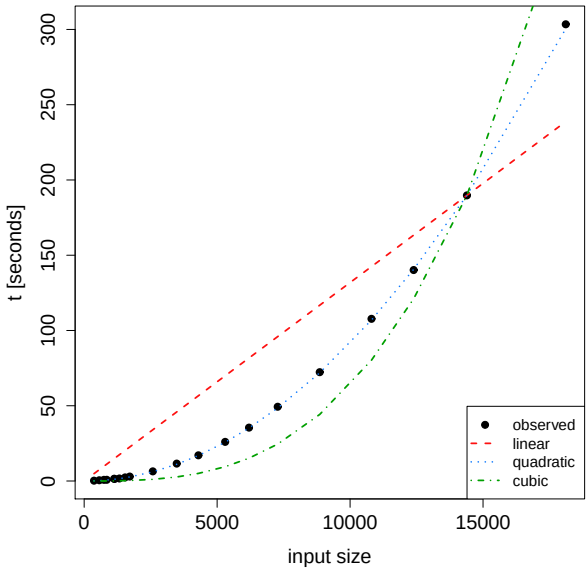


Figure 2.15: Runtime analysis. The measured time amounts in dependence of the size of the input alignment collection $\mathcal{A}$ (black line and points). For comparison we have given a linear (red), a quadratic (blue) and a cubic function (green) scaled by a linear factor so that the curves go through the penultimate data point. The measured runtime gives a good agreement with the quadratic function.

heuristic solution. If we extrapolate this value we have for an optimal solution with 223 alignments in average one missing alignment. Unfortunately, we cannot extend the comparison to bigger sets of input alignments since the computation of the exact result takes exponential time. But even with slightly growing error rates, this is a very good result and we can expect that our heuristic works.

2.4.2 Practical Runtime

In order to test the efficiency of our approach, we measure the amount of time t that our heuristic needs to determine all alternative results depending on different sizes of the input alignment collections $\mathcal{A}$. This collections are again randomly created as described above. We set the sequence number m to 20 and the expected alignment number e to the maximal value 19. The motif numbers l ranging from 1 to 100. The corresponding sizes of the input alignment collection start with 190 and reach for $l = 100$ a size of $18,111$ alignments. The measurements are performed on an Intel Xeon 2.13 GHz, 32 GB RAM, Fedora Linux computer and were repeated multiple times. The average amounts of time are summarized in Figure 2.15.

The runtime is pretty close to the quadratic curve. Since the complexity is mainly determined by the calculation of the extended scores, this is not surprising. Although for n alignments the complexity for this step is in the worst case in $\mathcal{O}(n^3)$, it is for practical usage

closer to $\mathcal{O}(n^2)$. During the calculation of the extended scores, the algorithm searches for pairs and triplets of alignments that support each other. In case of triplets, the algorithm looks for alignments that connect two overlapping alignments. If two alignments do not overlap, this search has not to be performed. For arbitrarily distributed alignments the number of alignments that overlap with a certain alignment is rather constant instead of being close to the worst case $\mathcal{O}(n)$. Hence, the expected complexity of the calculation of the extended score, and therefore of the whole algorithm, is likely to be in $\mathcal{O}(n^2)$. This is approved by the determined running times.

Summarized, our heuristic has a very good performance. The determination of consistent subsets with more then $18,000$ alignment is done in five minutes. The exact algorithm would have to check in this case around $2^{18000} \approx 10^{5400}$ subsets.

2.4.3 Biological Datasets

An evaluation of our heuristic on biological data is difficult. The problem here is that there does not exist any set of pairwise alignments with given consistent subset or multiple reference alignment. Instead existing benchmark data sets have been created for the evaluation of multiple alignment algorithms. Consisting of homologue sequences and a given multiple reference alignment of these sequences there is no possibility to extract the pairwise alignments that have been used to calculate the reference in a reliable manner. Nevertheless we can use these benchmark datasets to check our heuristic for the special case of the MCASP where the alignments consist of individual alignment edges. Therefore, we calculate all pairwise alignments for the given set of sequences $\mathcal{S}$ with an external alignment program and split this global alignments into the set $\mathcal{A}$ of individual alignment edges. The multiple alignment M that is created while determining the consistent subsets $\mathcal{A}' \subseteq A$ can then be compared to the given reference alignment M_{ref} over $\mathcal{S}$. Furthermore, this scenario allows us to compare our method to other state of the art multiple alignment programs.

As source for the sequences and reference alignments we use `BRaliBase II` (Gardner *et al.*, 2005). Consisting of five RNA families, namely *Group II Intron*, *5S RNA*, *SRP RNA*, *tRNA* and *U5 RNA*, this database was created to compare multiple sequence alignment programs upon structural RNAs. Each family has around 100 data sets where each set has 5 sequences with a length ranging from $60nt$ to $300nt$ and sequence identities ranging from below 25% up to almost 100%. For the comparison of M and M_{ref} we use the program `bali_score` which calculates the standard accuracy measures Sum of Pairs and Total Column (Thompson *et al.*, 1999). The sum-of-pairs score (SP) is defined as the rate of edges of the reference alignment M_{ref} that are also present in the calculated alignment M:

$$SP = \frac{|\{e : e \in E(M) \wedge e \in E(M_{ref})\}|}{|E(M_{ref})|}$$

Program	GII Intron	5S rRNA	SRP RNA	tRNA	U5 RNA
Hr. (ClW2)	73.77 / 62.55	92.88 / 86.12	87.10 / 76.90	86.27 / 75.48	79.58 / 64.89
Hr. (LocA)	76.35 / 63.54	**94.48** / **88.76**	**87.43** / **77.11**	**96.05** / **92.06**	**83.65** / **70.69**
ClustalW2	72.84 / 61.76	93.24 / 87.06	**87.43** / **77.17**	87.06 / 76.61	79.61 / 65.61
DIALIGN-TX	72.08 / 61.36	91.69 / 84.71	82.92 / 71.42	78.53 / 68.41	77.80 / 63.22
T-Coffee	**79.29** / **68.13**	**94.59** / **89.22**	87.31 / 76.77	**92.00** / **84.33**	**83.55** / **70.36**
MAFFT	77.20 / 64.02	93.83 / 87.78	87.10 / 76.75	90.14 / 83.02	80.43 / 66.35
MUSCLE	76.43 / 63.68	94.04 / 88.35	87.03 / 77.00	87.27 / 78.40	79.76 / 64.93
ProbCons	**78.69** / **66.99**	93.67 / 87.77	86.92 / 76.59	89.82 / 81.21	83.28 / 69.32

Table 2.2: The SP / TC scores in percent for our heuristic with ClustalW2 and LocARNA input alignments in contrast to different multiple alignment programs. The scores are determined with different RNA families in the BRaliBase II. The best and second best results are typed bold.

while the Total Column score (TC) is the rate of these columns c in M where all edges in c are aligned in M_{ref}:

$$TC = \frac{|\{c \in M : e \in E(c) \Rightarrow e \in E(M_{ref})\}|}{|\{c \in M\}|}.$$

For the computation of the pairwise alignments over the input sequences $\mathcal{S}$, which are needed as source for the edges in $\mathcal{A}$, we used two different programs. The first one is ClustalW2 (Thompson *et al.*, 1994), a program that determines a global multiple alignment based on sequence similarities by the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970). The second program is LocARNA (Will *et al.*, 2007). It is an enhancement of PMcomp (Hofacker *et al.*, 2004) and calculates pairwise local sequence structure alignments of two RNAs given by their base pair probability matrices by optimizing sequence and structure similarities at the same time, based on the Sankoff algorithm (Sankoff, 1985). We have chosen ClustalW2 since it also calculates multiple alignments in a progressive manner. The comparison of the quality of these multiple alignments and the quality of our alignments allows us to estimate the contribution of our heuristic to the correctness of the results. The second program LocARNA allows us the comparison with other state of the art alignment programs for RNAs that also considers the secondary structure.

Besides ClustalW2 (version 2.1) we use for the comparison the actual versions of the related programs DIALIGN-TX (version 1.0.2, with the '-D' option for DNA/RNA) and T-Coffee (version 7.97). Also the widely used programs MAFFT (version 6.851, with the local 'L-INS-i option), MUSCLE (version 3.8.31) and ProbCons (version 1.12) are used.

The results are summarized in Table 2.2. In general we observe that our method based on the sequence alignments with ClustalW2 and on the sequence structure alignments with LocARNA perform well and reach scores that are similar to the other programs. Regarding the sequence structure alignments, we have the best results for the structural *tRNA* and *U5 RNA*

Program	GII Intron	5S rRNA	SRP RNA	tRNA	U5 RNA
Hr. (ClW2)	76.57 / 65.61	93.18 / 86.38	**86.97** / **76.69**	87.52 / 76.83	81.05 / 67.07
Hr. (LocA)	78.18 / 65.37	**94.13** / **88.12**	**87.28** / **76.76**	**96.08** / **92.12**	**84.59** / **72.18**
ClustalW2	71.89 / 60.95	92.59 / 85.85	86.32 / 75.68	87.04 / 76.60	79.06 / 65.26
DIALIGN-TX	**79.46** / **69.22**	92.46 / 85.79	84.51 / 73.19	81.18 / 70.18	81.39 / 68.40
T-Coffee	**79.02** / **68.03**	**93.91** / **87.99**	86.65 / 75.86	**92.01** / **84.51**	**83.69** / **70.78**
MAFFT	78.17 / 65.28	93.23 / 86.63	86.10 / 75.21	90.35 / 83.13	81.00 / 67.77
MUSCLE	77.62 / 65.18	93.70 / 87.75	86.22 / 75.80	87.79 / 79.09	80.32 / 66.35
ProbCons	78.63 / 66.66	93.17 / 86.81	86.29 / 75.61	90.16 / 81.61	83.41 / 69.64

Table 2.3: The true positive SP / TC scores for the BRaliBase II data. This scores have been calculated by swapping the calculated and the reference alignment. The best and second best results are typed bold.

families and the second best results for *5S RNA* and *SRP RNA*. The only other program that achieved comparable quality was T-Coffee with the highest scores for the *Group II Intron* and *5S RNA* families and the second highest scores for *tRNA* and *U5 RNA*. Since T-Coffee performs the same score extension to determine biologically correct edges these results indicate that this method based on the support of alignments by other alignments is appropriated.

If we compare the results of the heuristic based on the ClustalW2 alignments with the direct results of ClustalW2 we observe that our method is worse in all cases except the *Group II Intron* family. At first glance, this is surprising even if the difference is small. Since the input of both methods consists of the same global pairwise alignments, and hence the same set of edges, we would expect an improvement of the quality in our method based on the extended scores. Nevertheless, this difference is not as surprising as it seems. Our heuristic uses only the information of the given edges for the assembly of the multiple alignment M, i.e. there is no edge in the multiple alignment that is not existent in at least one pairwise alignment. In contrast, the progressive assembly of M by ClustalW2 aligns sequences to the growing M. Therefore, new edges can arise. Since the scoring function prefers a mismatch instead of two gaps, the ClustalW2 alignments are more compact, i.e. with less columns and more pairwise edges. In order to affirm this assumption, we analyze only the true positive rate by swapping M and the reference alignment M_{ref}. Thus, we determine the rate of correct edges and columns in our alignments instead of the rate of reference alignment edges and columns that have been found by our heuristic. These results are summarized in Table 2.3. All heuristic scores are better then the ClustalW2 scores. The sequence structure results are even better than all other methods except for the *Group II Intron* data.

As already mentioned, for the evaluation of the arbitrary case of the MCASP no benchmark data sets exist. Instead we have to choose a different way. The idea is to define a set of sequences $\mathcal{S}$ consisting of related sequence subsets. By calculating local pairwise alignments between all sequence pairs in $\mathcal{S}$ we create our input alignments collection $\mathcal{A}$. This collection contains alignments between the related sequence subsets. Thereby these alignments should

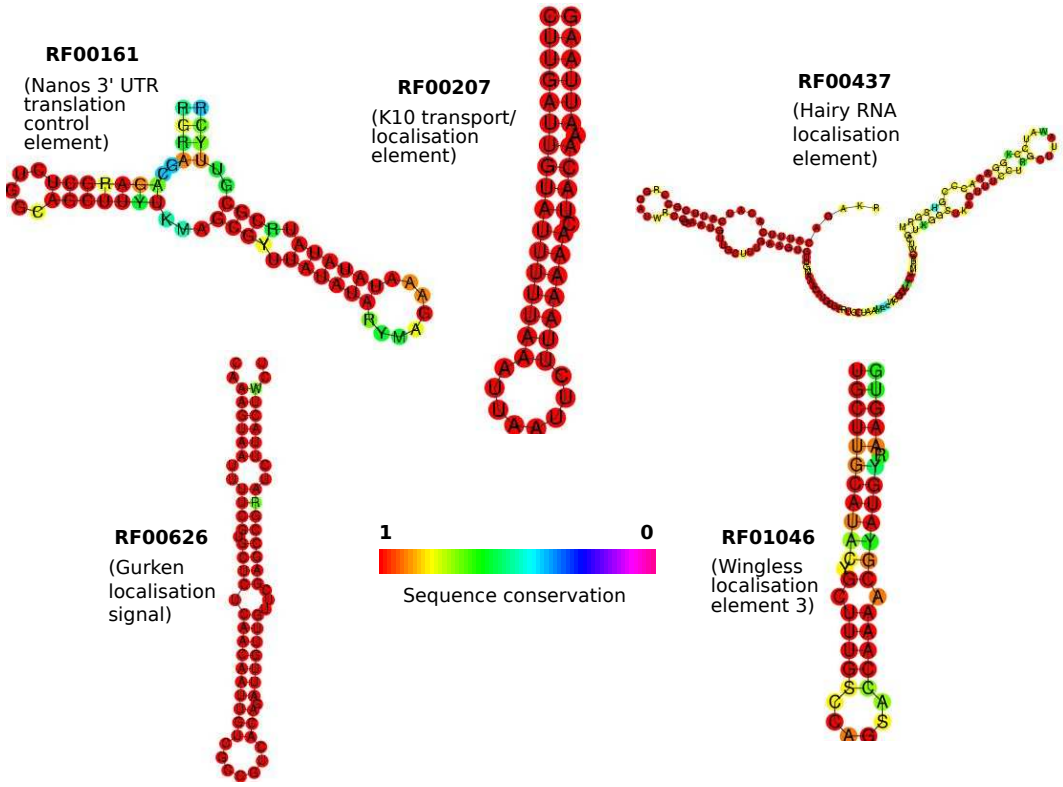


Figure 2.16: Consensus sequence and structure of the `Rfam` families used for the evaluation of the heuristic algorithm. All families are translation control or localization elements in Drosophila.

be consistent and there should be no alignments between sequences which belong to different related sequence subsets. In addition, we can expect random similarities and hence there will be some alignments in $\mathcal{A}$ that create inconsistencies. By using our heuristic to determine a consistent subset $\mathcal{A}'$ the corresponding multiple alignment M should consist of multiple columns where each column contains only sequence areas from related sequences.

As source for our sequence set $\mathcal{S}$, we used the `Rfam` database (Gardner *et al.*, 2009) that contains information about non-coding RNA families and other structured RNA elements. We use five ncRNA families and from each family we pick three sequences. Thus, our data set consists out of five related subsets where each subset consists of three sequences, see Figure 2.16. In order to enhance the probability of random similarity we furthermore enlarge the sequences by adding 100 nucleotides from the flanking region at both sides. For the computation of the local pairwise alignments between the ncRNA sequences we use again `LocARNA` (Will *et al.*, 2007). Based on the 15 sequences in $\mathcal{S}$ we determine an alignment collection $\mathcal{A}$ consisting of 105 alignments. Our heuristic determine 5 alternative solutions where the solutions with the highest score consist out of 43 alignments. All columns of the corresponding multiple alignment M consist mainly of sequences areas of one family. The

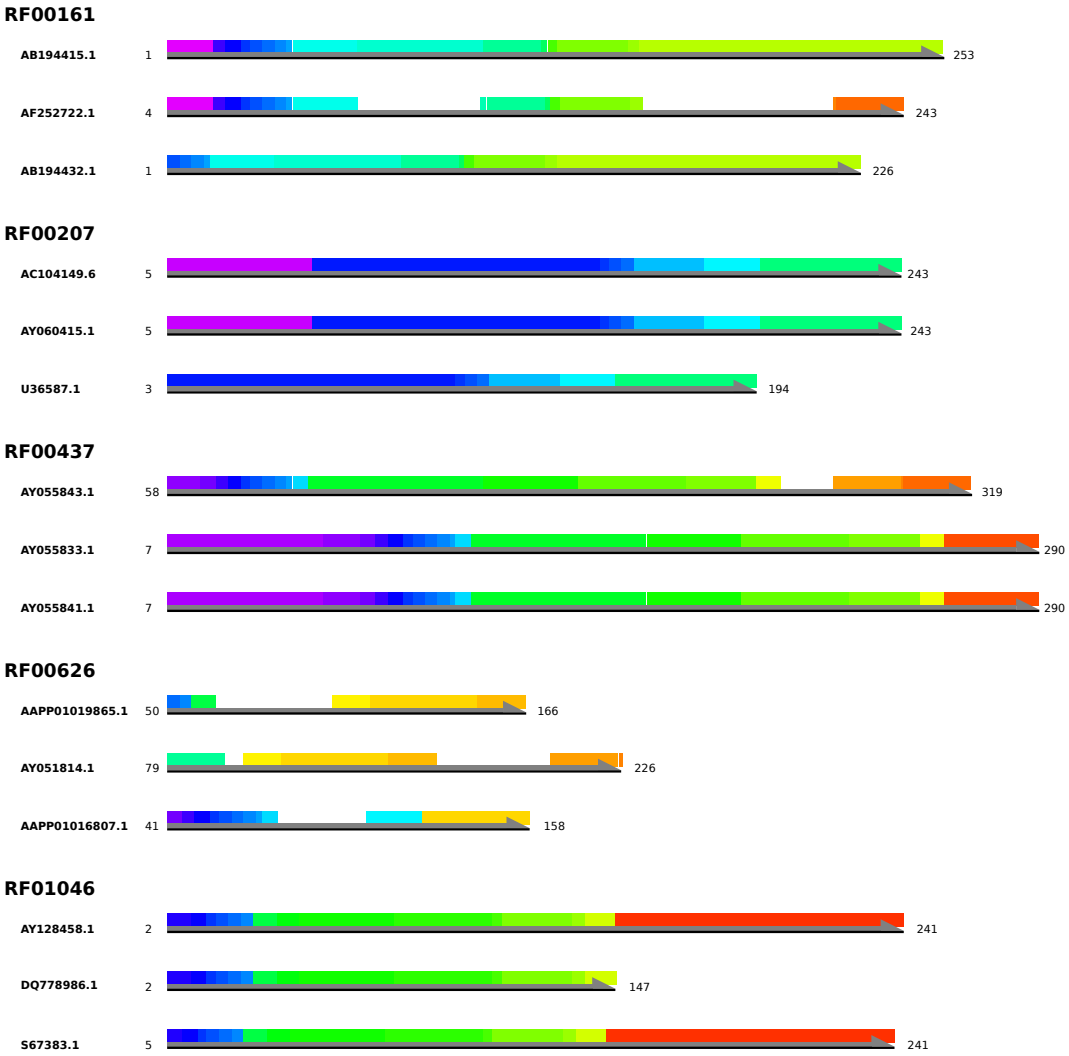


Figure 2.17: Multiple alignment of the `Rfam` evaluation. Areas with the same color are aligned in the same alignment column. The color itself codes the partial order of the columns, beginning with violet for the first column, the order follows the light spectrum, ending with red for the last column. Each Family has its own color pattern which is similar between all member sequences. Hence, the underlying consistent alignment subset reflects the biological relatedness of the sequences.

small number of exceptions is based on inter-familiar similarities that are not in conflict with the partial order of the family motifs, see Figure 2.17.

CHAPTER 3

Footprint Detection

> The universe, they said, depended for its operation on the balance of four forces which they identified as charm, persuasion, uncertainty and bloody-mindedness.
>
> Thus it was that the sun and moon orbited the Disc because they were persuaded not to fall down, but didn't actually fly away because of uncertainty. Charm allowed trees to grow and bloody-mindedness kept them up, and so on.
>
> Some druids suggested that there were certain flaws in this theory, but senior druids explained very pointedly that there was indeed room for informed argument, the cut and thrust of exciting scientific debate, and basically it lay on top of the next solstice bonfire.
>
> *The Light Fantastic*
> Terry Pratchett

Insights into the regulation of gene expression are crucial for discovering mechanisms of development and evolution. An important step towards this understanding is the capability of identifying regulatory sequence elements associated with a given gene. Based on the regulatory function, these elements are likely to be subject to stabilizing selection what makes them detectable by comparative sequence analyses (Tagle *et al.*, 1988) and phylogenetic informative (Prohaska *et al.*, 2004a).

Regulatory elements are located in the regulatory region, a non-coding sequence area around the gene and inside the introns, that can cover several thousand nucleotides (Dieterich *et al.*, 2002). In contrast to the region, the regulatory elements can be very short. Furthermore, they can undergo rapid changes that do not necessarily conform with the general phylogenetic relationships of the surrounding sequences (Chiu *et al.*, 2002; Schmidt *et al.*, 2010). This char-

acteristic makes it difficult for comparative sequence analysis to detect regulatory sequence elements. In order to detect small motifs, local alignments with a low stringency have to be calculated. This in turn produces plenty of alignments of which a large part is caused by heterologous random similarities. The rapid changes inside regulatory elements demands for additional input sequences to get additional evolutionary information which again increase the complexity of the calculation.

In the previous chapter we pointed out, that the support of alignments by other alignments as well as the consistence of alignments in respect to the linear order of the sequence sites can be used to detect homologous motifs. Based on these observations, we developed an algorithm that is able to separate alignments between phylogenetic footprints from alignments between unrelated random similarities by the assembly of a multiple local alignment. In this chapter, we use this method as basis for the development of the program `Tracker` for detecting phylogenetic footprints. In contrast to other algorithms for phylogenetic footprinting, see Section 1.3.2 (p.14) in the introduction, `Tracker` determines a local multiple alignment containing only conserved motifs. Additionally, `Tracker` maintains the linear order of the underlying sequences in order to ensure homology. This, as well as the efficiency, are significant advantages compared to approaches based on motif overrepresentation, see Section 1.3.2 (p.16) in the introduction.

3.1 The `Tracker` Algorithm

The algorithm is based on pairwise comparisons over all pairs of homologous sequences. It is, however, not necessary to compare the complete sequences. For example, if we are looking for homologous regulatory elements and the sequences contain a gene, we are in most cases not interested in the comparison of the exons of this gene. Instead we compare only the areas in front of the gene among themselves, the introns among themselves, and the areas behind the gene among themselves. Therefore, we define regions of homology, divide the sequences in disjunctive fragments that contain the areas of interest and compare only fragments that belong to the same region. In order to describe different scenarios, a region can consist of multiple fragments or a fragment can be part of multiple regions. This is, for example, useful if we combine multiple introns inside one region or if it is not possible to split a part of one sequence in multiple fragments because of a missing gene. See Figure 3.1.

The input of `Tracker` therefore consists of a fragment family F of p disjunctive fragments F_1 to F_p, that are defined on the m homologous sequences $S_1, \ldots, S_m$ under consideration. Furthermore, two tables ψ and ϕ have to be defined. The first table $\psi : \mathbb{N} \to \mathbb{N}$ assigns each fragment index i to the corresponding sequence index $\psi(i) = x$ with $1 \leq x \leq m$. The second table $\phi : \mathbb{N} \to \mathcal{P}(\mathbb{N})$ assigns each fragment index i to a set of regions $\phi(i) = \{y : i \text{ is in region } u\}$ with $1 \leq u \leq r$ where r is the number of different regions defined by the

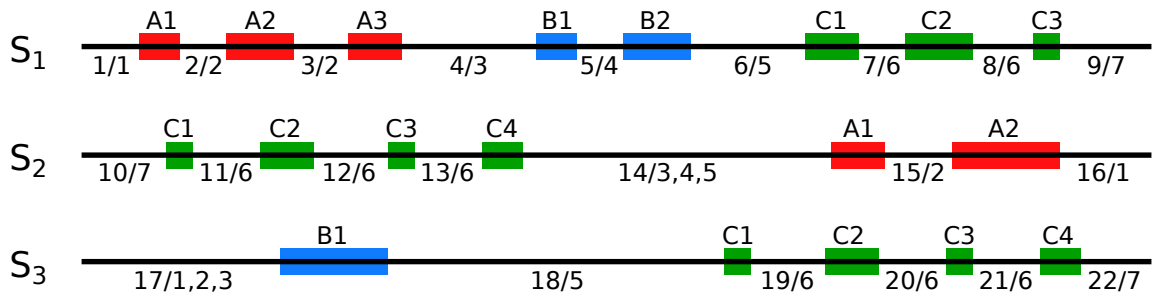


Figure 3.1: Example for the Tracker input consisting of $m = 3$ homologous sequences S_1, S_2 and S_3, where sequence S_2 is given in the reverse direction. The sequences contain three homologous genes A, B and C that consist of multiple exons. Each exon is indicated by the gene name and the index of the exons. Note that gene B is missing in S_2 and gene A is missing in S_3. Based on these genes, we define $r = 7$ regions, corresponding to the area in front of A (1), the introns of A (2), the intergenic region (IGR) between A and B (3), the introns of B (4), the IGR between B and C (5), the introns of C (6) and the area behind C (7). The $p = 22$ fragments corresponding to the regions are marked by their index and, separated by a slash, the regions they belong to. Note that, for example, the fragments 11 to 13 belong to the same region 6 since all the fragments are introns of the same gene. In contrast fragment 14 belongs to multiple regions since the B gene is missing in sequence S_2. It is also possible that a sequence does not contain all regions, as in the case of sequence S_3, where gene B has no introns and hence the corresponding region 4 does not exist.

user. For convenience, there are some alternative possibilities to define the Tracker input. A detailed overview about the exact input and output syntax, possible parameters and the availability can be found in Section B (p.135) in the appendix.

The subsequent computation of phylogenetic footprints consists of two steps. In a first step, we determine the initial data set of pairwise alignments over the homologous regions and perform optional pre-processing steps in order to enhance the quality of these alignments. The second step is the assembly of the alignments using the algorithm described in Chapter 2 (p.23) and the evaluation of the motifs.

3.1.1 Calculation and Processing of the Pairwise Alignments

The algorithm starts with the calculation of the initial data set of pairwise alignments. Given r homologous regions over m sequences, we calculate for each region $1 \leq u \leq r$ and for each pair of sequences $1 \leq x < y \leq m$ all local alignments between all fragments i and j with $\psi(i) = x$, $\psi(j) = y$ and $u \in \phi(i) \cap \phi(j)$.

Pairwise Alignments

Since genomic sequences can be very long, we use here a seeding strategy for the alignment calculation. Alignment algorithms based on seeds first compute a set of all possible seeds, i.e. all words of a certain length that match a pattern in the query sequence with sufficient score.

Then, the algorithms search in linear time for all gap-less motifs that match one of the seed patterns and elongate this seeds in both directions to obtain high scoring segment pairs. In case of consecutive patterns, as used in `BLASTN` (Altschul *et al.*, 1997), the algorithm searches for identical words of a certain length. Additionally, some degree of degeneracy can be allowed to identify nearly exact words (Brudno *et al.*, 2003). For example, an eight-consecutive model with two levels of degeneracy will identify all subsequences that have at least six identical nucleotides. In case of non-consecutive or two weighted-space patterns, the algorithm detects seeds where the positions for required matches and allowed mismatches are strictly specified. For example, `BLASTZ` (Schwartz *et al.*, 2003) uses a 12of19 two weighted-spaced pattern of 1110100110010101111, where 1 stands for a required match and 0 for a position that does not matter. This precise combination was shown to be most sensitive among all combinations of the 12of19 seeds (Ma *et al.*, 2002).

Here, we use the `LASTZ` algorithm, a drop-in replacement for `blastz` (Schwartz *et al.*, 2003). Per default, we search for seeds with a low stringency, in detail eight matches or transitions in all positions and report all segment pairs with a score of at least 1000. Also we search per default both strands for conserved regions and we treat ambiguous IUPAC characters like an 'N'. Besides the default parameters that are used to minimize the number of false negatives, `Tracker` provides the possibility to use other parameters for the `LASTZ` search.

The resulting set $\mathcal{A}$ of the `LASTZ` alignments can contain alignments that consist of highly conserved blocks that are separated by divergent stretches or that align repetitive elements. Therefore, it is possible to enable a pre-processing parameter. In this case, `Tracker` continues with the processing of $\mathcal{A}$ in order to remove unconserved and repetitive areas. Since the `LASTZ` segment pairs are given as sequence coordinates, it is necessary to determine first the exact pairwise alignments for each segment pair. This part of `Tracker` is generic and each pairwise alignment algorithm could be used. In our implementation, we use the conventional sequence alignment algorithm `ClustalW2` (Thompson *et al.*, 1994) since it is a well-known standard program that performs for this task well and that can be easily included in our program.

Removing of Unconserved Areas

In order to remove the unconserved areas, we process each alignment A in $\mathcal{A}$ by a sliding window of length 12. For each window starting at position i we determine the sequence identity w_i. If w_i is higher than 0.75, we mark each position in the window as highly conserved while unmarked positions in windows with w_i lower than 0.40 are marked as unconserved. Subsequently, all unconserved regions are removed while all remaining parts that contain a highly conserved part are inserted instead of the original alignment A.

Removing of Repetitive Areas

For removing repetitive elements, we use a local entropy measure. Since complex repetitive elements that are conserved could be functional, this simple method is a good choice in order to remove only repetitive elements with low complexity. Technically, we again use a sliding window w of length 20 over each sequence of the corresponding pairwise alignment. Inside w, we determine the nucleotide frequencies f^x for all nucleotides x and the joint frequencies $f^{x\delta y}$ for all pairs of nucleotides x and y with the distances $1 \leq \delta \leq 5$. Based on these values, we calculated for each position i in w the entropy

$$H_i = -\sum_x f_i^x \log_2 f_i^x$$

and the joint entropy

$$H_i^\delta = -\sum_{xy} f_i^{x\delta y} \log_2 f_i^{x\delta y} \ .$$

If the resulting mutual information measure

$$M_i = \frac{1}{\delta} \sum_\delta H_i^\delta - H_i$$

is smaller then 0.75 or the entropy H_i is smaller then 1.25 we regard position i as repetitive. These threshold values have been determined by evaluating test sets. Finally, we remove all parts of the corresponding alignment that have been marked as repetitive in one of the both sequences.

After the removal of repetitive areas, the pre-processing is finished. Note that for big data sets, the calculation of the exact alignments by `ClustalW2` can take much time. In these cases it is better to omit the optional processing steps and to continue directly with the calculation of the multiple, local alignment.

3.1.2 Determination of the Multiple, Local Alignments

The alignments in the initial alignment collection $\mathcal{A}$ are represented as interval pairs $A = \{[i, b_x, e_x], [j, b_y, e_y]\}$. Note that, in contrast to the previous chapter, i and j here denote the index of the fragment instead the index of the sequence. For the sake of simplicity, we also omit the case distinction for the strands for the description of the algorithm. During the real assembly of the multiple alignment M by `Tracker`, the first inserted alignment that contains a fragment of sequence x determines the orientation of x in M. If the alignment orientation is the reverse complement of the orientation given by the input, we adopt the comparisons during the assembly. In detail, we say that an interval $[i, b, e]$ is "in front" of a second interval $[i, b', e']$, if the orientation of both is equal to the input and if $e < b'$, or if the orientation of both is the reverse compliment of the input and if $e' < b$. The analogous case yields for

the definition of the term "behind". If the orientation of intervals is not equal, they are incompatible. Hence, the multiple alignment M is created by alignments where all intervals of the same sequence have the same orientation. For the further description, we assume therefore, without loss of generality, that all alignment intervals have the same orientation as the input sequences.

Clusters of Independent Subsets

In order to minimize the effort for the assembly of the multiple alignment, we first split the alignment collection $\mathcal{A}$ into independent subsets to which we refer as *clusters*. Therefore, we start with an arbitrary alignment A in $\mathcal{A}$ and insert it into the new cluster $\mathcal{C}$. Then we use this alignment as representative and insert all remaining alignments B in $\mathcal{A}$ that overlap with A into $\mathcal{C}$. Two alignments $A = \{[i, b_x^A, e_x^A], [j, b_y^A, e_y^A]\}$ and $B = \{[j, b_y^B, e_y^B], [k, b_z^B, e_z^B]\}$ overlap, if and only if they have a common fragment j and if the aligned areas of this fragment overlap, i.e. if $b_y^A \leq e_y^B$ and $b_y^B \leq e_y^A$. After all overlapping alignments are inserted, we continue with the next alignment in $\mathcal{C}$ as representative and repeat the procedure until all alignments in $\mathcal{C}$ have been used as representative. In this case, $\mathcal{C}$ contains all alignments that overlap directly or indirectly with the first inserted alignment. We repeat this procedure with a new cluster until all alignments in $\mathcal{A}$ are subdivided into clusters.

Inconsistencies based on contradictions between alignments of different clusters are not possible by construction. Nevertheless, alignments of different clusters can create crossings. Therefore we compare all pairs of clusters and join them if they cross each other. Two clusters $\mathcal{C}$ and $\mathcal{C}'$ create a crossing if they contain alignments $A, B \in \mathcal{C}$ and $A', B' \in \mathcal{C}'$ and if A is in front of A' in one sequence and B behind B' at another sequence. After this step, all clusters are independent since inconsistencies can only be created by alignments within the same cluster. Clusters mostly correspond to homologous regions. In cases where fragments belong to multiple regions, a cluster can contain multiple regions while in cases where the sequences have high evolutionary distances, fragments of the same region can be part of different clusters.

Consistent Subsets and Multiple Alignment Assembly

For the calculation of the multiple alignment that aligns conserved local motifs in thick columns, we use the algorithm introduced in Chapter 2 (p.23) to determine the best and all alternative solutions of the Maximal Consistent Alignment Subset Problem for each cluster $\mathcal{C}_i$. The alignments in $\mathcal{C}_i$ define the input. Since they contain the fragment index instead of the sequence index, we furthermore translate the indices x into the sequence indices $\psi(x)$. The error tolerance ε can be set via a parameter. If no value is given, the algorithm takes the highest difference between the length intervals of the initial alignment collection, i.e. $\varepsilon = \max_{A \in \mathcal{A}}\{\delta\}$ with $\delta = |(e_x - b_x) - (e_y - b_y)|$ for each alignment $A = \{[x, b_x, e_x], [y, b_y, e_y]\}$ in

$\mathcal{A}$. This value is an approximation for the highest number of gaps in the alignments and hence a good approximation for acceptable shifts. The resulting consistent subsets for each cluster are sorted in a descending order by their score, i.e. the sum of the scores over alignments in the consistent subset. Since the alignments in each solution are mutually consistent, i.e. each alignment is consistent with every other alignment, we denote the solutions as *cliques*. The expression $\mathcal{C}_i^j$ stands for the clique with the jth highest score of cluster i. The corresponding multiple alignment is denoted by M_i^j.

Smoothing of Thick Columns

Each resulting alignment M consist of thick columns $c = \{[z, b_z^c, e_z^c]\}$, where each column is a set of intervals that are defined by the index of the sequence z, the beginning of the interval b_z^c and the end of the interval e_z^c. Columns represent local conserved motifs and by construction, two columns c and d differ either in the composition of their sequences , i.e. $\{z : [z, b_z^c, e_z^c] \in c\} \neq \{z : [z, b_z^d, e_z^d] \in d\}$, or they include both an interval over the same sequence z that are separated by a gap, i.e. $\exists_z [z, b_z^c, e_z^c] \in c \wedge [z, b_z^d, e_z^d] \in d \wedge (e_z^c + 1 < b_z^d \vee e_z^d + 1 < b_z^c)$. In a biological sense, these differences can be interpreted as evolutionary events such as insertions or deletions of sequence parts. Hence, the sequence areas aligned in one column can be expected to share the same evolutionary history.

Nevertheless, during the assembly of the multiple alignment M, we determine splitting positions inside an interval by linear interpolation and we allow an error tolerance $\varepsilon > 0$. Therefore, the boundaries of the intervals are only approximations, and it is possible that prefix or suffix areas of intervals are more likely to be part of another column with a different evolutionary history. For an example see Figure 3.2 (a). The left part shows the true evolutionary history between column c, d and d' in form of a combined multiple alignment between c and d and between c and d'. As you can see, some suffixes of c are more likely to be part of d or d' while prefixes of d and d' are likely to be part of c. Unfortunately, we only know the interval boundaries determined by the heuristic, as shown in the right part of the figure. The next step in our algorithm is therefore the correction of these misplacements by adopting the boundaries between adjacent intervals.

The optimal solution for this problem would be a rearrangement of the interval boundaries, i.e. the exchange of prefixes and suffixes between intervals, that maximizes the sum-of-pair scores over all columns. Obviously, an algorithm for this problem would also be able to solve the general alignment problem. This makes the smoothing problem NP-complete (Elias, 2006) and we cannot expect to find optimal solutions in an efficient manner. In order to create a useful heuristic, we first summarize some helpful facts:

- By construction, misplaced interval boundaries are only possible between adjacent intervals, i.e. intervals in different columns that are defined on the same sequence and where the beginning of one interval is directly behind the end of the other interval. In

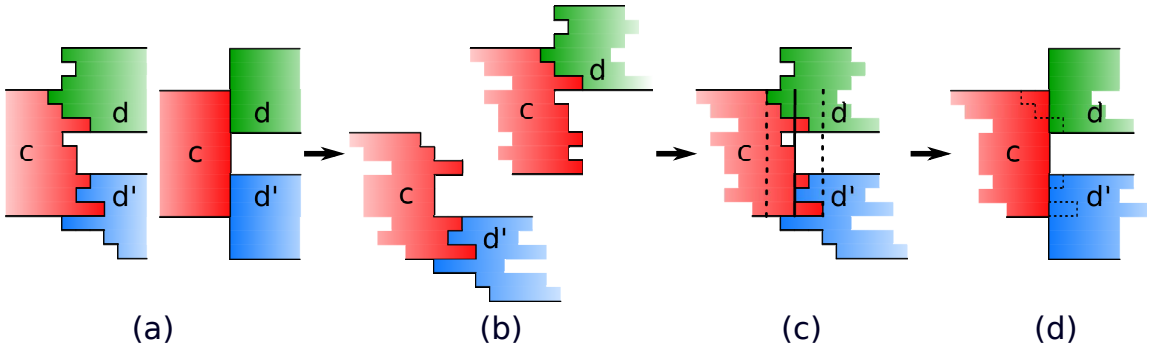


Figure 3.2: Smoothing of columns. The interval boundaries between the thick columns c (red) and d (green) and between c and d' (blue) do not correspond to the relationship of the sequences as indicated by the similarities shown by the combined alignment (a, left figure). Note that the combined alignment contains only information about the similarities between c and d respective d'. It contains no information about d and d' even if d is displayed above d'. Unfortunately, the true adjustment is hidden and we know only the interval boundaries as determined by the heuristic assembly (a, right figure). In order to correct the boundaries, we take the last ε sequence sites of the c intervals and align them with the first ε sequence sites of each adjacent column d and d' (b). Both resulting multiple alignments are combined to a single one, whereas the highest scoring alignment (here the c-d alignment) guides the exact arrangement of the c intervals (c). Now we compute for each alignment column of the combined alignment the sum-of-pair score of all the thick columns under the assumption, that the considered alignment column defines the new end of c. Thereby, it is sufficient to restrict the computation to columns between the last column that contain only sites from c and the last column that contains at least one site of c (c, black doted lines). The boundaries of the intervals are then adapted in respect to the column with the highest sum-of-pair score (c, black continuous line). The resulting intervals correspond now two the biological history of the motifs (d, original interval boundaries are indicated by black dots).

all other cases, the heuristic does not perform a split of a column and the boundaries are unambiguously defined by a pairwise alignment of the input alignment collection. Hence, we only have to correct boundaries of adjacent intervals.

- If we correct the boundaries at the end on one interval, we simultaneously correct the beginning of the adjacent interval. Hence it is sufficient to correct only the interval ends of all columns.

- The correction of interval boundaries between two columns also affect the optimal adjustment of boundaries of other columns with adjacent intervals. For example, if we move the boundaries between c and d as in Figure 3.2 towards c, the sum-of-pair score of the three columns will be maximal if we also move the boundaries between c and d' towards c. Hence, changes of boundaries have side effects and we should correct boundaries only once to avoid infinite loops.

Based on this information, we use again a greedy approach where we iteratively correct the

interval ends of each column. Clearly, the system of thick columns is also not a matroid or greedoid and we cannot expect to find optimal solutions. Therefore, we try to minimize corrections that are disadvantageous in a global scale by starting with the correction of columns that have a high influence onto the final sum-of-pair scores over all columns, i.e. columns that have high scores itself. Hence, we correct the interval boundaries iteratively for all columns in the order of their score.

For the correction of one single column c, we first determine all columns d that contain an interval that is adjacent to one interval in c. For each of these adjacent columns, we need to know the arrangement that corresponds to the evolutionary history between c and d in order to adapt the interval boundaries. Therefore, we calculate a multiple alignment of the sequences in c and d, whereas adjacent sequences are first concatenated. Instead of aligning the complete sequences, it suffices to analyze only the areas within the error tolerance ε. Hence, we restrict the calculation of the multiple alignment to the last ε sites of c and the first ε sites of d, cf. Figure 3.2 (b). For the generic step of the alignment calculation, we use again the progressive multiple alignment program `ClustalW2` but other multiple alignment programs can be used as well.

The next step is to merge the multiple alignments of the different adjacent columns d into a single multiple alignment M^c, cf. Figure 3.2 (b) and (c). If c has only one adjacent column, we already have the result. Otherwise, we use the multiple alignment with the highest score as a guide. In this case we remove for each multiple alignment all sequences that are only based on c and join the remaining parts as indicated by the arrangement of the c intervals in the guiding alignment.

The arrangement of the thick columns in M^c corresponds to the evolutionary history between them, cf. Figure 3.2 (c). The final step is the determination of that alignment column of M^c as new end for thick column c, that corresponds to this history. In other words, we are looking for that column x in M^c, that maximizes the sum-of-pair scores of the thick columns if we set the boundary between c and the adjacent columns d behind x. Obviously, that column x is located between the last alignment column of M^c that contains only sites out of c and the last alignment column that contains at least one site out of c. Hence, we determine the new boundary column x by computing the sum-of-pair scores for all columns under consideration. The boundaries of non-adjacent intervals are not changed. The resulting intervals correspond now to the biological history of the motifs, cf. Figure 3.2 (d).

Post-Processing and Representation

In order to gain more information about the single motifs, the next step is the calculation of a global sequence alignment of each motif represented by a column. In our implementation, we do this also with `ClustalW2` but again, each global alignment algorithm would do. After this step, the multiple alignments M_i^j determined during the calculation of consistent subsets

is an ordered list of not necessarily consecutive global alignments of conserved motifs. These motifs are supported by a maximal number of pairwise alignments and they are consistent to the linear order defined by the sequence sites.

The last step is the reunion of the clusters to our final results $\mathcal{R}$. We do this by iterating over all clusters in the order of their position relative to the m homologous input sequences $S_1, \ldots, S_m$. For each cluster $\mathcal{C}_i$, we iterate over all cliques $\mathcal{C}_i^j$ in the order of their score. Each of the corresponding alignments M_i^j has a characteristic vector $\{+, -, 0\}^m$ that describe the strands that are aligned by M_i^j, whereas 0 is used if the corresponding sequence is not part of M_i^j. If $\mathcal{R}$ already contains a multiple alignment M' with the same characteristic strand vector then the M_i^j under consideration, M_i^j and M' are consistent and we append M_i^j. Thereby, the 0 in the strand vector is neutral and match $+$, $-$ and 0. If multiple cliques of the same cluster match an alignment in $\mathcal{R}$, we only append the alignment corresponding to the highest scoring clique in order to avoid a combinatorial explosion. After checking all clusters, $\mathcal{R}$ contains all possible solutions whereas each solution is a list of phylogenetic footprints.

By default, `Tracker` present this results in from of a plain text file that includes the data in an hierarchic manner. For each solution R in $\mathcal{R}$, the file contains meta information about characteristics and scores of R together with a detailed list of all columns in R. Besides this basic output, it is also possible to obtain a graphical `HTML` output. For a detailed description of the output Chapter B (p.135) in the appendix.

3.1.3 Runtime and Memory Requirements

The worst case amount of time and space, that is needed to find phylogenetic footprints with `Tracker`, depends on the number m of sequences $S_1, \ldots, S_m$ and of the upper bound l for their length. For the calculation of the initial alignment set, we have to calculate `LASTZ` alignments for each of the $\mathcal{O}(m^2)$ sequence pairs. One alignment calculation consists of the computation of all valid seeds, the search for these seeds, and their extension. For a constant seed length, the first two steps can be performed in $\mathcal{O}(l)$ space and time while the extension asymptotically takes $\mathcal{O}(l^2)$ time and $\mathcal{O}(l^2)$ space. For all sequence pairs we get a complexity of $\mathcal{O}(m^2 l^2)$ time and $\mathcal{O}(l^2 + m^2 l)$ space for the calculation and memorizing the results.

Since the site space is restricted by $\mathcal{O}(ml)$, the number of possible resulting pairwise alignments corresponds approximately to the possible number of subsets of site space pairs and is therefore in $\mathcal{O}(2^{m^2 l^2})$. Nevertheless, meaningful scoring schemes prevent most of these possible alignments so that the number will be more likely in $\mathcal{O}(m^2 l)$. For further consideration, we denote the number of alignments by n, where the number of columns in all alignments is also restricted by l.

The optional processing of the initial alignment set takes $\mathcal{O}(nl^2)$ time and $\mathcal{O}(l^2 + nl)$ space for the calculation and the storage of the pairwise alignments and $\mathcal{O}(nl)$ time and space for the removal of not conserved and repetitive areas.

The computation of all consistent subsets needs $\mathcal{O}(n^3 + n^2ml)$ time and $\mathcal{O}(ml)$ space, see Section 2.3.4 (p.50) in the previous chapter. The division into clusters has no influence on this asymptotically complexity but it will speed up the computation of the extended scores and the subsequent assembly. For example, c clusters with approximately n/c alignments, speed up the calculation of the extended scores by factor $1/c^3$. The determination of the clusters, as well as their reunion, are neutral compared to the assembly, since they can be done in $\mathcal{O}(n^2)$ time and $\mathcal{O}(n)$ space.

The smoothing of the interval ends depends on the number of columns in the multiple alignments which is restricted by $\mathcal{O}(ml)$. Each of the columns has at most m intervals, where in the worst case each interval has an adjacent interval. In this case, we have to calculate $\mathcal{O}(m)$ multiple alignments of m sequences. Since the sequence lengths of these alignments are restricted by the constant value ε, the calculation of m multiple alignments between m sequences by a progressive alignment algorithm needs $\mathcal{O}(m^4)$ time and $\mathcal{O}(m^2)$ space. The final computation of the global alignment needs additional $\mathcal{O}(m^3 + m^2l^2)$ time and $\mathcal{O}(l^2 + m^2l)$ space for each column. For all $\mathcal{O}(ml)$ columns, we would need $\mathcal{O}(m^5l + m^3l^3)$ time and $\mathcal{O}(l^2 + m^3l^2)$ space for the calculation and storage. This worst case scenario assumes $\mathcal{O}(ml)$ columns with length $\mathcal{O}(l)$ and $\mathcal{O}(m)$ intervals per column. Nevertheless, the product of these values is restricted by the site space and hence below $\mathcal{O}(ml)$. The real complexity is therefore significantly smaller.

In summary, we need $\mathcal{O}(m^2l^2)$ time and $\mathcal{O}(l^2 + m^2l)$ space for the calculation and storage of the initial alignments, $\mathcal{O}(nl^2)$ time and $\mathcal{O}(l^2+nl)$ space for the optional processing, $\mathcal{O}(n^3+n^2ml)$ time and $\mathcal{O}(ml)$ space for the determination of the consistent alignments and $\mathcal{O}(m^5l+m^3l^3)$ time and $\mathcal{O}(m^3l^2)$ space for smoothing the intervals and the calculation of the final global alignments. All together, `Tracker` needs $\mathcal{O}(n^3+n^2ml+m^5l+m^3l^3)$ time and $\mathcal{O}(m^3l^3)$ space in the worst case. If we process the initial alignment set, the amount of time extends by $\mathcal{O}(nl^2)$ and the amount of space by $\mathcal{O}(nl)$. However, for practical uses, the effort of time and space is mainly influenced by the determination of the initial alignment set by `LASTZ`, which on average has linear running times, and the calculation of the extended scores, which on average needs $\mathcal{O}(n^2)$ time. Therefore, one can expect to end up with approximately $\mathcal{O}(m^2l+n^2)$ time.

3.2 Application to Biological Data

In order to demonstrate the functionality and performance of `Tracker`, we applied the program on two biological data sets concerning regulatory elements and the evolutionary history of Hox Clusters in birds and fish respectively.

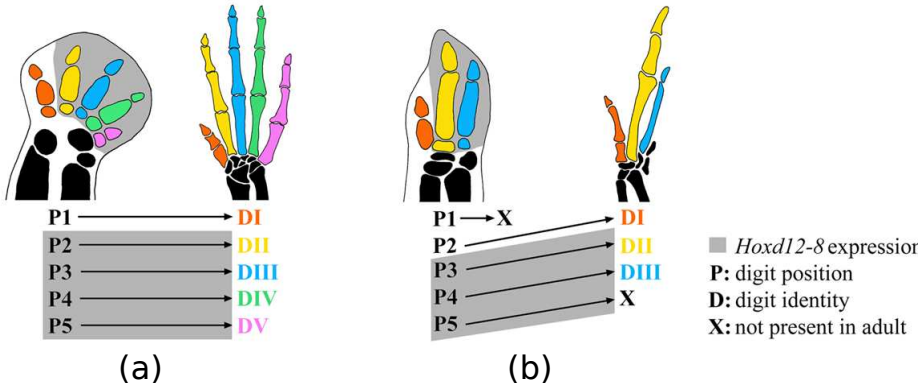
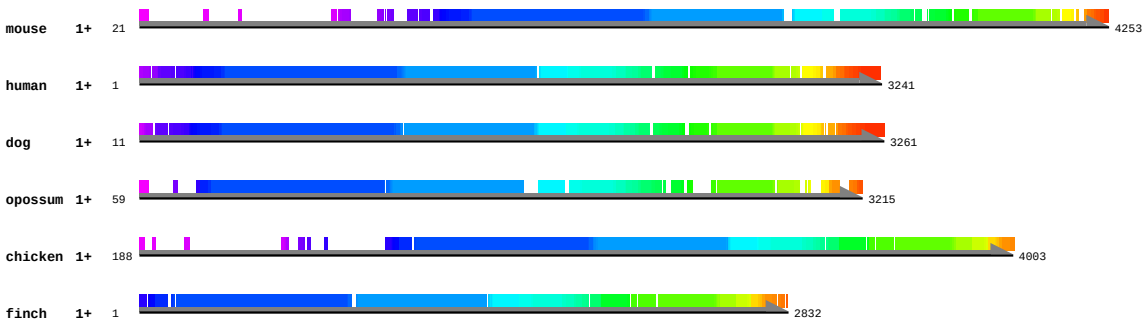


Figure 3.3: Digit position, identity and 5'HoxD expression patterns in the ancestral (a) and frame shift conditions (b). (a) In the ancestral condition (e.g. mouse), digit position corresponds to digit identity. The expression of Hoxd12 to Hoxd8 is restricted to the posterior digits (gray shading). (b) In the bird wing, there is a mismatch between digit position and digit identity. Digit positions 2 to 4 develop into digit identities I to III. Associated with the shift in digit identity relative to position, is a shift in expression of the 5'HoxD genes to positions 3 and 4.

3.2.1 Regulatory Elements of 5' HoxD Gene Expression in Birds

The study of homeotic mutants, where one body part is transformed into another, has led to advancements in our understanding of how genetics, development, and evolution interact in generating morphological diversity. One example for homeotic transformations is the evolutionary reduction in digit number in birds that has left three digits on the wing. Arguments over the identity of these three remaining digits have been ongoing for over 150 years. The debates stem from a direct disagreement between anatomical/paleontological and embryological evidence of digit identity (Wagner, 2005). On the one hand, paleontological evidence, tracking digit loss over evolutionary time, coupled with anatomical evidence of digit morphology and function identifies these digits as I, II, and III (Sereno, 1999). On the other hand, embryological evidence reveals that these digits develop in positions that typically form digits II, III, and IV in most amniotes (Burke and Feduccia, 1997). To resolve this conflict, it was hypothesized that the digits of the avian wing have undergone a homeotic transformation of identity. This *frame shift hypothesis* suggests that the developmental genetic determinants of digit identities I, II, and III have shifted and are now expressed in positions 2, 3, and 4 resulting in the transformation of digit identities (Wagner, 2005).

Recent experimental work provides strong support for the frame shift hypothesis. Examinations of posterior HoxD gene expression in the late limb development reveal a highly conserved expression pattern in the developing digits that occurs regardless of digit position (Young *et al.*, 2009). Specifically, the expression of the 5'HoxD genes Hoxd12 to Hoxd8 is restricted to the more posterior digit identities II to V, see Figure 3.3. Moreover, examinations



(a)

```
mouse:    0|+|3404-3435|32  74 (0)  TCATTACCT-TTTTGGAAAAACACTTCTCTCCC  (27) 77
human:    0|+|2251-2282|32  72 (16) TCATTACTT-TTTCAGAAAAGCACTTTTTTCCC  (0)  76
dog:      0|+|2251-2282|32  72 (16) TCATTACCT-TTTCGGAAAAGCACTTTTTTGCC  (0)  76
opossum:  0|+|2312-2342|31  74 (0)  TCATTGCCTCTTTTGAAAGAGCA--TGTTTCCC  (0)  76
chicken:                            --------------------------------
finch:                              --------------------------------
                                    ***** * * ***   ** * **  * * * **
```

(b)

```
mouse:                              -------------------------
human:                              -------------------------
dog:                                -------------------------
opossum:                            -------------------------
chicken:  0|+|3376-3400|25  82 (0)  AAAAGGAGGTAATACTTAAAGGAAA  (1) 93
finch:    0|+|2145-2169|25  82 (0)  ATAAGGAGGCAATACTTAAAGTGAA  (7) 93
                                    * ******* ***********  **
```

(c)

Figure 3.4: Alignments of the conserved sequence block B (CSB). Conserved motifs are indicated by the same color. (a) Although the sequences are highly conserved, the best scoring solution indicates small differences between mammals and birds, mainly in the green and yellow areas. For an example of clade specific motifs see (b) and (c).

of the activity of the known 5'HoxD limb cis-regulatory element *Conserved Sequence Block B* (CSB, 4kb) in primary chicken fore- and hindlimb mesenchymal cell cultures reveals that limb-specific 5'HoxD expression is associated with limb-specific regulatory activity of CSB. In order to identify the region of the CSB responsible for the limb-specific 5'HoxD expression, further experiments have to be made. One possibility are serial truncations of the CSB and the test for a limb-specific activity in primary chicken fore- and hindlimb mesenchymal cell cultures. Therefore, it is important to know the homologous regions of CSB and we demonstrate here how `Tracker` can be used for this task.

First, we extract the sequences that contain the CSB in front of the 5'HoxD genes for six species. In detail, we extract sequences for mouse, human, dog, opossum (mammals) and for chicken and zebra finch (birds). These sequences are around 40,000 bp long and they are used as input for `Tracker`. The computation of the initial alignment set provides 1,976 local pairwise alignments that result in 69 cliques. The highest scoring clique consists of 137 columns and is based on 178 alignments. The graphical representation is shown in Figure 3.4.

This best result has a significantly higher sum of column scores than the remaining 68

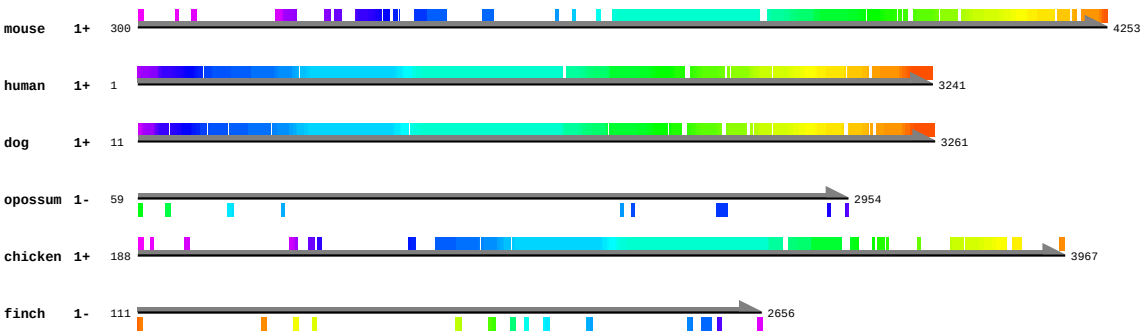


Figure 3.5: Alternative alignments of the conserved sequence block B (CSB). For the CSB dataset, the alternative solutions detect mainly small alternative motifs. In some cases, like here for the zebra finch, these motifs are on different strands. The motifs are then extended by consistent alignments already used for the determination of the best solution.

solutions. These solutions detect in this CSB data set mainly small alternative motifs that are extended with consistent alignments already used for the best solution, cf. Figure 3.5. The computation of all solutions needs on an Intel Xeon 2.13 GHz, 32 GB RAM, Fedora Linux computer less then three minutes.

3.2.2 Evolutionary Analysis of the HoxA Clusters in Ray-Finned Fish

In order to demonstrate the improvements of `Tracker` compared to the first version (Prohaska *et al.*, 2004b), we repeat the extensive evolutionary analysis of non-coding sequences in the HoxA cluster of horn shark (*Heterodontus francisci*), human (*Homo sapiens*), bichir (*Polypterus senegalus*), striped bass (*Morone saxatilis*), zebrafish (*Danio rerio*) and pufferfish (*Takifugu rubripes*) by Chiu *et al.* (2004). In this study, the authors used the newly sequenced HoxA cluster of bichier to examine if the considerable sequence evolution of derived ray-finned fishes (Teleostei) such as zebrafish and pufferfish are associated with duplications that produced additional Hox clusters.

One part of this study was the examination of the evolution of non-coding sequences in order to determine, whether the single HoxA cluster of bichir exhibits the dramatic loss of non-coding sequence conservation observed in the duplicated HoxA clusters of the ray-finned fishes (Chiu *et al.*, 2002). Therefore, the authors used the first `Tracker` version, as described in Section 2.2.1 (p.36), to identify conserved HoxA non-coding sequence tracts between Evx1 and HoxA1 in all seven species.

The analysis reveled 567 motifs distributed over all species. Several notable conserved footprint clusters were detected which indicate that bichir has one HoxA cluster that is heterogeneous in its patterns of non-coding sequence conservation and gene retention relative to the single HoxA cluster of human and shark, and the duplicated HoxAα and HoxAβ clusters

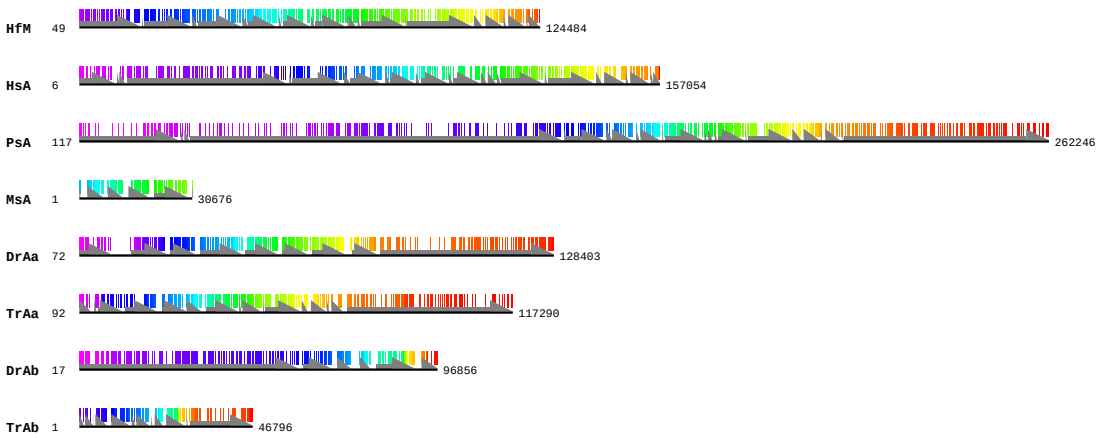


Figure 3.6: Alignments of the HoxA clusters in fishes. Fragments, presenting intergenic regions and introns, are indicated by gray arrows. Conserved sequence areas are indicated by colored lines whereas sequences with the same color are aligned together to a motif.

of zebrafish, pufferfish, and striped bass. This shows that the duplication, that produced additional HoxA clusters in derived ray-finned fishes occurred after the bichir diverged from the rest of the ray-finned fishes.

For our repeat of the study, we used the same sequences with the same homologue regions. The six sequences ranged from 30,676 bp (*Morone saxatilis*) to 262,246 bp (*Polypterus senegalus*). `Tracker`computes 424,121 pairwise local alignments that result in 2,693 cliques. The best scoring clique a significant higher score (five times the second best score) and consist of 3330 motifs, see Figure 3.6. These motifs are, in analogy to the original study, distributed over all species. An analysis of the motif pattern in the result table verifies the conserved footprint clusters in Chiu *et al.* (2004). The computation of the solutions for the HoxA data needs on an Intel Xeon 2.13 GHz, 32 GB RAM, Fedora Linux computer approximately four hours whereas the computation of the initial alignments together with the extended scores take most time.

This results indicates, that the new `Tracker` version is a significant improvement of the first version. The higher sensitivity in combination with the use of consistency in order to detect non-homologous similarities leads to a higher detection rates of homologous motifs. Furthermore, the new version is more efficient in time and memory which makes the new `Tracker` to an ideal tool for evolutionary analysis.

CHAPTER 4

Evolution of Binding Site Abundance

> The first words that are read by seekers of enlightenment in the secret, gong-banging, yeti-haunted valleys near the hub of the world, are when they look into The Life of Wen the Eternally Surprised.
>
> The first question they ask is: "Why was he eternally surprised?"
>
> And they are told: "Wen considered the nature of time and understood that the universe is, instant by instant, recreated anew. Therefore, he understood, there is in truth no past, only a memory of the past. Blink your eyes, and the world you see next did not exist when you closed them. Therefore, he said, the only appropriate state of the mind is surprise. The only appropriate state of the heart is joy. The sky you see now, you have never seen before. The perfect moment is now. Be glad of it."
>
> The first words read by the young Lu-Tze when he sought perplexity in the dark, teeming, rain-soaked city of Ankh-Morpork were: "Rooms For Rent, Very Reasonable". And he was glad of it.
>
> *Thief of Time*
> Terry Pratchett

Evolution of gene expression occurs through changes in transcriptional control mechanisms, including the modification of cis-regulatory elements (CREs) (Wilson and Odom, 2009; Li and Johnson, 2010; Schmidt *et al.*, 2010) and the evolution of novel functions in regulatory proteins such as transcription factors (TF) and co-factors (Lynch *et al.*, 2008; Wagner and Lynch, 2008). To date, most studies have focused on the evolution of CREs. Modification of CREs mostly consists of the acquisition and/or the loss of transcription factor binding sites (TFBSs) (Istrail and Davidson, 2005). Hence, investigating the molecular

evolution of CREs may reveal the timing and the kind of evolutionary changes that affect gene regulation through changes in CREs similar to the success of methods for characterizing the tempo and mode of protein evolution.

However, studying the molecular evolution of cis-regulatory sequences poses unique challenges compared to that of coding sequence evolution (Wray *et al.*, 2003; Wray, 2007). The most important obstacle to the study of non-coding sequence evolution is the lack of a "genetic code", i.e. the near impossibility to deduce the functional role of a nucleotide from the sequence context alone. Nevertheless, great progress has been made in elucidating the selective forces acting on non-coding sequences (MacArthur and Brookfield, 2004; Wong and Nielsen, 2004; Mustonen and Lässig, 2005; Wilson and Odom, 2009). There are two approaches that have been mostly used for studying the evolution of non-coding sequences. The first approach focuses on experimentally well characterized cis-regulatory elements, like those of the sea urchin gene Endo-16 (Romano and Wray, 2003; Davidson, 2006). The second uses conserved non-coding sequences as a guide to identify functionally important non-coding sequences and study their evolution (Tagle *et al.*, 1988; Blanchette and Tompa, 2002; Prohaska *et al.*, 2004b; de la Calle-Mustienes *et al.*, 2005; Otto *et al.*, 2011).

Both have serious drawbacks as general approaches to the study of cis-regulatory evolution. The first approach is certainly the one that provides results of highest quality, but is limited to a handful of well-studied genes in model organisms and their close relatives. The phylogenetic reach of this method, however, is limited because it is known that even functionally conserved enhancers can undergo rapid sequence divergence (Fisher *et al.*, 2006; Tsong *et al.*, 2006; Crocker and Erives, 2008; Wilson and Odom, 2009; Schmidt *et al.*, 2010). Thus, the functional significance of any base pair position is also rapidly changing during evolution and any molecular evolution model based on nucleotide by nucleotide substitution rate parameters is hard to parameterize except for comparisons among most closely related species.

The disadvantage of the second approach, using conserved non-coding sequences, is also caused by the inherent instability of cis-regulatory sequences. While there are some large islands of highly conserved non-coding sequences (Sandelin *et al.*, 2004a; Siepel *et al.*, 2005), there is also strong evidence suggesting that these conserved regions are only a small fraction of the functionally relevant non-coding sequences (de la Calle-Mustienes *et al.*, 2005; Wray, 2007; Schmidt *et al.*, 2010). Any study using conserved non-coding sequences as a guide is thus likely to miss much of the evolutionary action in the evolution of gene regulation. Population genetic and molecular evolutionary approaches to CRE evolution have applied a number of strategies to circumvent these problems. For example, one strategy is to first identify highly conserved non-coding sequences and then investigate the pattern and rate of nucleotide substitutions in these conserved non-coding sequences (Wagner *et al.*, 2004; Prabhakar *et al.*, 2006). Another approach is to focus on closely related species, in which the sequence divergence is minimal, and to then apply population genetic methods (Wong and

Nielsen, 2004; Haygood *et al.*, 2007), but computational methods of examining the rate and pattern of CRE evolution are still in their infancy.

Here, we propose a third approach. Non-coding sequences are not compared on a nucleotide-by-nucleotide basis following standard molecular evolution methods, but instead, focus on the abundance of specific TFBS motifs and its change during evolution. The rational for this approach is that both experimental as well as sequence evolution evidence suggests that TFBSs can undergo divergence and turnover even when the transcriptional output remains conserved (Fisher *et al.*, 2006; Tsong *et al.*, 2006). Furthermore, it is likely that there exist lineage specific differences in the retention rate of binding sites that make it desirable to estimate the rate of acquisition and decay of TFBSs from comparative sequence data. Hence the specific location of a binding site seems to be less important than that of a codon and that binding sites underlie a turnover during phylogeny, which may or may not affect function (Wilson and Odom, 2009; Schmidt *et al.*, 2010).

Changes in cis-regulatory activity in evolution may thus be reflected in a change in a quantitative character n, i.e. the number of TFBSs. In this chapter we present a stochastic, phenomenological model for TFBS abundance evolution. This model is then affirmed by simulation of sequence evolution and by real world data analyses.

4.1 The Stochastic Model

We consider the number of copies n of binding sites for a specific TF in a given genomic region of length l. We assume that the density of binding sites and other functional constraints is low enough so that the probability of the "arrival" of a new binding site by mutation is not influenced by the number of binding sites already present in that region. This assumption is considered to be plausible on the basis of preliminary studies of steroid receptor response elements in the vertebrate HoxA clusters, which vary between less than 5 to about 30 in a $100,000$ bp region of the genome (Wagner *et al.*, 2007). This constant arrival rate shall be called λ. The origination process is thus a Poisson process (Taylor and Karlin, 1998). The mutational decay of existing binding sites shall occur at a per site rate of μ. The decay process is then an exponential "death" process (Taylor and Karlin, 1998).

Stochastic models of this kind have been extensively studied in queuing theory (Medhi, 2003), where the counting variable n stands for the number of service calls in a service center. Models with both exponential arrival times and service times are called $M/M/c$, where M stands for the exponential arrival and service time distributions and c stands for the number of servers that can handle incoming calls. The transient characteristic of $M/M/c$ models has been solved by Saaty (1960) and Jackson and Henderson (1966), and summarized in Taylor and Karlin (1998), but the results are expressed in terms of the Laplace transform of the generating function, rather than the explicit probabilities, except for the case $c = 2$, which

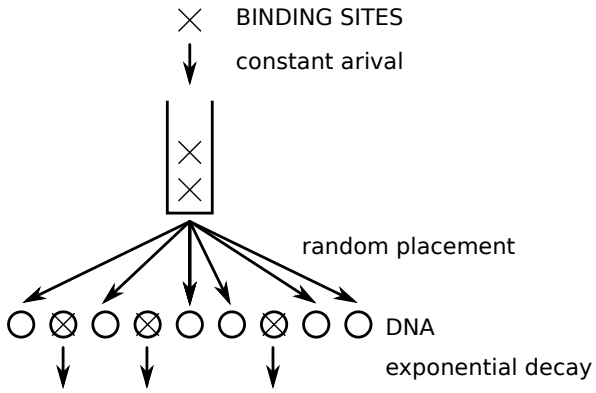


Figure 4.1: Interpretation of the stochastic model in queuing theory. Binding sites arrive with a constant rate and are randomly distributed to the serves, i.e. in our case the DNA. Binding sites on the DNA disappear with a exponential decay rate.

was calculated by Saaty (1960). For $c = \infty$ the transient probability distribution is given, without reference or derivation, in Gross and Harris (1998) for the initial condition $n_0 = 0$. Here we derive the conditional probability for the number of binding sites at time t with arbitrary initial frequency n_0.

4.1.1 A Partial Differential Equation for the Generating Function

The Kolmogorov forward equation of the system is easily derived:

$$\begin{aligned}\dot{p}_0 &= -\lambda p_0 + \mu p_1, \\ \dot{p}_n &= -\left(\lambda + \mu n\right) p_n + \lambda p_{n-1} + \left(n+1\right) \mu p_{n+1}\end{aligned}$$

where $p_n = \Pr\left(x(t) = n \mid x(t=0) = n_0\right)$, i.e. p_n is the probability that we have n binding sites at time t given that there were n_0 binding sites at $t = 0$, and $\dot{x}$ stands for the time derivative of a variable x.

The stationary solution of this system has been derived by Agner Erlang, a Danish telephone engineer who was originator of queuing theory. It is a Poisson distribution with parameter λ/μ:

$$\hat{p}_n = \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n e^{-\left(\frac{\lambda}{\mu}\right)}$$

with expectation and variance $\mathrm{E}[n] = \mathrm{Var}[n] = \lambda/\mu$.

The Kolmogorov forward equation determines the time dynamics of the probability of the number of binding sites as a function of time. The system is only a linear dynamical system but has infinitely many equations, which makes the prospect of solving them unpleasant. Instead we replace the probability distribution $p_n(t)$ with the generating function $P(z,t)$,

defined as

$$P(z,t) = \sum_{n=0}^{\infty} z^n p_n, \quad 0 \leq z \leq 1.$$

The generating function is unique to each probability distribution and the probabilities and the moments of the distribution are easily derived from the generating function. A differential equation for the time development of the generating function can be obtained from the identity:

$$\frac{\partial P(z,t)}{\partial t} = \sum_{n=0}^{\infty} z^n \dot{p}_n$$

by substitution the Kolmogorov equations into the right-hand side of the above equation. After substituting the Kolmogorov forward equations into the time derivative of $P(z,t)$ and slightly rearranging the terms one obtains:

$$\begin{aligned}\frac{\partial P(z,t)}{\partial t} = &-\lambda \sum_{n=0}^{\infty} z^n p_n + \lambda \sum_{n=1}^{\infty} z^n p_{n-1} - \mu \sum_{n=1}^{\infty} z^n n p_n \\ &+ \mu \sum_{n=1}^{\infty} z^n (n+1) p_{n+1} + \mu p_1\end{aligned}$$

The first term on the right-hand side simply is $-\lambda P(z,t)$, and the second term can easily be rearranged to yield $\lambda z P(z,t)$. The third term can be shown, after slight rearrangement, to be

$$-\mu \sum_{n=1}^{\infty} z^n n p_n = -\mu z \frac{\partial P(z,t)}{\partial z}$$

and in the fourth term the index can be redefined such that

$$\mu \sum_{n=1}^{\infty} z^n (n+1) p_{n+1} = \mu \sum_{n=2}^{\infty} z^{n-1} n p_n$$

which, with the last term μp_1 adds up to

$$\mu \sum_{n=2}^{\infty} z^{n-1} n p_n + \mu p_1 = \mu \frac{\partial P(z,t)}{\partial z}.$$

These four terms combine to give the partial differential equation (PDE) for the time-dependent generating function:

$$\frac{\partial P(z,t)}{\partial t} = \lambda(z-1)P(z,t) + \mu(1-z)\frac{\partial P(z,t)}{\partial z}.$$

This quasi-linear PDE can be solved using the method of characteristic equations. Under the initial conditions z_0, $t = 0$, and $P(z, t = 0) = z^{n_0}$ the characteristic equations of this PDE

are:

$$\frac{dz}{d\tau} = \mu(1-z), \text{ and}$$
$$\frac{dt}{d\tau} = -1$$

which have the solutions

$$t = -\tau, \text{ and } \frac{1-z_0}{1-z} = e^{\mu\tau}.$$

The third characteristic equation is

$$\frac{dP(z,t)}{dt} = \lambda(1-z)P(z,t)$$

where $z = z(\tau)$ rather than a constant and we thus need to substitute z for $z(z_0, \tau)$ from the solution of the first characteristic equation before we solve this ordinary differential equation:

$$\frac{dP(z,t)}{dt} = \lambda(1-z_0)e^{-\mu\tau}P(z,t)$$

which has the solution

$$P(z,t) = P(z_0, t=0)\exp -\frac{\lambda}{\mu}(1-z)(1-e^{-\mu t}).$$

For the initial condition $p_{n_0}(t=0) = 1$, the initial generating functions becomes

$$P(z_0, t=0) = (1-(1-z)e^{-\mu t})^{n_0}.$$

Thus, for the initial condition $p_n(t=0) = \delta_{n,n_0}$, i.e. at time $t=0$ the system has exactly n_0 binding sites, the solution of this PDE is:

$$P(z,t) = P(z_0, t=0)\exp\left(-\frac{\lambda}{\mu}(1-z)(-e^{-\mu t})\right)$$

with

$$P(z_0, t=0) = [1-(1-z)e^{-\mu t}]^{n_0}.$$

For $t=0$ this equation gives $P(z,t) = z^{n_0}$, which is the generating function of the initial distribution $p_n(t=0) = \delta_{n,n_0}$. For $t \to \infty$ the equation reduces to

$$P(z, t\to\infty) = \exp -\left(\frac{\lambda}{\mu}(1-z)\right)$$

which is the generating function of the Poisson distribution for the stationary distribution. In fact, the equation can be understood as a mixture of two distributions, on the one hand

the initial distribution with the generating function z^{n_0} and on the other hand the stationary distribution with the generating function $\exp -((\lambda/\mu)(1-z))$. Each of these generating functions is supplemented with a time-dependent term that includes a $e^{-\mu t}$, which determines the relative weight of the initial distribution and the eventual stationary distribution have at time t. The rate of approach to the stationary distribution only depends on the decay rate μ, which also determines the rate at which binding sites initially present are replaced by new arrivals, and thus the rate at which history is erased by binding site turnover.

From the PDE of $P(z,t)$ it is easy to derive the dynamical equation for the expectation, by taking advantage of the identity

$$\mathrm{E}[n] = \left.\frac{\partial P(z,t)}{\partial z}\right|_{z=1}$$

one obtains

$$\frac{d\,\mathrm{E}[n]}{dt} = \lambda - \mu\,\mathrm{E}[n]$$

which has the solution

$$\mathrm{E}[n(t)] = \frac{\lambda}{\mu}(1-e^{-\mu t}) + \mathrm{E}[n(t=0)]e^{-\mu t}. \tag{4.1}$$

This equation again has the same structure as that for the generating function, namely a term for the influence of the initial mean $\mathrm{E}[n(t=0)]$ which is decreasing with time according to a negative exponential function, and the expectation of the stationary distribution λ/μ whose influence is increasing with t according to the function $(1-e^{-\mu t})$.

The equation for the time-dependent variance is also directly derivable:

$$\mathrm{Var}[n(t)] = \frac{\lambda}{\mu}(1-e^{-\mu t}) + n_0 e^{-\mu t}(1-e^{-\mu t}).$$

4.1.2 The Conditional Probability Distribution

The generating function can also be used to calculate the complete transient probability distribution using the identity

$$p_n(t) = \left.\frac{1}{n!}\frac{d^n}{dz^n}P(z,t)\right|_{z=0}.$$

The generating function can be written as

$$P(z,t) = e^{-a}B(z)e^{za}$$

with $a = (\lambda/\mu)(1 - e^{-\mu t})$ and $B(z) = [1 - (1 - z)e^{-\mu t}]^{n_0}$. Applying Leibniz's rule for the nth derivative of F(x) = u(x)v(x) gives

$$\frac{d^n}{dx^n}F(x) = \sum_{k=0}^{n} \binom{n}{k} \frac{d^k u}{dx^k} \frac{d^{n-k} v}{dx^{n-k}}$$

In order to obtain an explicit expression for the time-dependent probability distribution we need to calculate the nth derivative of $P(z,t) = e^{-a}B(z)e^{za}$ for $z \to 0$. Using Leibniz's rule we get

$$\frac{d^n P(z,t)}{dz^n} = e^{-a} \sum_{k=0}^{n} \binom{n}{k} \frac{d^k B(z)}{dz^k} \frac{d^{n-k} e^{za}}{dz^{n-k}} \ .$$

We have $d^{n-k}e^{za}/dz^{n-k} = a^{n-k}e^{za}$, which reduces to d^{n-k} for $z = 0$. B(z) can be written as

$$B(z) = \sum_{l=0}^{n_0} \binom{n_0}{l} \alpha^l (z\beta)^{n_0 - l}$$

with $\alpha = (1 - e^{-\mu t})$ and $\beta = e^{-\mu t}$. This expression then can be derived and for $z = 0$ leads to

$$\left.\frac{d^k B(z)}{dz^k}\right|_{z=0} = \begin{cases} \binom{n_0}{k} k! \alpha^{n_0 - k} \beta^k & \text{for } k \le n_0, \\ 0 & \text{for } k > n_0. \end{cases}$$

which then combines with the Leibniz's formula to give

$$\left.\frac{d^n P(z,t)}{dz^n}\right|_{z=0} = \sum_{k=0}^{\min(n,n_0)} k! \binom{n}{k} \binom{n_0}{k} \left(\frac{\lambda}{\mu}\right)^{n-k} \times (1 - e^{-\mu t})^{n+n_0-2k} (e^{-\mu t})^k$$

All together we obtain the conditional probability distribution

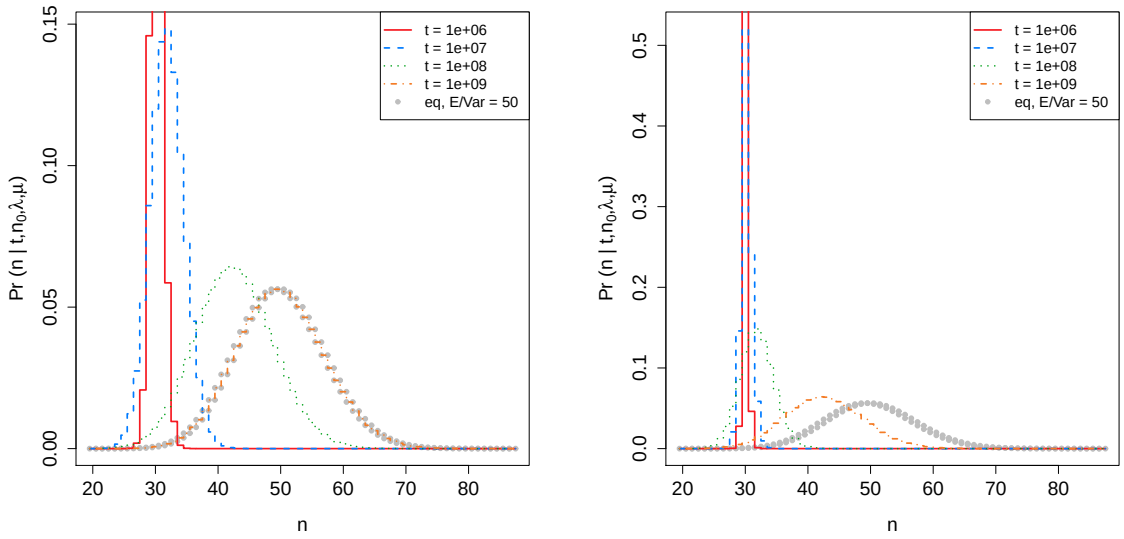
$$p_n(t) = \frac{1}{n!} e^{-\left(\frac{\lambda}{\mu}\right)(1-e^{-\mu t})} \sum_{k=0}^{\min(n_0,n)} k! \binom{n}{k} \binom{n_0}{k} \left(\frac{\lambda}{\mu}\right)^{n-k} \times (e^{-\mu t})^k (1 - e^{-\mu t})^{n+n_0-2k} \tag{4.2}$$

which, for $t \to \infty$, gives the stationary Poisson distribution

$$\hat{p}_n = \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n e^{-\left(\frac{\lambda}{\mu}\right)}, \tag{4.3}$$

for $t \to 0$ gives

$$p_n = \begin{cases} 1 & \text{if } n = n_0, \\ 0 & \text{else.} \end{cases}$$



(a) (b)

Figure 4.2: Characteristics of the Conditional Probability Distribution. The diagrams shows the probability for binding site numbers for different times, in detail $t = 10^6$ (red), $t = 10^7$ (blue), $t = 10^8$ (green), $t = 10^9$ (orange) and in the equilibrium state (gray). The binding site number n_0 for $t = 0$ is 30. In the left diagram (a) we use parameters $\lambda = 5 \times 10^{-7}$ and $\mu = 1 \times 10^{-8}$ while in the the right diagram (b) we use $\lambda = 5 \times 10^{-8}$ and $\mu = 1 \times 10^{-9}$. In both cases, the ratio λ/μ is 50. For small time values, the distribution is mainly determined by n_0. With ascending times, the distribution converts to the Poisson distribution that depends only on the rate λ/μ. The rates in (b) are the tenth part of (a). Therefore, the same distribution is reached after the tenfold time.

and for $n_0 = 0$ reduces to

$$p_n(t) = \frac{1}{n!} e^{-\left(\frac{\lambda}{\mu}\right)(1-e^{-\mu t})} \left(\frac{\lambda}{\mu}\right)^n (1 - e^{-\mu t})^n$$

(see Gross and Harris, 1998, page 101).

The explicit expression for the conditional probability distribution allows us the analytical determination of the probability to have n binging sites in a genomic region at time t. This value is determined in dependence of the constant rate of binding site origination λ, the constant per site decay rate μ, and the number n_0 of binding sites at $t = 0$. Thereby, $\mu \sim 1/t$ and $\mu \sim \lambda$ since t is always linked to μ and λ is always linked to $1/\mu$. For a graphical representation of the characteristics of Equation 4.2 see Figure 4.2.

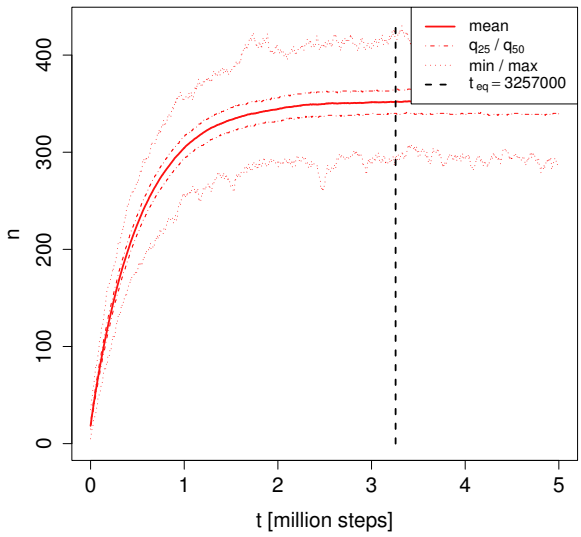


Figure 4.3: Spreading of simulated binding site numbers over time. The figure displays the mean binding site number (solid red line) and the distribution of the data by the low and high quartile (discontinuous red line) and the min and max values (dotted red line). The number of binding sites increases linearly at the beginning and the rate decreases until finally an equilibrium state (discontinuous black line) is reached at $t_{eq} = 3,257,000$.

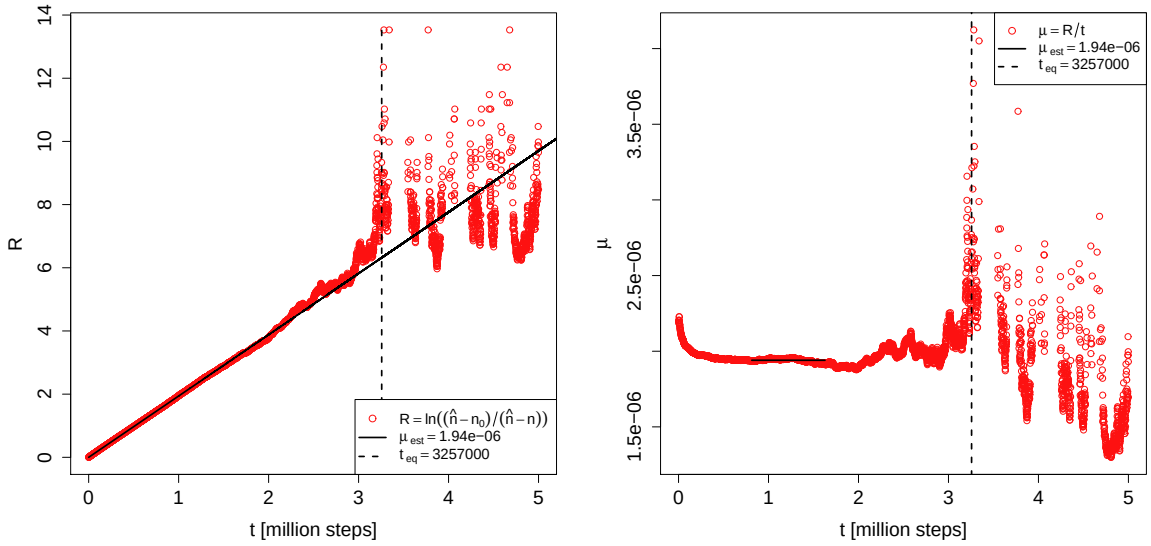
4.2 Validation of the Phenomenological Equations

In order to confirm our model, we compare distributions of binding site numbers deduced by simulation of sequence evolution with the analytical results of the model. Subsequently, we demonstrate that the assumptions of the model apply to biological data.

4.2.1 Simulation of Sequence Evolution

For the determination of reference binding site distributions we created a simulation model of sequence evolution. Starting with a single random sequence of length l, the sequence is composed of equally frequent nucleotides. The simulation is done over t_{max} time steps. In each time step, mutations occur with a mutation rate m. Thus, the number of mutations in a time step is a Poisson variable with the expectation ml.

For each mutation, we determine a position in the sequence by a uniform random process and perform a mutation whereby each of the three possible new nucleotides has the same probability to replace the current one. After all mutations within a time step are done, we check if these mutations will be fixed. In order to do so, we distinguish between two different events. In the first case, more new binding sites are created by the mutations than are destroyed or just as many binding sites are created as are destroyed, respectively. These mutations are fixed by a fixation rate ϕ^+. In the second case, if more binding sites are destroyed by mutations than new ones created, we use a fixation rate ϕ^-. We use $\phi^+ > \phi^-$

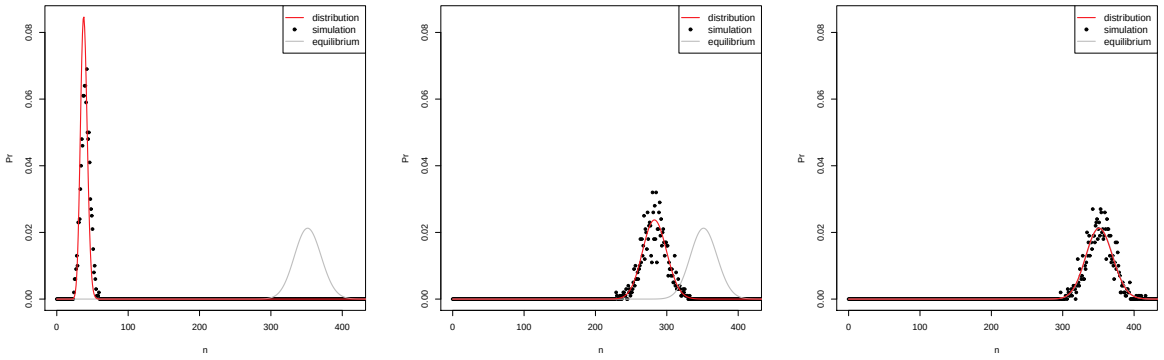


(a) (b)

Figure 4.4: Estimating the effective decay rate μ of binding sites. The left diagram (a) shows the linear relationship (solid black line) between time t and the regression variable $R = \ln((\hat{n} - n_0)/(\hat{n} - n))$ in the transient section of the development of the mean binding site number. This relationships ends before reaching the equilibrium state (discontinuous black line). The right diagram (b) shows the resulting values for μ as a function of time. The solid black line indicates the area that is used for the estimation of the value of μ.

since we assume that binding sites are maintained by stabilizing selection, and thus the loss of binding sites is associated with a selective penalty which decreases the fixation probability.

For our simulations we set the sequence length to 100,000 nt, which is approximately the size of the HoxA cluster, which we investigate later. We scan for binding site patterns with a shape of A^l, whereby A stands for the nucleotide Adenine and l for the length of the binding site. This choice is arbitrary and does not affect the results in this simulation because any specific sequence is equally probable with equal nucleotide frequencies. We use a binding site length of $l = 6$ which is typical for small TFBSs. Mutations occur in a time step with a mutation rate $m = 10^{-5}$ per nucleotide and are fixed by a fixation rate $\phi^+ = 1$ for neutral mutations, leading to a per nucleotide evolution rate equal to the mutation rate. We use $\phi^- = 0.05$ for deleterious mutations. Mutations which do not affect binding sites are also fixed at the neutral rate 1. To reach the equilibrium states for these parameters, we simulate over $t_{max} = 5,000,000$ time steps and we record the number of binding sites on the coding strand every 10,000 steps. In order to minimize the influence of extreme random effects we perform 1000 simulations and calculate the average number of binding sites for each time step. The number of binding sites increases linearly at the beginning. Later, the increase slows down until finally an equilibrium state is reached, see Figure 4.3.



(a) (b) (c)

Figure 4.5: Comparison of the transient ensemble distribution of binding sites resulting from simulations with the analytical prediction. Note the close relationship between the predicted (red line) and the simulated frequency distribution (black points). The chosen times correspond to 0.01 (a), 0.25 (b) and 1.00 (c) of the equilibrium time, see Figure 4.3. The simulation closely resembles the model. With growing time, both distributions converge to the stationary Poisson distribution (gray line).

In order to validate our phenomenological model, we compared the probability distribution $p_n(t)$ predicted by the phenomenological model with the frequency distribution of the binding site number n. For calculating the probabilities of the binding site number predicted by our model, we first have to determine the arrival rate of new binding sites λ and the decay rate μ for the sequence evolution model. We determine these values from the time development of the mean binding site number, given by Equation 4.1 (p.83):

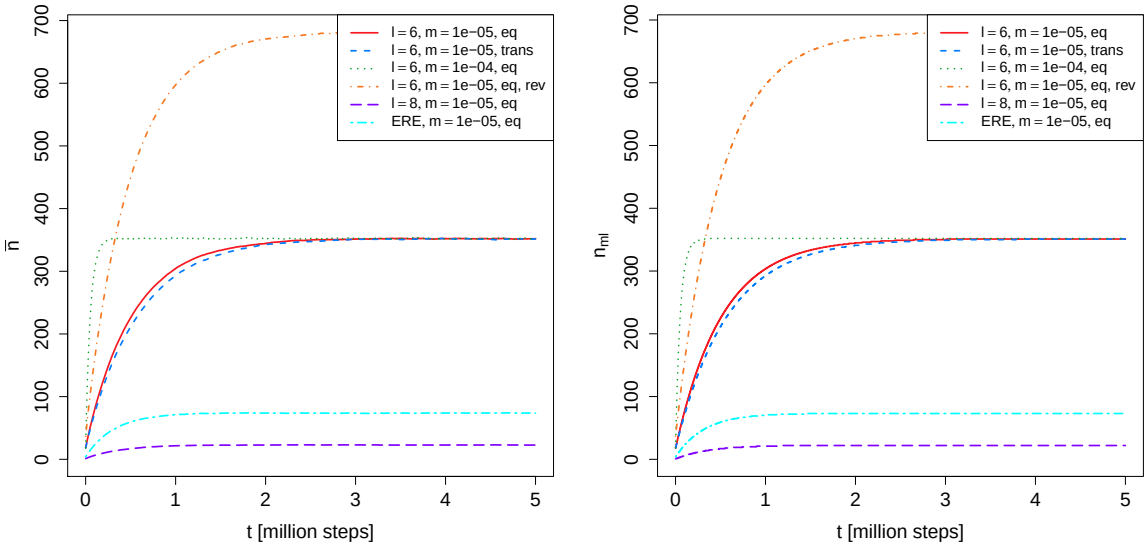
$$\mathrm{E}[n(t)] = \frac{\lambda}{\mu}(1 - e^{-\mu t}) + \mathrm{E}[n(t=0)]e^{-\mu t}.$$

Note that $\lambda/\mu = \lim_{t\to\infty} E[n(t)] = \hat{n}$ is the equilibrium expectation of the number of binding sites and $E[n(t=0)]$ is n_0 according to our initial conditions. If we solve this equation for t we can determine μ as a regression coefficient. Denoting the mean binding site number $\mathrm{E}[n(t)]$ with n we get

$$t = \mu^{-1} \ln\left(\frac{\hat{n} - n_0}{\hat{n} - n}\right).$$

In the transient section of the function $\mathrm{E}[n(t)]$ exists a linear relationship between time t and the regression variable $R = \ln((\hat{n}-n_0)/(\hat{n}-n))$ where the slope is μ^{-1}, see Figure 4.4 (a). The resulting constant value of μ in the transient section is taken for the further analysis, see Figure 4.4 (b). The number of binding sites at the equilibrium state $\hat{n}$ is an estimate of λ/μ, and thus we can also estimate the arrival rate as $\lambda = \hat{n}\mu$.

This allows us to compare the transient probability distribution predicted by the analytical model with the simulation data. The distribution of the simulated data is determined by



(a) (b)

Figure 4.6: Effect of different parameters on the results. The left diagram (a) shows the mean binding site number of the simulation for different parameters. The right diagram (b) shows the max like binding site number determined by our phenomenological model. Both diagrams match perfectly. The used parameters are given in the label, where l is the motif length, ERE stands for the estrogen response element, m for the mutation rate, eq for equally distributed mutations, $trans$ for a transition/transversion rate ratio of 2/1 and rev for motif count on both strands.

the relative frequency of binding site numbers at a given time. As shown in Figure 4.5, the simulation closely resembles the predicted probability distributions.

We repeat the simulations with different values for l and m and use a more realistic sequence evolution model assuming a transition/transversion rate ratio of 2/1. Furthermore, we prove the independence of our model from the recognition method of TFBSs by using a position weight matrix for the determination of binding sites for the estrogen receptor. The estrogen receptor is a ligand-activated enhancer protein that is a member of the steroid/nuclear receptor superfamily. It binds to specific DNA sequences called estrogen response elements (EREs) with high affinity and transactivates gene expression in response to estradiol. The PWM for the ERE was created by 41 natural vertebrate ERE motifs (Klinge, 2001). We use the `MATCH` algorithm (Kel *et al.*, 2003) with a threshold of 0.85 which corresponds to two mismatches in the perfect pattern `AGGTCANNNTGACCT` to count the EREs in the sequences. The reliability of this counting method was cross-validated against the `Dragon ERE Finder 2`(Bajic *et al.*, 2003), which uses a hidden Markov model to identify EREs. Our approach slightly undercounts the ERE numbers but the numbers obtained are highly correlated and we use the PWM method for this exploratory analysis.

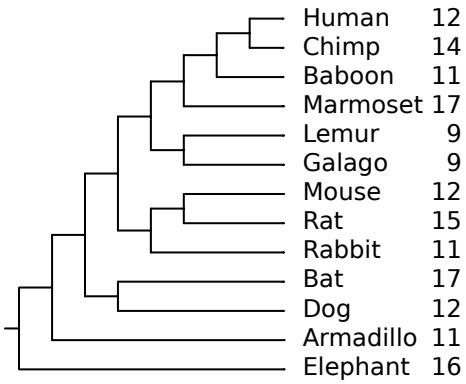


Figure 4.7: Evolution of estrogen response element (ERE) number in the HoxA cluster among mammals, represented as a unscaled phylogenetic tree. Note the variation of the number of EREs in closely related species.

For all tested parameters, we obtained the same close fit to the phenomenological model, see Figure 4.6. The detailed graphical analyses can be found in the appendix, see Section A.1 (p.127). In general we observe an higher increase of the binding site number for shorter binding sites lengths l since they are more likely to arise. High mutation rates m enhance this effect. Short binding sites also reach higher equilibrium levels, see Figure A.1 (p.128). The decay rate μ grows with higher mutation rates and longer motif lengths since both enhance the probability that binding sites are destroyed, see Figure A.3 (p.130). The arrival rate λ grows with high mutation rates, while longer motifs result in smaller rates for λ. In all cases, the simulation closely resembles the predicted probability distributions, see Figure A.4 (p.131) and Figure A.4 (p.132).

4.2.2 Application to Biological Data

In order to see whether the assumptions of the model developed here apply to real genomic sequences, we investigate the variation in the number of EREs in the HoxA cluster sequences of mammals. HoxA clusters are a phylogenetically stable genomic locus within mammals and even sharks and lobe-finned fishes (Chiu *et al.*, 2002) which should allow the identification of homologous binding sites. In detail, we count the EREs in human, chimpanzee, baboon, marmoset, lemur, galago, rabbit, rat, mouse, dog, bat, armadillo and elephant, see also Table A.1 (p.133) in the appendix. The numbers are again determined by the `MATCH` algorithm (Kel *et al.*, 2003), using the PWM created by 41 natural vertebrate ERE motifs (Klinge, 2001) and a threshold of 0.85, see Section 4.2.1 (p.89). We search the whole HoxA cluster sequences including $5,000$ bp at the $5'$ end of *HoxA13* and $5,000$ bp at the $3'$ end of *HoxA1*. These sequences are on average $115,600$ bp long. The results are summarized in Figure 4.7.

In mammals, the number of EREs in the HoxA cluster varies between 9, as found in the galago and lemur, and 17, as found in the marmoset and bat. The overall mean for all mam-

	Human	Chimp	Baboon	Marmoset	Lemur	Galago
Human	**1**	6.6	30.5	42.9	77.5	77.5
Chimp	11	**1**	30.5	42.9	77.5	77.5
Baboon	5	6	**3**	42.9	77.5	77.5
Marmoset	2	3	1	**13**	77.5	77.5
Lemur	1	0	0	3	**5**	57.1
Galago	2	3	2	1	1	**4**

Table 4.1: The number of autapomorphic uniquely derived EREs in the respective species is given in the diagonal (bold). The lower triangular part of the table gives the number of shared homologous EREs while the upper half of the table contains the time in million years since separation of the corresponding species (Steiper and Young, 2006).

mals is 12.8. The stationary distribution of the $M/M/\infty$ process would be a Poisson process with variance equal to the mean. In our data the variance is about 60% of the mean which is expected given that the individual observations are not stochastically independent, because of the phylogenetic structure among the species, and thus the variance is expected to be less than that of the Poisson distribution of stochastically independent events. The distribution appears to be homogeneous among subgroups of species. For instance, the primates have a mean of 12.0 and a variance of 9.6 while the rest of the mammals have a mean of 13.4 and a variance of 6.3, although a formal test of homogeneity is difficult due to the phylogenetic structure in our data. Overall the variation in ERE number is consistent with a stationary process with an expectation between 12 and 13 ERE per HoxA cluster.

In order to test whether there was a process of binding site turnover we determine the number of homologous binding sites among pairs of species in primates and compare these numbers to the time since lineage separation (Steiper and Young, 2006). We found that only among the primate species we could reliably identify homologous sites because of the high rate of intergenic sequence divergence. We thus proceed analyzing data from the six primates which yield 15 pairwise comparisons. The results are summarized in Table 4.1.

If the binding sites are gained and lost by a process modeled due to $M/M/\infty$, the expected number of shared homologous sites, $\mathrm{E}[n_\cap]$, should decrease exponentially with time since lineage separation (Taylor and Karlin, 1998):

$$\mathrm{E}[n_\cap] = ne^{\mu t}. \tag{4.4}$$

Together with the number of binding sites in the species, which adds six data points at $t = 0$, we perform a linear regression of $\ln(n_\cap)$ over time since lineage splitting, see Figure 4.8. For the regression we omit the two data points with 0 shared ERE between chimp and lemur as well as between baboon and lemur. Based on this data, we get for t in million years (Myr) a regression equation of

$$\ln n_\cap = -0.0259t + 2.34.$$

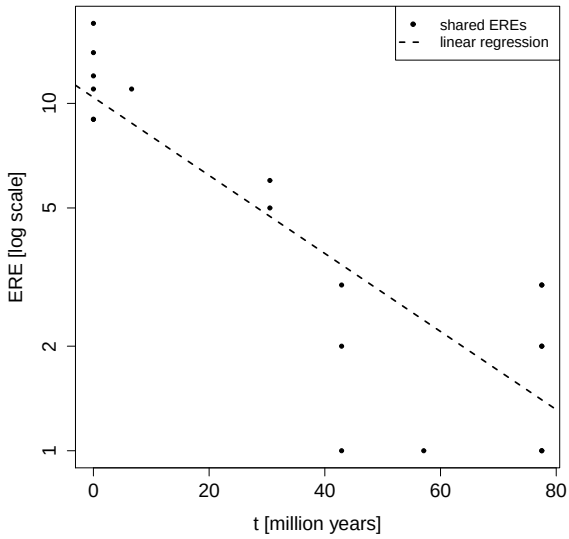


Figure 4.8: Dependency of the number of shared EREs (log scale) among primate species as a function of time (million years) since lineage separation according to the divergence time estimates (Steiper and Young, 2006). Note the decline of shared EREs with time since separation. The slope of decline suggests a half-life time of ERE in the primate HoxA clusters of 27 million years.

Interpreting this equation on the basis of the theoretical model

$$\ln(\mathrm{E}[n_\cap]) = -\mu t + \ln n$$

leads to the estimate for the decay rate of binding sites of $\mu \approx 0.0259/(2 \times 10^6) = 1.3 \times 10^{-8}$ binding sites per year or a half-life time for the ERE sites in the mammalian HoxA clusters of about 27 Myr. Thereby, the term $1/(2 \times 10^6)$ is used since the regression is performed for 10^6 years and the distance of two species is the sum of the times since lineage separation. Note that the true decay rate is expected to be higher since we omitted two data points with 0 shared ERE binding sites.

A formal test of the fit between the data and the exponential decay model is hard to perform, because of issues of stochastic dependency among the data points as well as the fact that the exponential death process has a time-dependent variance. Instead we use the fact that the transient probability distribution of the exponential death process is known to be a binomial distribution with a survival probability at time t of $e^{-\mu t}$ (Taylor and Karlin, 1998). Therefore, we have:

$$\Pr(n_\cap \mid t) = \binom{n}{n_\cap} (e^{-\mu t})^{n_\cap} (1 - e^{-\mu t})^{n - n_\cap}.$$

We can calculate whether any data points are out of bounds in our regression. In particular the spread of data points at $t = 77.5$ Myr between zero and three could either be due to

chance or indicate a class of binding sites that are more stable than the rest of them. Using the probability distribution above we calculate that the probability of having more than two shared homologous sites after 77.5 Myr half-life time of 27 Myr is 0.208 for $n = 12$. Hence, the observation that there are three homologous sites shared between chimp and galago HoxA clusters is not unexpected based on the exponential decay model, see Figure 4.8.

Another suspicious data point is that the marmoset and the baboon only share a single homologous ERE after only 42.9 Myr of separation. Assuming the exponential decay model, the probability of sharing one or zero ERE is 0.059 for $n = 12$, which is marginal in terms of statistical significance. Overall the data is consistent with a model that assumes a constant turnover of ERE with a half-life time of 27 Myr.

We can also use this data to estimate the parameters of our model. Assuming that the mean value of sites estimates the expected value of the stationary $M/M/\infty$ process, one obtains $\hat{n} = \lambda/\mu \approx 12$. This inference is supported by the fact that data partition by phylogenetic group does not lead to clearly different average numbers of EREs. The regression analysis of shared EREs among primates already gave us an estimate of $\mu \approx 1.3 \times 10^{-8}$ binding sites per year. Combining this result with the $\hat{n} = \lambda/\mu \approx 12$ leads to an estimate of the production rate of new binding sites of $\lambda \approx 1.6 \times 10^{-7}$ binding sites per year.

In order to test whether these numbers of sites are higher than expected for random sequences, we produce 1000 random DNA sequences of 104,350 bp, which is the length of our mammalian HoxA clusters minus the non-synonymous sites and the sites that are conserved between human and shark (Prohaska *et al.*, 2004b), and search for the presence of ERE elements. We observe that on average the probability of finding 12 or more ERE in a random DNA sequence with the base composition typical for mammals is about 0.074. This number alone is not sufficient to conclude that any of the mammalian HoxA clusters individually has more EREs than expected by chance. However, the fact that many mammalian lineages have approximately 12 ERE allows us to conclude that the number of ERE in the mammalian HoxA clusters cannot be explained by chance alone.

Above we showed that the half-life of a ERE in the mammalian HoxA cluster is about 27 Myr, and hence any lineage that is separated by more than approximately 80 Myr is essentially a stochastically independent observation. Even if there were only two such lineages the probability for both of them to have about 12 EREs is about 0.0055. Hence, we conclude that the number of EREs found in mammalian HoxA clusters is higher than expected by chance and probably maintained at that level by natural selection. This is important since it shows that the turnover with a half-life time of 27 Myr affects functionally important EREs, i.e. those which are affected by stabilizing selection. Hence, the turnover of TFBSs seems to be the predominant mode of evolution.

CHAPTER 5

Measuring Binding Site Turnover

> Few religions are definite about the size of Heaven, but on the planet Earth the Book of Revelation (ch. XXI, v.16) gives it as a cube 12,000 furlongs on a side. This is somewhat less than 500,000,000,000,000,000,000 cubic feet. Even allowing that the Heavenly Host and other essential services take up at least two thirds of this space, this leaves about one million cubic feet of space for each human occupant — assuming that every creature that could be called 'human' is allowed in, and the the human race eventually totals a thousand times the numbers of humans alive up until now. This is such a generous amount of space that it suggests that room has also been provided for some alien races or – a happy thought – that pets are allowed.
>
> *The Last Hero*
> Terry Pratchett

Acquisition of specific binding sites and the subsequent conservation was for a long time thought to be the predominant biological mode of binding site evolution. By the current state of knowledge, this is unlikely. The turnover of transcription factor binding sites (TFBSs) is well established (Fisher *et al.*, 2006; Tsong *et al.*, 2006; Crocker and Erives, 2008). Therefore, any molecular evolution approach for the discovery of novel regulatory architecture in developmental evolution has to take the possibility into account that derived cis-regulatory elements are not characterized by a conserved sequence.

In the previous chapter, we explicitly derived the transient conditional probabilities for changes in binding site number and proposed a stochastic model for TFBS evolution. In this approach, it is assumed that the functionally important measure is likely to be binding site

number rather than their location or precise identity. This model further assumes that TFBSs in the genomic region of interest have a constant rate of origination λ and decay exponentially with a relative rate μ.

We think that biologically significant changes in the upstream regulation of a locus will be reflected in changes in the origination and decay rates of specific binding sites in certain clades rather than the acquisition of a specific set of binding sites in the derived clade. In fact the purpose of the development of the phenomenological model of binding site number dynamics is to make predictions that can be tested with rigorous statistical methods to detect deviations from a constant rate of turnover. These deviations are a proxy for differences in the selective pressures acting on TFBSs in different clades. Therefore, the present model is meant to set the stage for a sequence analysis tool, similar to the neutral sequence evolution model used to detect natural selection in coding regions (Graur and Li, 2000).

We use the model to search for two kinds of deviations from the model predictions. One is heterogeneity in the rate of origination and/or decay between different lineages or clades. That could be done by comparing the likelihood of the observed distribution of binding site numbers on the tips of a phylogenetic tree assuming constant rates of origination and decay with a model which allows for different parameters in different parts of the phylogenetic tree. A significant difference in likelihood would constitute evidence for a difference in the selective constraints on binding sites in different groups of organisms.

The second possible application of the model is to provide evidence for binding sites with different turnover rates. For instance it is likely that a non-coding region contains binding sites with strong selective constraints and others with weaker constraints (Dermitzakis and Clark, 2002). This could be detected by testing for deviations from the exponential loss of homologous binding sites as a function of time since lineage separation.

In either case it will be necessary to use the stochastic model to calculate the likelihoods of data sets on phylogenies. The development of this maximum likelihood approach and the validation with simulated and biological data is subject of this chapter.

5.1 The `Creto` Algorithm

In order to measure the turnover rates of a TFBS on phylogenies we developed the program `Creto`, short for Cis-Regulatory Element Turn-Over. Details about the exact input and output syntax, possible parameters and the availability are given in the appendix, Chapter C (p.143).

The algorithm takes as input a phylogenetic tree T with branch lengths t and the binding site numbers n for the terminal taxons represented by the leaves of T. In the simplest form, the algorithm estimates the decay rate μ, and the origination rate λ, for these binding sites by maximizing the likelihood of the observed binding site numbers on the tree. The likelihood is calculated based on the time-dependent conditional probability distribution, derived in the

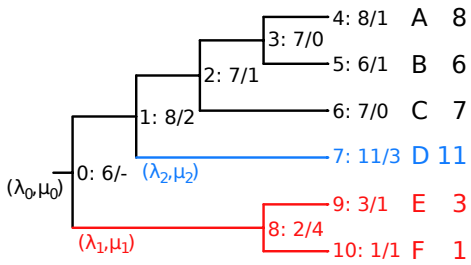


Figure 5.1: Example of binding site number evolution for multiple turnover parameters. The phylogenetic tree has six species (A to F) with 1 to 11 binding sites. Each node is labeled with its unique id followed by the mean binding site number of the corresponding subtree and the difference of that number to that of the ancestor node. While the subtree defined by the root node 0 always has the parameters (λ_0, μ_0) (black), the alternative parameters are per default assigned to the subtree that has the highest difference of the mean binding site number to the mean binding site number of its ancestor. In this example showing three parameter pairs, node 8 with a difference of 4 has parameters (λ_1, μ_1) (red) while node 7 with difference 3 has the pair (λ_2, μ_2) (blue). Besides this automatic assignment, it is also possible to select the subtrees manually. Note that the rates are valid from the ancestor of the subtree on.

last chapter. To facilitate the detection of statistically significant differences in binding site dynamics, it is also possible to estimate the parameters for the case that a subtree or several subtrees have alternative binding site turnover rates, see Figure 5.1. Note that the parameter pair for a subtree is applied to a clade and the stem lineage of the clade.

5.1.1 Likelihood Calculation

The likelihood L of a given phylogenetic tree T is based on the conditional likelihoods $\mathrm{L}_i(n)$ of the nodes i inside T. Thereby, $\mathrm{L}_i(n)$ is the likelihood of all evolutionary histories conditional on the assumption that at node i exactly n binding sites are present.

Based on this formalization, the likelihood of the whole tree T is the weighted average of the conditional likelihoods of the root node r over all possible binding site numbers with the prior probability $\pi(n)$ (Felsenstein, 2003):

$$\mathrm{L} = \sum_n \pi(n)\, \mathrm{L}_r(n).$$

For the prior probability, we take the equilibrium distribution for the TFBS model, which is the Poisson distribution. We parameterize the Poisson distribution with the mean binding site number $\bar{n}$ averaged over all species. Because the number of all possible binding site numbers is potentially infinite, we have to restrict n to a finite interval $n_{\min} \leq n \leq n_{\max}$ for

computational reasons.

$$\mathrm{L} = \sum_{n_{\min} \le n \le n_{\max}} \frac{\bar{n}^n e^{-\bar{n}}}{n!} \mathrm{L}_r(n).$$

For determining $n_{\min}$ and $n_{\max}$, we use a bound value (default is 10^{-6}) and calculate the smallest and largest n so that the probability of the Poisson distribution from 0 to $n_{\min} - 1$ is smaller than half the value for bound, and the probability of $n > n_{\max}$ is also smaller than half the value for bound. Since the Poisson distribution is more dispersed than any transient probability distribution, see Figure 4.2 (p.85), the bound value is a conservative estimate of the error in the likelihood calculation. If the smallest or the highest binding site number is outside this range, we adapt $n_{\min}$ or $n_{\max}$, respectively.

The calculation of the conditional likelihoods $\mathrm{L}_i(n)$ depends on the kind of the node i. If the node is a leaf, i.e. representing a terminal taxon, the likelihood is calculated as

$$\mathrm{L}_i(n) = \begin{cases} 1 & \text{if leaf } i \text{ has } n \text{ binding sites,} \\ 0 & \text{else.} \end{cases}$$

If the node i is an internal node, then the likelihood is calculated as

$$\mathrm{L}_i(n) = \prod_{j \in \mathrm{Descendants}(i)} \left[\sum_{m=n_{\min}}^{n_{\max}} \Pr(m \mid n, t_j)\, \mathrm{L}_j(m) \right].$$

The likelihood of the node i having exactly n binding sites is proportional to the product of the probabilities of all events in all the lineages that emanate from node i. In each lineage with descendant j, we sum the transient likelihood over all possible binding site numbers m of j. Given the length t_j of the branch leading from i to the descendant node j, the transient likelihood is the likelihood $L_j(m)$ of node j having m binding sites multiplied with the probability, given by Equation 4.2 (p.84), to have n binding sites after t_j time starting with m binding sites:

$$\begin{aligned} \Pr(n \mid m, t) = & \frac{1}{n!} e^{-(\lambda/\mu)(1-e^{-\mu t})} \sum_{k=0}^{\min(m,n)} k! \binom{n}{k} \binom{m}{k} \left(\frac{\lambda}{\mu}\right)^{n-k} \\ & \times (e^{-\mu t})^k (1 - e^{-\mu t})^{n+m-2k}. \end{aligned}$$

If the branch length t is long, i.e. if $t \gg 1/\mu$, then the conditional probability converges to the equilibrium distribution

$$\Pr(n) = \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n e^{-(\lambda/\mu)}.$$

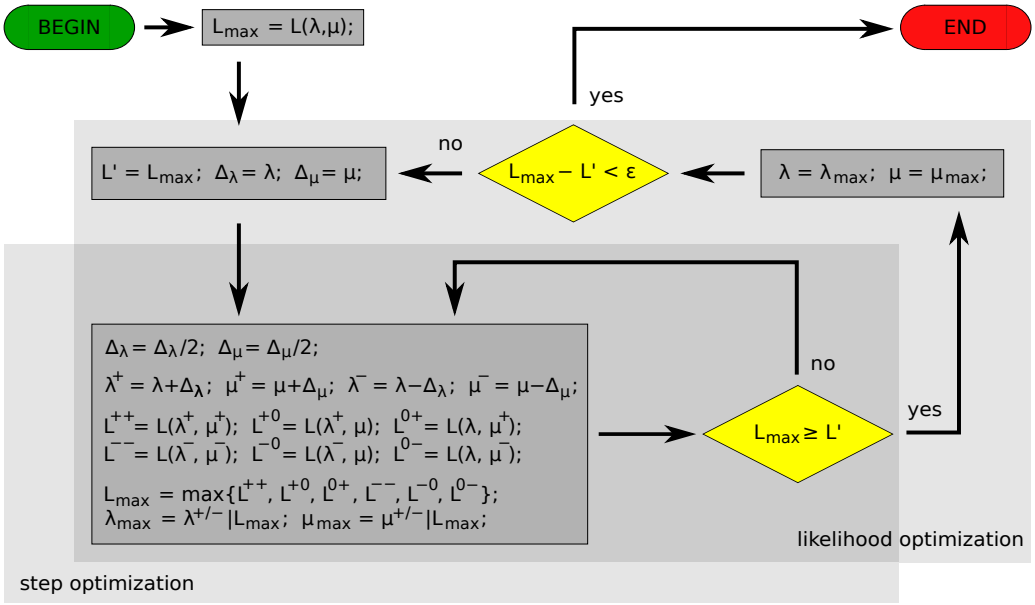


Figure 5.2: Workflow of the maximum likelihood algorithm. The likelihood optimization is performed until the likelihood improvement is smaller the abort criterion ε. In each improvement cycle, a step optimization is performed in order to determine parameters with a higher likelihood.

In the other case, i.e. if $t \ll 1/\mu$, then the conditional probability converges to

$$\Pr(n \mid m) = \begin{cases} 1 & \text{if } n = m, \\ 0 & \text{else.} \end{cases}$$

5.1.2 Parameter Optimization

For a given data set, the origination and decay parameters λ and μ of the binding sites are estimated by a maximum likelihood (ML) procedure that uses iterative hill climbing for optimization, see Figure 5.2 for a outline. The algorithm consists of two nested optimization loops, one for maximizing the likelihood and one for determining the step length of the search algorithm.

The initial step sizes are determined as $\Delta_\lambda = \lambda/2$ and $\Delta_\mu = \mu/2$. Then, the likelihoods for all six combinations of parameters (λ', μ), (λ, μ') and (λ', μ') with $\lambda' = \lambda \pm \Delta_\lambda$ and $\mu' = \mu \pm \Delta_\mu$, are calculated. If the maximum L_{max} of these six new likelihoods is smaller than the original likelihood L' with parameters (λ, μ), then the step size is reduced by a factor $1/2$ and the step determination is repeated. If L_{max} is higher than L', then the step determination ends and the corresponding parameter combination is adopted. This procedure is repeated until the difference between L_{max} and L' is lower than the abort criterion ε (default is $\varepsilon = 10^{-6}$). In this case, the algorithm returns the last parameter pair as the ML estimate of the binding

site turnover rates. Note that for determination of step sizes for parameter optimization the evaluation of (λ', μ) and (λ, μ') would be sufficient. Because the likelihoods for this model are strongly influenced by the ratio λ/μ, the evaluation of likelihoods for (λ', μ') makes the optimization more efficient.

For an efficient optimization, it is also important to choose reasonable starting estimates for the turnover parameters. For this purpose, we assume that about 1/2 of the binding sites present at the root are still present in any of the species sampled. The expected number of homologous binding sites $\mathrm{E}[n_\cap]$ after time t and n binding sites at the begin is given by Equation 4.4 (p.91):

$$\mathrm{E}[n_\cap] = ne^{-\mu t}.$$

For $n_\cap = 1/2 \times n$, we initially estimate μ with $\ln(2)/(t)$. The origination rate is derived from the predicted equilibrium-binding site number $\hat{n} = \lambda/\mu$, i.e. $\lambda = \hat{n}\mu$. For $\hat{n}$ we take the average binding site number in the leaves for the subtree for which the parameters are estimated, while t is the average distance from the root of the subtree to the leaves.

If the evolutionary distances between nodes in the tree are long compared to the half-life of the binding site, the transient probability distribution approaches the stationary Poisson distribution with parameter λ/μ (Medhi, 2003). In this case, only this ratio is optimized by the algorithm by keeping the original μ and only adapting λ.

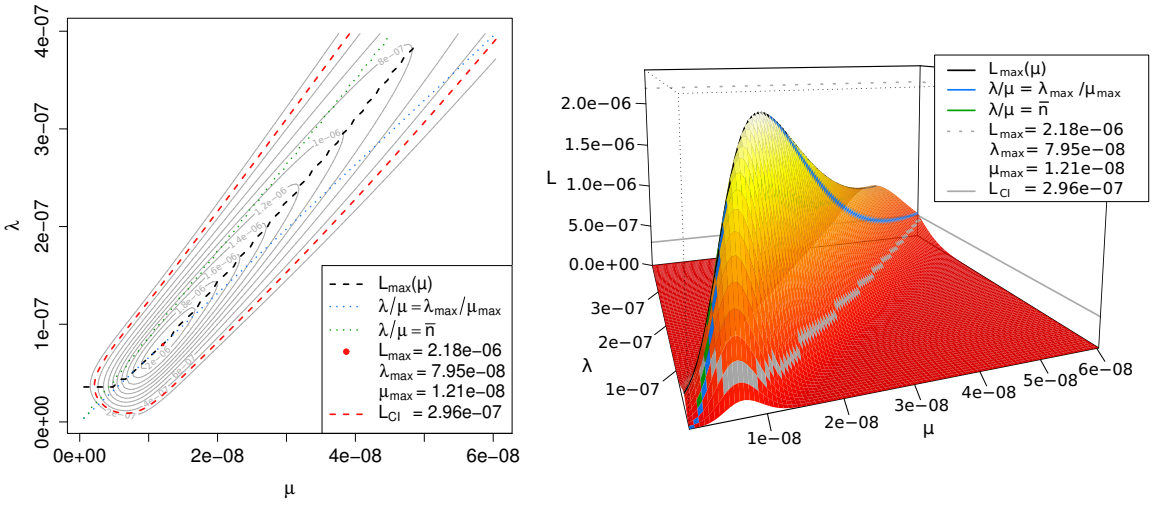
5.1.3 Model Characteristics

In Figure 5.3, the likelihood surface for the CBF1, a TF involved in the methionine pathway of yeasts, in a fungal data set consisting of 13 species is plotted. The likelihood function has a very distinct maximum and an extended ridge around the parameter values with the same λ/μ ratio as the ML estimates of λ and μ. This functional form demonstrates that deviations in the λ/μ ratio affect the likelihood more than individual parameter changes that leave the λ/μ ratio unaffected. We also note that the likelihood ridge levels off for larger parameter values, rather than approaching zero. Thus, the likelihood only depends on the λ/μ ratio and not on the individual parameters. This leveling off occurs because, as the turnover parameters increase, the estimated half-life time of a binding site decreases. Hence, the probability distribution approaches the stationary distribution, which only depends on the λ/μ ratio and not on the individual parameter values.

In order to determine the confidence intervals (CIs) of the parameter estimates, we approximate the likelihood function around the optimum with the Gaussian likelihood function

$$\mathrm{L}(\lambda/\mu) \approx e^{-\frac{1}{2}(v-v_0)^T C^{-1}(v-v_0)},$$

where $v = (\lambda/\mu)$ and $v_O = (\lambda_{opt}/\mu_{opt})$, and C is the covariance matrix for the parameter estimates. To give a rough estimate of the CI, we use the fact that the log-Gaussian likelihood



(a) (b)

Figure 5.3: Likelihood surface for the CBF1 transcription factor in a fungal data set in 2D (a) and 3D (b). The likelihood is plotted as a function of the origination rate λ and the decay rate μ. The likelihood function has a distinct maximum with an extended ridge (black). This ridge is located between the λ/μ rate corresponding to the mean binding site number (green) and the λ/μ rate corresponding to the λ/μ rate at the likelihood maximum (blue). The likelihood of the confidence interval corresponding to two standard deviations is also given in red (a) respective gray (b). Note that the likelihood of the ridge is higher than the likelihood of the confidence interval. Therefore, we have no upper confidence limit for λ and μ.

L for a random variable χ two standard deviations (SDs) σ away from the optimum is:

$$\log \mathrm{L}(\chi \pm 2\sigma) = \log \mathrm{L}_{\max} - 2.$$

Hence, we find the contour line on the likelihood function that corresponds to a likelihood value of

$$\mathrm{L}_{CI} = \exp[\log \mathrm{L}_{\max} - 2]$$

to determine the CIs for λ and μ. In case of normal distributed data, the 2 SDs correspond to the 95% CI. The limits of the CIs are then the maximal and minimal values which are compatible with L_{CI}.

There is always a lower confidence limit for λ and μ, but in cases where the phylogenetic signal in the data is weak, it can happen that there is no upper confidence limit. This degenerated case occurs when the likelihood for the equilibrium case is above the likelihood of the CI. Then both values can be arbitrarily high as long as the ratio λ/μ remains equal.

5.2 Simulation of Binding Site Evolution

In order to confirm our approach, we determine binding site numbers for a given phylogenetic tree by simulation of binding site evolution. Then we use our ML algorithm to determine the turnover rates and compare this result with the rates that have been used to generate the data.

In order to generate the data, we simulate the stochastic process of TFBS turnover. Given a tree with known binding site number at the root, and the parameters λ and μ of the model, the binding site number at a certain node of the tree is drawn randomly from a distribution given by the mathematical model based on Equation 4.2 (p.84). The randomly drawn binding site numbers at the terminal nodes of the tree is then taken as input data for the analysis by the ML algorithm described above.

The simulations are performed on linear and binary trees, i.e. trees with a pectinate structure and trees that are symmetrical. We simulate data over trees with $2, 3, 4, 6, 8$ and 16 taxa. Trees with fewer than 16 taxa are obtained by randomly deleting taxa from a symmetrical 16-taxa tree. We use a λ/μ ratio of 10 and we set the number at the root of the tree to the equilibrium-binding site number $\lambda/\mu = 10$. In order to obtain simulations with different relationships between clade age and turnover rate we perform simulations with different clade ages. We express the "age of the clade" in terms of relative clade age (RCA) , i.e. the age of the root node divided by the half-life time of the binding sites. Therefore, we set the root node to an age of 10^6 years and adjust the values of the parameters λ and μ by a linear factor. Based on Equation 4.4 (p.91), a RCA of 1 corresponds to a decay rate of $\mu = \ln(2)/10^6 = 6.93 \times 10^{-7}$. For each parameter set, 1000 simulationa are performed in order to minimize the influences of extreme random effects.

5.2.1 The Effect of Taxon Sampling

Here, we address the question, how much the estimate of the model parameters is affected by the number of taxa that are included in the analysis.

The first notable trend is that for is a considerable fraction of simulations the algorithm did not converge, see Figure 5.4. On the one hand, there are cases in which the likelihood of an equilibrium model equals that of the full model taking phylogenetic structure into account. In the equilibrium model, the likelihood of any binding site number on the tree is estimated from the Poisson distribution, i.e. the equilibrium distribution of the stochastic process of binding site turnover. Equilibration of binding site numbers among species can happen even in cases where the simulated clade is young compared to the half-life of binding sites and thus should not be in equilibrium. If this equilibration occurs, it is impossible to estimate the individual parameters because the equilibrium distribution only depends on their ratio. On the other hand, there are cases in which the parameter estimates diverged toward zero. These cases

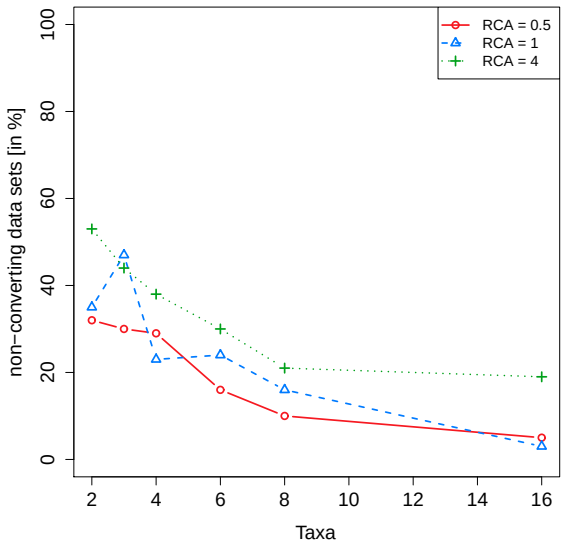


Figure 5.4: Rate of non-converting data sets in dependence of taxa numbers for different relative clade ages (RCAs) on binary trees. The percentage of simulated data sets for which the algorithm did not converge ranged from about 30% for data with $2, 3$ and 4 taxa to only 5% in the case of 16 taxa. With older clades, i.e. a RCA of 1 or 4, the percentage of non-converging is larger because they are more likely to be in equilibrium.

occur for instance when, by chance, the binding site counts of taxa are too similar suggesting a low rate of evolution. Taking these cases together, the percentage of simulated data sets for which the algorithm does not converge ranges for young clades with a RCA of 0.5 from about 30% for data with $2, 3$ and 4 taxa to only 5% in the case of 16 taxa. Hence, there is a large chance (about 30%) that small data sets, i.e. 4 taxa or less, cannot be analyzed because there is a high probability of a data structure that is "misleading" to the algorithm. In our simulations, however, the chance that the algorithm is not converging is only 10% with eight taxa and 5% with 16 taxa. With older clades, RCA 1 or 4, the percentage of non-converging is larger because they are more likely to be in equilibrium.

All the results discussed below are based on only those simulations in which the algorithm converged, i.e. the likelihood of the full model is higher than the equilibrium model and no parameter estimate diverged close to zero. Note that there is no ambiguity in classifying the cases as converging or diverging. The simulated parameters are of the order of 10^{-5} and 10^{-6} for λ and μ, respectively. The optimization stops if the start parameters falls more than six orders of magnitude, typically 10^{-17} or lower.

5.2.2 Estimating the λ/μ Ratio

Here, we consider the influence of the mean binding site number in the taxon sample on the estimated ratio of λ/μ. Remember that the ratio λ/μ predicts the expected binding

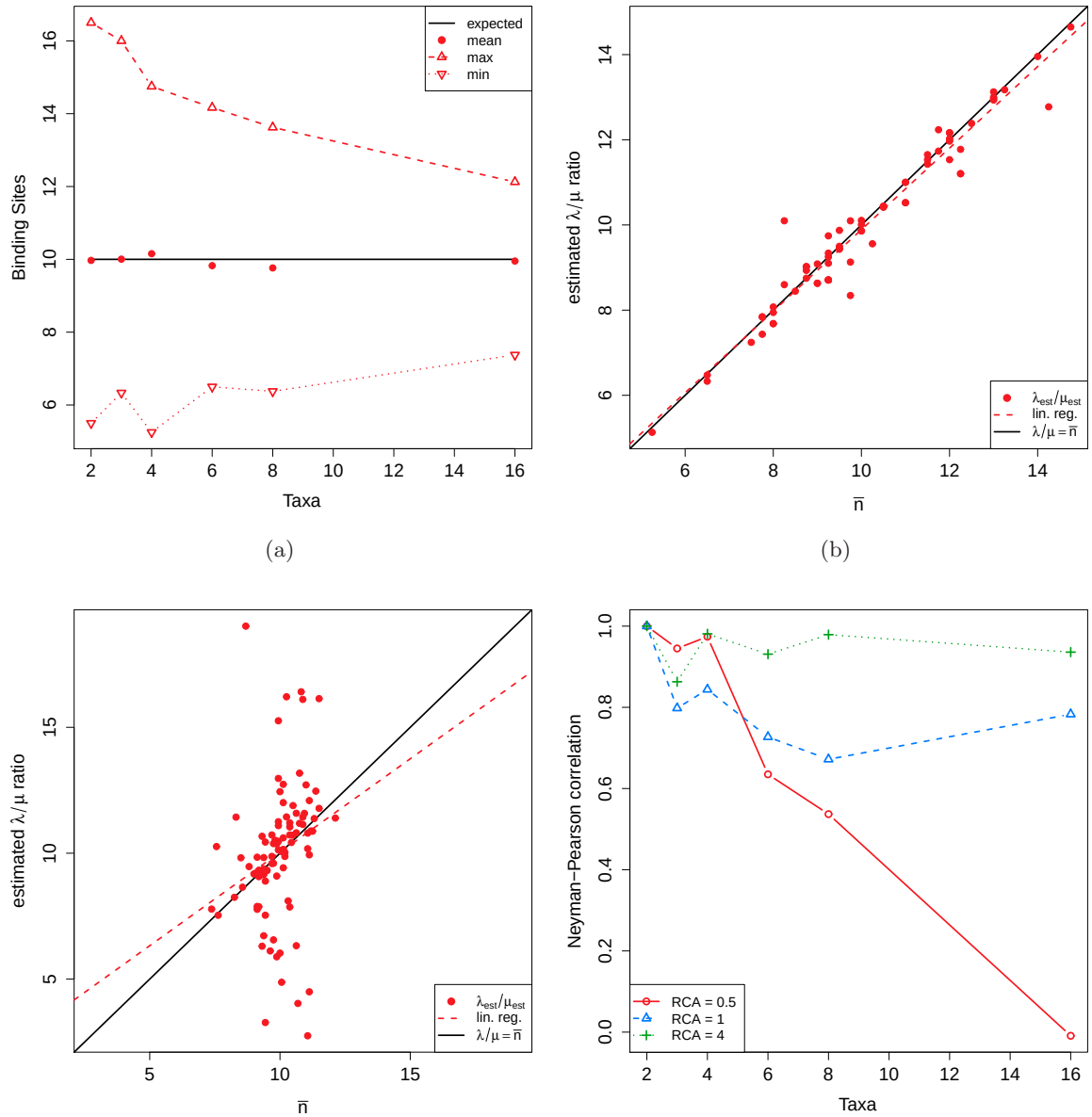


Figure 5.5: Influence of the mean binding site number on the estimated λ/μ ratio for binary trees. (a) The mean binding site number (RCA $= 0.5$) tends to differ more from the equilibrium number for few taxa than for many taxa. (b) The correlation between the binding site number and the estimated λ/μ ratio is for 4 taxa and a relative clade age (RCA) of 0.5 very strong (Neyman–Pearson: 0.974). (c) In contrast the correlation for 16 taxa at the same RCA is virtually 0 (Neyman–Pearson: -0.009). (d) In general, this correlation falls with rising taxa numbers and increase strongly with higher RCA. This suggests that λ/μ estimates are more reliable with high taxa numbers and low RCA.

site number in stochastic equilibrium. In the simulations, the root node is assigned to the predicted equilibrium binding site number, in our case 10. Nevertheless, the mean binding site number differs in some cases quite dramatically from the equilibrium expectation. In Figure 5.5 (a), we plot the minimum, the mean, and the maximum over all mean binding site numbers of simulations with the same parameters. While the mean corresponds to the expected binding site number, the extrema indicate a high deviation. The differences between the mean binding site number and the extrema get smaller with higher taxa numbers, but even for "large" data sets with 16 taxa, the minimum mean binding site number is 7.4 and the maximum is 12.1. These are $20 - 25\%$ different from the expectation of 10.

One can expect that the algorithm would over- and underestimate λ/μ if the mean binding site number in a data set deviates strongly from the expectation. In fact, for small data sets, there is a strong correlation between the mean binding site number in a data set and the estimated λ/μ ratio. This correlation is shown in Figure 5.5 (b) for a sample of data sets with four taxa. The Neyman–Pearson correlation is 0.974 and the scatter is very tight. On the other hand, in a sample of data sets with 16 taxa, the correlation is with -0.009 virtually zero, see Figure 5.5 (c). Hence, with a moderately large data set and a young clade, for example, 16 taxa and a RCA of 0.5, the algorithm is able to correctly estimate the λ/μ ratio, even when the mean binding site number deviates strongly from the expected equilibrium-binding site number.

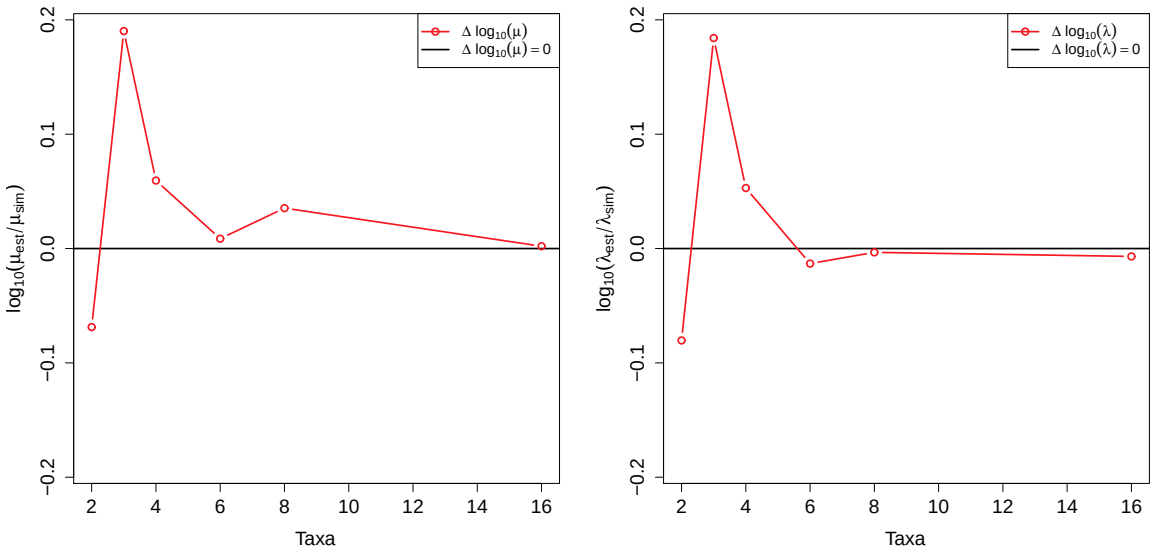
We further investigated the relationship between mean binding site number and estimated λ/μ ratio for older clades with RCA of 1 and 4, see Figure 5.5 (d) for data sets of $4, 6, 8$ and 16 taxa. We find that the correlation between mean binding site number and estimated λ/μ ratio remains high even from clades with RCA $= 1$ and 16 taxa. It seems that the demand on data amount increases strongly with RCA so that the algorithm can estimate the correct λ/μ ratio. For instance, with a RCA of 1 and 16 data, the correlation is still 0.78 and with a RCA of 4, it is higher than 0.9.

We conclude that the accuracy and reliability of the estimated λ/μ ratio critically depends on the number of taxa sampled. With sufficient data and clades that are young enough relative to the half-life of the binding site, the algorithm accurately estimates the equilibrium binding site density for a clade, even when the mean binding site number observed is considerably different from the expectation. This is also the case for linear trees.

5.2.3 Estimating the Individual Model Parameters

Since the λ/μ ratio strongly influences the expected mean binding site number among taxa, this ratio is much easier to estimate than the individual parameters, λ and μ. In Figure 5.6, we plot the difference between the estimated rates and the rates that are used for the simulations in a $\log_{10}$ scale.

These data show that the average accuracy of parameter estimates can be pretty good

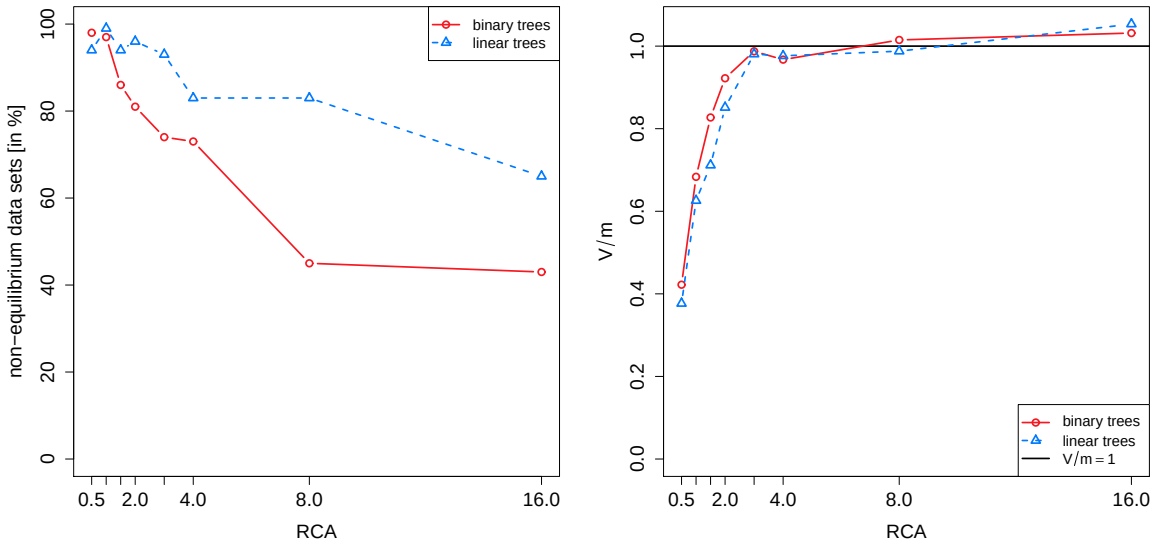


(a) (b)

Figure 5.6: Influence of the number of taxa on the accuracy of the individual parameter estimates for binary trees and an relative clade age of 0.5. The values for μ (a) as well as the values for λ (b) show relatively accurate estimates from six taxa on. For smaller taxa numbers, the predictions become more inaccurate which suggest that parameter estimation requires at least six taxa.

all the way down to samples of six taxa in a clade with RCA of 0.5. Hence, there seems to be no bias in the estimates. The SDs for both $\log_{10}\mu$ as well as $\log_{10}\lambda$ estimates are pretty high, roughly around 0.3, and slightly higher for smaller data sets but generally in the same order of magnitude. Evaluation of the SDs of the estimated parameters yields to CIs for these estimates. These SDs translate into factor of two differences between the estimated and the real parameters. The 95% CIs then would be compatible with roughly a 4-fold difference between estimates and true parameter values. Point estimates of binding site turnover parameters are expected to be inaccurate up to a 4-fold difference even with young clades and a moderately large taxon sample below 16. Estimated parameter values can thus only be considered as order of magnitude estimates.

These trends are remarkably similar to clades of different age. For $\log_{10}\lambda$, the SD is 0.369 if averaged over taxa numbers and clade ages. This value varies between 0.49 and 0.28 if averaged over clade age and between 0.41 and 0.32 if averaged over taxa. For $\log_{10}\mu$, the overall average SD is 0.365 and shows a similar range of variation as $\log_{10}\lambda$. Hence, the CIs for parameter estimates are not much influenced by either taxon number or clade age. Samples with more taxa are better for estimating parameters but only slightly.



(a) (b)

Figure 5.7: Effects of the relative clade age (RCA) for 16 taxa. (a) While young clades are almost every time outside the equilibrium state, the fraction of non-equilibrium data sets tended to slowly decrease in older clades. (b) The variance/mean ratio V/m of the binding site numbers corresponds for large clade ages to the expected equilibrium rate $V/m = 1$. For small RCAs, V/m is below 1, indicating a phylogenetic signal in the data for binary as well as for linear trees. Hence, clades which are at least three times as old as the binding site half-life are expected to be in equilibrium.

5.2.4 The Effect of Clade Age

We performe simulations of binding site turnover in clades of 16 taxa with either binary trees or linear trees. We simulated clades of different age that were composed of the same number of taxa. We express the clade age in terms of RCA and simulate clades with 16 taxa and eight different RCAs, in detail 0.5, 1, 1.5, 2, 3, 4, 8 and 16. In Figure 5.7 (a), we show the percentage of simulations for which the likelihood of the equilibrium model is less than that of the full model. This difference between likelihoods is manifested in almost all cases with young clades, i.e. with an RCA of 0.5 and 1. With older clades, the fraction of non-equilibrium data sets decreases slowly. It reaches its lowest level of $40 - 60\%$ in old clades with an RCA of 16. In old clades, binary trees seem to produce a smaller fraction of non-equilibrium data sets than linear trees. This difference may occur because linear trees with the same number of taxa will have more recent nodes than in a binary tree of equally spaced internal nodes.

In Figure 5.7 (b), the variance/mean (V/m) ratio, i.e. the ratio of the variance in binding site number across species divided by the mean binding site number, is plotted as a function of the RCA. Note that for the equilibrium data set, we expect $V/m = 1$ and phylogenetic signal should lead to $V/m < 1$. The results for linear and binary trees are very similar, with

binary trees having slightly larger average V/m for young clades. The V/m ratio starts out around 0.4 for RCA = 0.5 but quickly approaches the equilibrium value of 1 at an RCA of 3. Hence, clades which are three times as old or older than the binding site half-life are expected to approach an equilibrium value of $V/m = 1$.

Estimates of the binding site loss rate μ on binary trees are relatively accurate in clades of RCA up to 4 but were then systematically biased downwards, see Figure 5.8 (a). The bias decreases approximately linearly, with a regression equation of $\Delta \log_{10}(\mu) = -0.037 \times \text{RCA} + 0.023$. In contrast, μ estimates derived from linear trees, see Figure 5.8 (c), are accurate for RCA = 0.5 but seem to be biases toward higher values for RCA = 1 to RCA = 4. The average $\Delta \log_{10}(\mu)$ is 0.085, i.e. the actual μ estimates are about 22% higher than the simulated values. Above RCA = 4 the bias in μ estimates decreases with $\Delta \log_{10}(\mu) = -0.025 \times RCA + 0.079$.

For binary trees, estimates of the birth rate of binding sites, λ, are relatively accurate in the same range as the μ estimates, see Figure 5.8 (b), i.e. between RCA = 0.5 and 4. For RCA 4 and 8, we find a positive bias in λ estimates but with RCA = 16, a curious inversion of the trend is observed with a negative bias for λ, whereas the μ estimates have a positive bias. A similar pattern is found for linear trees, see Figure 5.8 (d), except that for RCA between 1 and 3 the λ estimates have a negative bias. The reversion of bias in μ and λ might be related to the fact that the estimates of very old clades are mainly influnced by the λ/μ ratio and any stochasticity in the binding site density in terminal taxa leads to higher μ rate estimates.

Overall, these simulations show that the method performs well in estimating binding site turnover rates in young clades with RCA = 0.5 regardless of tree structure, and moderately well up to RCA = 4 for binary trees. In older clades, μ estimates seem to be systematically biased toward larger values and λ estimates seem to have variable biases depending on RCA.

5.2.5 The "Back of the Envelope" Method

The dependency of V/m on RCA shown in Figure 5.7 (b) suggests that V/m follows an inverse exponential function of RCA

$$V/m \approx 1 - \exp[-k \times \text{RCA}]. \tag{5.1}$$

This functional form is reasonable given that the effect of history decreases exponentially with time in our model derived in Chapter 4. In fact, the model estimating the coefficient k from the simulation data by regressing $-\ln(1 - V/m)$ onto RCA gives a good agreement between the simulated data and the inverse exponential function. When simulating smaller data sets for 4 and 8 taxa, in order to see whether the rate of increase in V/m depends on the number of taxa, we find that the rate of increase is larger with more taxa than with fewer, see Figure 5.9. The respective regression coefficients are for 16 taxa $k = 1.07$, 8 taxa $k = 0.81$, and 4 taxa $k = 0.58$. The coefficient increases almost linearly with the number of

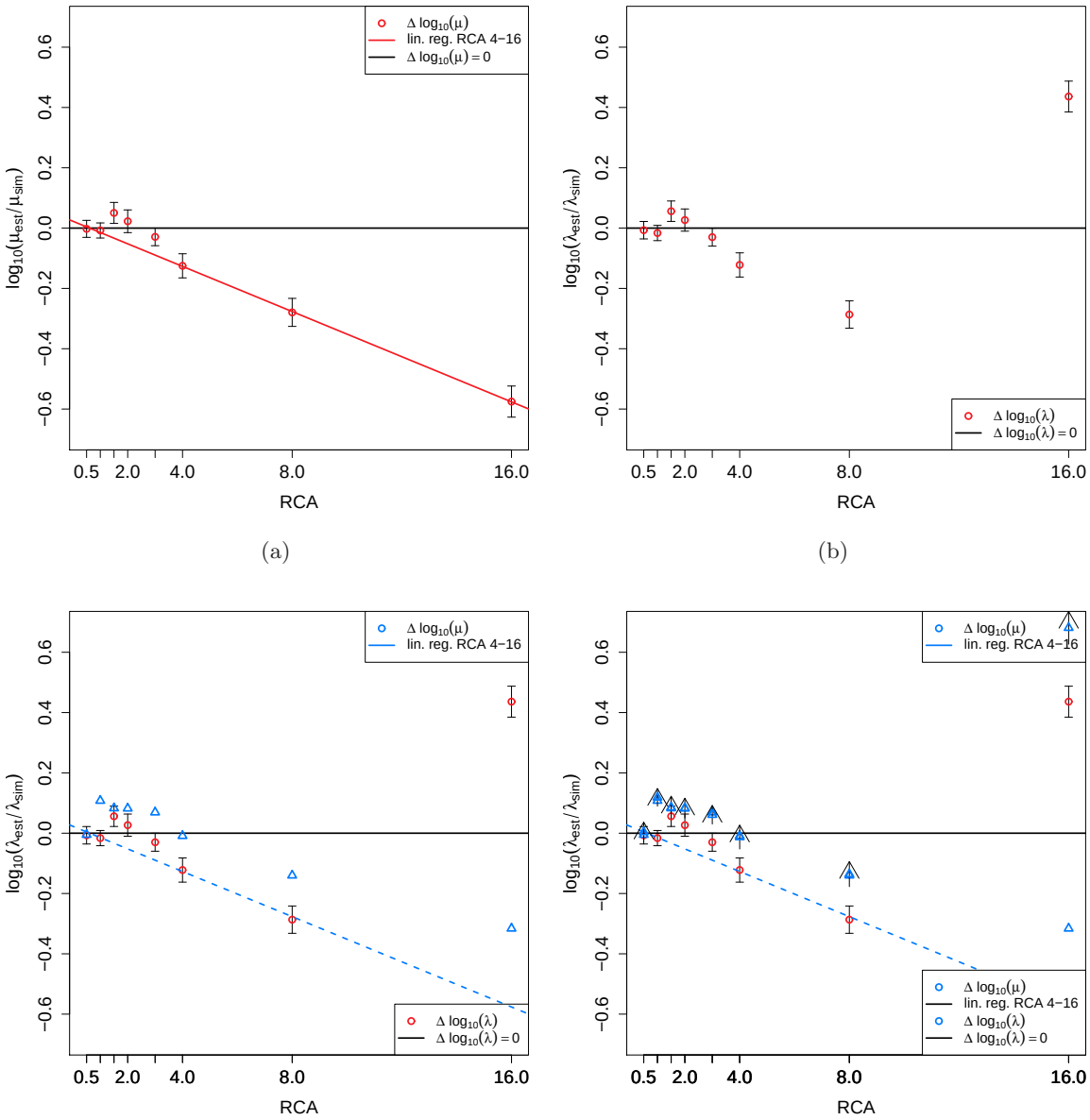


Figure 5.8: Influence of the relative clade age (RCA) on the accuracy for binary trees and 16 taxa. (a) For binary trees, the estimates of μ are relatively accurate up to a RCA of 3 and start then to be biased toward higher values. (b) The estimates of λ act similarly except the negative bias at a RCA of 16. In case of linear trees the estimates of μ (c) and λ (d) are similar to that on binary trees.

taxa, showing a regression of

$$k = 0.04 \times N_{\text{taxa}} + 0.45 \ . \tag{5.2}$$

Using these empirically determined relationships, we can estimate the model parameters.

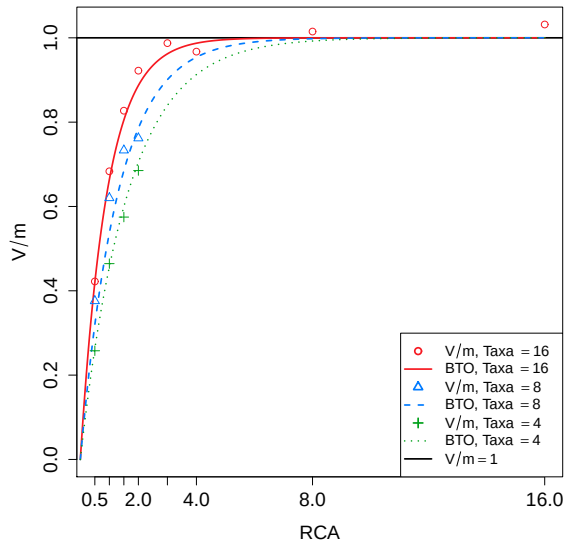


Figure 5.9: Influence of the relative clade age (RCA) and the taxa number on the V/m ratio of the binding site numbers for binary trees. The inverse exponential function based on Equation 5.1 (lines) gives a good agreement with the simulated data (points).

Given a clade with N taxa and clade age T as well as the mean m and the variance V of the binding site numbers for a TF with a V/m ratio substantially smaller than 1, we can estimate the RCA of the clade relative to this transcription factor (TF) by

$$\text{RCA} = \frac{-\ln(1 - V/m)}{0.04 \times N + 0.45}.$$

From that we obtain the half-life of the TFBS using the clade age

$$T_{1/2} = \frac{T}{\text{RCA}},$$

which then relates to the decay rate μ of the TFBS by

$$\mu = \frac{-\ln(0.5)}{T_{1/2}}.$$

If we assume that the mean binding site number is in equilibrium, the origination rate λ can also be calculated by

$$\lambda = m \times \mu.$$

However, the mean binding site number does not need to be in equilibrium, and thus, this latter estimate should be considered with caution. Although this *Back Of the Envelope* method (BOE) is just an approximation, it is useful to get a rough estimate for the parameters based only on the V/m ratio.

5.3 Application to Biological Data

The motivation for developing the method was to be able to detect differences in selective constraints acting on cis-regulatory regions. Specifically, we like to determine whether biologically interesting patterns can be detected using this method in terms of lineage- or clade-specific differences in turnover parameters. Here, we analyze two data sets in order to see how the method performs on biological data. The first data set is based on yeast genes (phylum Ascomycota) related to the methionine biosynthetic pathway while the second data set is based on hormone response elements in the mammalian (phylum Chordata) HoxA gene clusters.

5.3.1 The Methionine Pathway of Yeasts

The upstream regions for the predicted protein-coding genes across thirteen yeast species were obtained from Wapinski *et al.* (2007). Out of 27 methionine biosynthesis genes compiled by Gasch *et al.* (2004), eight genes, in detail Met2, Met3, Met6, Met8, Met10, Met16, Met30 and Sah1, were identified as single-copy orthologous in all thirteen species using the `Fungal Orthogroups Repository`.

To determine the TFBS density, we use a compilation of 80 binding sites represented by consensus sequences (Gasch *et al.*, 2004) in IUPAC nucleotide code. We searche 500 bp upstream of our eight orthologous single-copy genes for each of the binding sites, counting the occurrences in both strands. In 41 of 104 cases, the non-coding upstream sequence was less than 500 bp, so the count was extrapolated linearly to this value (e.g., if eight binding sites were found in 400 bp of upstream sequence, our final count was assigned as 10). In two outlier cases, we instead assigned the binding site count from the closest relative to two species because they had upstream sequences of 1 and 37 bp. Out of 80 TFBSs, five were found to be enriched in the single-copy orthologous genes compared to the rest of the genome (bound by Bas1p, Cbf1p, Gcn4p, Met30/31p, and Rtg1/Rtg3p). Enrichment was inferred by applying Fisher's exact test to compare the times the binding site was found for all species in the single-copy orthologous genes (104 promoter regions total) and in the rest of genome (73944 promoters). To get the final binding site count for these TFs, we summed up all the occurrences in the eight related genes for each of the species. All these processes were automated using `PERL` scripts. The results are summarized in Table 5.1.

To obtain the chronogram (ultrametric tree with branch lengths proportional to time), we used the phylogenetic tree topology from Wapinski *et al.* (2007). We estimated divergence times by penalized likelihood with a truncated Newton algorithm in `r8s` version 1.71 (Sanderson, 2006) setting the smoothing parameter to 0.06. The tree species phylogeny was calibrated by fixing the split of *Debaryomyces hansenii* and *Candida albicans* from the other yeast at 272 million years (Myr) (Miranda *et al.*, 2006).

Scientific name	BAS1	CBF1	GCN4	MET30/31	RTG1/3
Saccharomyces cerevisiae	3	12	10	8	5
Saccharomyces paradoxus	3	11	10	9	7
Saccharomyces mikatae	5	10	9	6	11
Saccharomyces bayanus	4	10	8	5	7
Candida glabrata	2	5	4	4	15
Saccharomyces castellii	1	5	8	2	6
Kluyveromyces lactis	2	11	4	6	9
Ashbya gossypii	2	11	2	7	17
Kluyveromyces waltii	1	3	1	4	4
Candida albicans	2	1	4	1	5
Debaryomyces hansenii	5	2	9	1	2
Yarrowia lipolytica	5	6	8	16	15
Schizosaccharomyces pombe	0	0	1	0	2

Table 5.1: Number of nuclear receptor response elements significantly enriched in eight single copy orthologous methionine biosynthesis genes (Met2, Met3, Met6, Met8, Met10, Met16 Met30, Sah1) in yeast.

Data Analysis

The mean binding site density, averaged over species, is highly variable, ranging from 2.7 for BAS1-binding sites to 8.1 for RTG1/3-binding sites. In order to explore the evolution of binding site density, we computed the mean binding site density and the ratio of binding site variance to mean (V/m ratio) for various clades of yeast species. According to our model, the V/m ratio is predicted to be equal to 1 if the binding site density distribution is in equilibrium, i.e. if the time of separation among the lineages is long enough to erase the phylogenetic signal, see Equation 4.3 (p.84). If the clade is younger, the model predicts that the V/m ratio is less than 1. If, however, the lineages are separated for a long time and differ in their expected binding site densities, the V/m ratio can be larger than 1. With these criteria in mind, we can perform an exploratory analysis of our data.

Figure 5.10 (a) shows the relationship of the V/m ratio over the estimated age of the clade. In the youngest clade D (cf. Figure 5.10), consisting of 4 species most closely related to *Saccharomyces cerevisiae* (about 20 Myr), the V/m ratio for all five binding sites is less than 1 (sign test: $P = 0.0125$) and two binding sites individually have a V/m ration significantly less than 1 (CBF1: $P = 0.032$; GCN4: $P = 0.039$). For older clades, the V/m ratio tends to be larger but with considerable variation among binding site classes, roughly consistent with the expectation of the model. In all binding site classes, the count of binding sites in *Schizosaccharomyces pombe* is low and will be excluded from further analysis. Below, we summarize the variation in binding site density for individual binding site classes.

BAS1: See Figure 5.10 (b). The binding site density for BAS1 is low, with an average of 2.69 and ranging from 0 in *Schizosaccharomyces pombe* to a maximum of 5. In general, mean

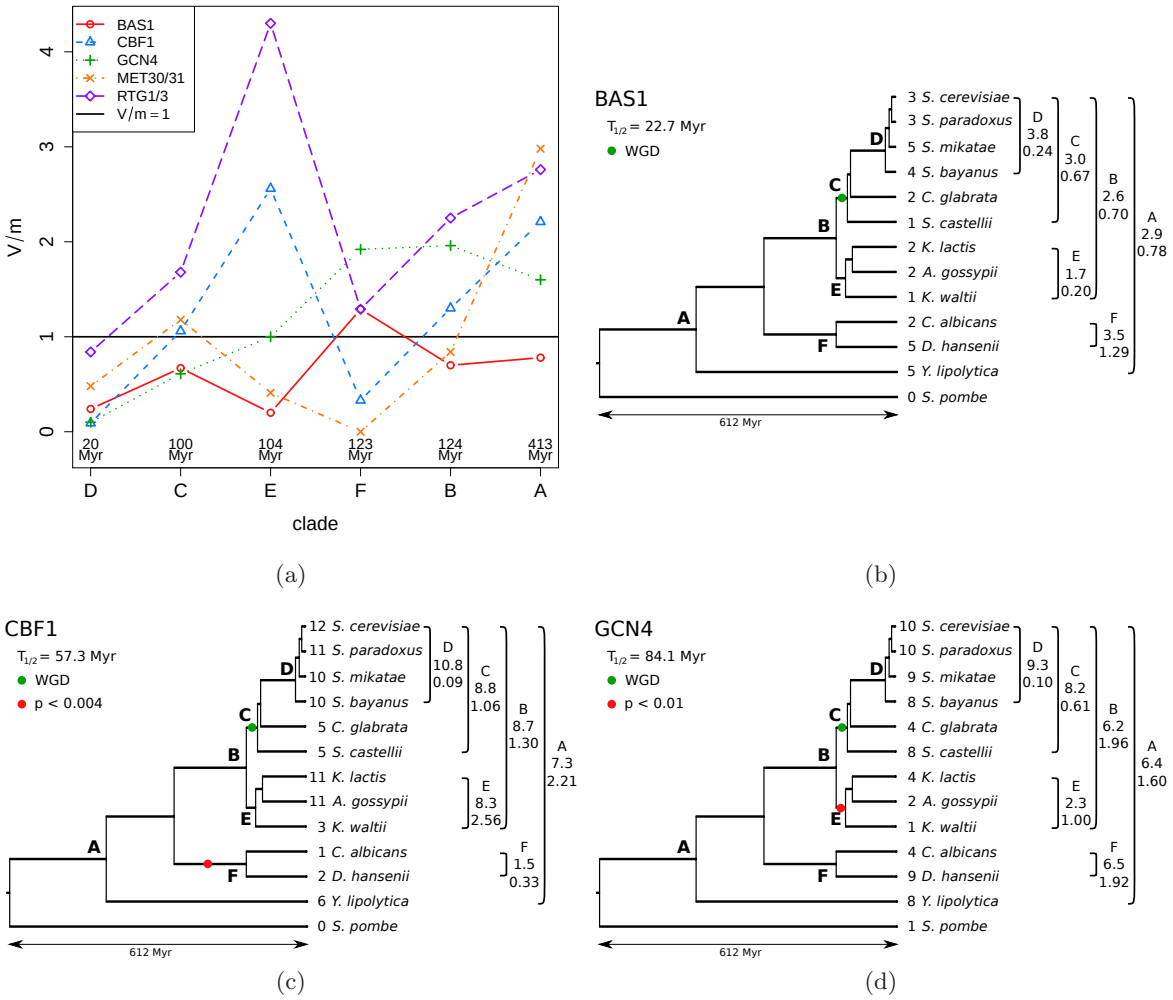
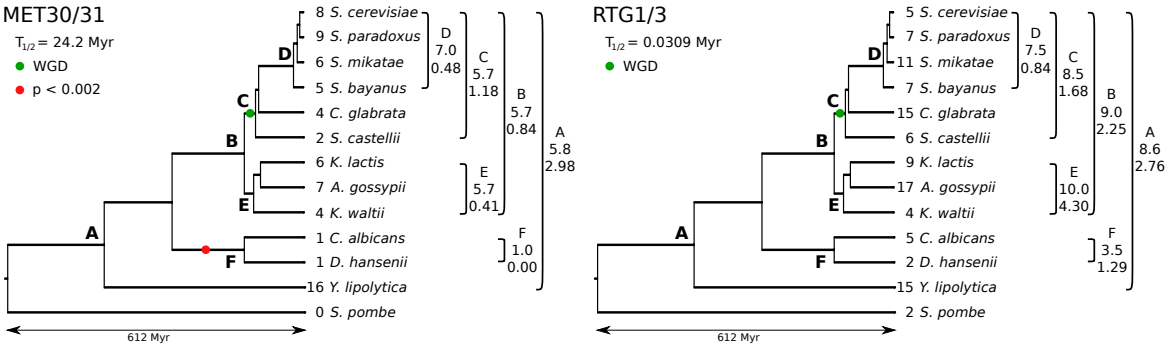


Figure 5.10: Evolution of binding site numbers in the methionine pathway of yeast (A). The first diagram (a) shows the relationship of the clade age and the V/m ratio of binding sites. The letters refer to the clade labels in (b) to (f). Whereas for the youngest clade D the V/m ratios for all transcription factors are less than 1, the ratios tends to be large for older clades but still have a considerable variation among binding site classes. Note that V/m ratios larger than 1 suggest heterogeneity of binding site number among taxa. For the remaining diagrams, clades are indicated by a letter at their root and a bracket displaying the mean binding site number and the V/m ratio behind the corresponding binding site numbers and species names. The whole-genome duplication of the C clade is marked by *WGD*. For BAS1 (b), the likelihood model estimates a half-life of 22.7 Myr. CBF1 (c) has an estimated half-life of 57.3 Myr. There also seems to be a significant decrease in the binding site density in stem lineage of the F clade. GCN4 (d) has with 84.1 Myr, the longest estimated half-life and a significant loss of binding sites in the E clade.

binding site density is similar among clades and the V/m ratio generally remains below 1. There is no particular pattern to binding site density differences. The estimated origination rate is 0.0244 binding sites/(kb $\times$ Myr) for the 4000 bp area consisting of



(e) (f)

Figure 5.10: Evolution of binding site numbers in the methionine pathway of yeast (B). MET30/31 (e) has an estimated half-life of 24.2 Myr and like CBF1 a significant binding site loss in the F clade. RTG1/3 (f) has in the D clade a V/m ratio of 0.84. This is close to 1, suggesting equilibrium within the 20 Myr time frame of the D clade. This makes the estimation of the rate parameters difficult and yields to the shortest half-life of only 30,900 years.

500 bp upstream for each of our eight orthologous single-copy genes. The half-life for a BAS1-binding site is estimated to be 22.7 Myr.

CBF1: See Figure 5.10 (c). CBF1-binding site density in the A clade (i.e., all species except *Schizosaccharomyces pombe*) is 7.3. The V/m ratio in the A clade is significantly elevated above 1 ($V/m = 2.21$, $P = 0.012$) and thus indicates heterogeneity in average binding site density among lineages. This heterogeneity arises from differences among the B clade and its two outgroup clades, clade F and *Yarrowia lipolytica*. The B clade itself has a V/m ratio of 1.30, which is not significantly elevated above 1 ($P = 0.30$). Within the B clade, the youngest clade D has a significantly decreased V/m ratio of 0.09 ($P = 0.032$). This indicates that the half-life of the CBF1-binding sites is longer than the age of the D clade. Estimating the half-life with the likelihood model yields a value of 57.3 Myr, which is almost three times the age of D clade, about 19.8 Myr. Comparing the binding site densities in the F clade (mean is 1.5), the direct outgroup of B, suggests that binding site density might have been decreased in the stem lineage of the F clade. A likelihood ratio test using our model suggests that this rate difference is significant ($\chi^2 = 11.05$; $P = 0.004$, 2 degrees of freedom). The ML-estimated CBF1-binding site origination rate is 0.02 binding sites/(kb $\times$ Myr).

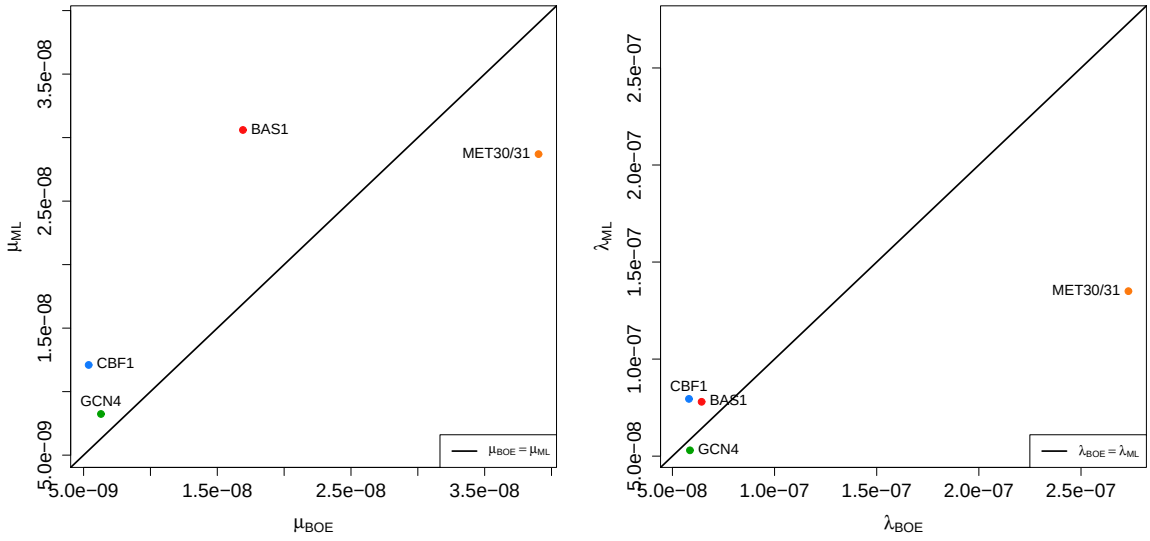
GCN4: See Figure 5.10 (d). The average density of GCN4-binding sites is 6.4 and has only a slightly elevated V/m ratio of 1.6 ($P = 0.09$). In contrast, the B clade has significant heterogeneity as indicated by a V/m ratio of 1.96 ($P = 0.047$), which is caused by a lower binding site density in the E clade (mean = 2.3, $V/m = 1.0$) than in the C clade (mean = 8.2, $V/m = 0.61$; $P = 0.31$). The outgroups of the B clade, F

clade, and *Yarrowia lipolytica* are more similar to the C clade (combined mean = 7.0, V/m = 1.0) suggesting that the E clade lost GCN4-binding sites in evolution. This inference is supported by a likelihood ratio test with our model ($\chi^2 = 9.29$; $P = 0.0096$, 2 degrees of freedom). GCN4 has the longest estimated half-life among the binding sites investigated here, 84.1 Myr, and a slightly lower than average origination rate of 0.013 binding sites/(kb × Myr).

MET30/31: See Figure 5.10 (e). Over all species (except *Schizosaccharomyces pombe*), the binding site density for MET30/31 is 5.8 and heterogeneous, with a significantly elevated V/m ratio of 2.98 ($P = 5.810^{-4}$). The heterogeneity is caused by a difference between the binding site densities in the B clade, which has a mean of 5.7 and a V/m ratio of 0.84, and the two outgroups. The first outgroup, the F clade, has a much lower binding site density of 1 while the second outgroup, *Yarrowia lipolytica* has an elevated binding site density of 16. Testing for a decreased binding site density with our likelihood model supports the inference that the F clade has a lower equilibrium density ($\chi^2 = 13.02$; $P = 0.0015$, 2 degrees of freedom). The B clade itself seems to be homogenous ($P = 0.43$). The half-life of MET30/31-binding sites is relatively short, at 24.2 Myr, and the origination rate is 0.034 binding sites/(kb × Myr).

RTG1/3: See Figure 5.10 (f). The average binding site density of RTG1/3 in the A clade is 8.6 and highly heterogeneous, $V/m = 2.76$ ($P = 1.4 \times 10^{-3}$). The heterogeneity arises at various levels. As for MET30/31, the density is high in *Yarrowia lipolytica* ($m = 15$) and depressed in the F clade ($m = 4$). Also, the B clade is heterogeneous with a mean of 9.0 and a V/m ratio of 2.25 ($P = 0.021$). The heterogeneity in the B clade is caused by heterogeneity in the E clade, which has a mean of 10 and a V/m ratio of 4.3 ($P = 0.014$), where the densities range from 17 to 4. ML estimates of the rate parameters are not possible with accuracy because the reconstructions suggest that the process quickly equilibrates, which only makes the ratio of rate parameters determined by data. This observation is consistent with the relatively high V/m ratio of 0.84 among the species of the D clade, the youngest clade in our taxon sample. This V/m ratio is not significantly different from 1 suggesting equilibration within the 20 Myr time frame of the D clade. Using the BOE method, which is just based on the V/m ratio, the data suggests a half-life of only 6.2×10^6 and an origination rate as high as 0.21 binding sites/(kb × Myr).

Comparing the data among binding site types suggests that binding site densities come in two modes. One mode ranges from 5 to 10 sites and the other consists of lineages having binding site densities of 1 or 2. Given that binding sites were sampled from the 5′ region of eight genes, binding site densities of 1 or 2 probably represent spurious binding site distributions, suggesting that this TF is not functional in the methionine pathway of the respective species.



(a) (b)

Figure 5.11: Comparison of the Maximum Likelihood and the Back of the Envelope estimates. Between both methods exists a correlation ($r^2 = 0.764$) for the μ estimates (a). In contrast, the λ estimates (b) have a low correlation of 0.392.

If this interpretation is correct, it seems that certain TFs have been replaced in some clades, often in the *F* clade, by some other TF, as has been demonstrated for the ribosomal protein module (Tuch *et al.*, 2008). This scenario applies to RTG1/3, MET30/31, and CBF1 in the *F* clade, represented here by *Debaryomyces hansenii* and *Candida albicans*. A similar drop in binding site density happens in the *E* clade, represented by *Ashbya gossypii*, *Kluyveromyces lactis* and *Kluyveromyces waltii*, for the GCN4-binding sites. These inferences from the binding site density patterns predict a shift in the functional role of the respective TFs. Another possibility is that in this clade, the binding site motifs and the TF DNA-binding specificity has changed. Thus, the TFBSs are no longer recognizable.

We also compare the ML and the BOE estimates of the model parameters. The ML estimates use an explicit likelihood framework to estimate parameters for a given phylogeny, whereas the BOE method only uses empirical relationships between V/m to estimate parameters. Given the difficulties of estimating these parameters, the correlation between the ML and BOE estimates of μ are quite reasonable ($r^2 = 0.764$) excluding RTG1/3 which inflates the correlation, see Figure 5.11 (a). In contrast, the λ estimates have a low correlation of 0.392, excluding MET30/31, which is an outlier, see Figure 5.11 (b). However, the three binding site origination rate estimates for CBF1, BAS1,and GCN4 are close to the line where the BOE and ML rate are equal.

Species	ERE	PRE	RARE
Human	12	12	3
Chimp	14	7	4
Baboon	11	8	5
Marmoset	17	10	6
Lemur	9	14	2
Galago	9	12	4
Mouse	12	17	4
Rat	15	14	4
Rabbit	11	16	4
Bat	17	19	3
Dog	12	15	3
Armadillo	11	8	3
Elephant	16	12	5
Opossum	12	13	2
Platypus	14	10	3
Chicken	8	7	4
Frog	1	8	4
Coelacanth	7	4	0
Bichir	3	3	4
Shark	7	7	3

Table 5.2: Number of nuclear receptor response elements in the vertebrate HoxA clusters.

5.3.2 The Vertebrate HoxA Cluster

The complete HoxA clusters from 20 vertebrates, in detail human, chimp, baboon, marmoset, lemur, galago, mouse, rat, rabbit, bat, dog, armadillo, elephant, opossum, platypus, chicken, frog, coelacanth, bichir and shark, were downloaded from NCBI, see Table A.1 (p.133) in the appendix. The sequence around the cluster was trimmed to include 5 kb of sequence upstream of the most 5′ Hox gene and 5 kb downstream of the most 3′ Hox gene.

Putative estrogen, progesterone, and retinoic acid response elements are identified using motif finding programs and in-house `PERL` scripts based on regular expressions. The numbers for ERE is determined using the `MATCH` algorithm (Kel *et al.*, 2003) and the PWM created by 41 natural vertebrate ERE motifs (Klinge, 2001) with a threshold of 0.85 as described in Section 4.2.1 (p.89) in the previous chapter. The progesterone response element (PRE) consensus is based on experimentally determined high-affinity progesterone receptor binding sites (Nelson *et al.*, 1999). The motif `TCTTGTNNNACAAGA` has a variable half-site and a fixed half-site, separated by three arbitrary nucleotides. One substitution in the six nucleotides of the fixed half-site and three mismatches to the perfect PRE are allowed in the variable half-site. For the detection of the binding sites of the retinoic acid response element (RARE) , we perform an exact forward and reverse search of 18 experimentally defined DR5 RAREs (Mainguy *et al.*, 2003) that consist of five defined nucleotides on each side separated by five arbitrary nucleotides. The results are summarized in Table 5.2.

Data Analysis

In the total data set (clade A), containing 15 mammalian species (clade B) and 5 non-mammalian species, the steroid response elements (SRE) , i.e. either ERE or PRE, are significantly overdispersed, suggesting heterogeneity among lineages in terms of the ERE and PRE density, see Figure 5.12. Specifically, the V/m ratio for PRE is 1.73 ($P = 0.025$) and for ERE 1.68 ($P = 0.013$). This heterogeneity is caused by a difference between the mammalian and the non-mammalian taxa. The mammalian clade has higher SRE densities (PRE: mean is 10.8; ERE: mean is 10.9) than the non-mammalian species (PRE: mean is 5.8; ERE: mean is 5.2). The mammalian clade shows no evidence of heterogeneity (PRE: V/m is 0.97; ERE: V/m is 0.53; $P = 0.17$). This suggests that the SREs experienced a twofold increase in equilibrium density from about 5 to about 10. The likelihood ratio test using our model supports this conclusion (PRE: $\chi^2 = 17.67$, $P = 1.46 \times 10^{-4}$, 2 degrees of freedom; ERE: $\chi^2 = 22.72$, $P = 1.16 \times 10^{-5}$, 2 degrees of freedom).

The density of RARE elements is low compared to that of the SRE, with an overall mean of 3.5. The variation of RARE density is also low, with an overall V/m ratio of 0.47 ($P = 0.024$), suggesting that RARE have a lower turnover rate than SRE. No evidence of heterogeneity has been found in this data set, which is consistent with the ancestral function of RARE in Hox gene regulation.

The estimated half-life for SREs is similar, about 10 Myr based on ML estimates (PRE: $T_{1/2} = 11.7$ Myr; ERE: $T_{1/2} = 7.23$ Myr). We choose the primate clade to do a BOE estimate for ERE, which yields a half-life for ERE of 22.7 Myr. Within the mammals, the PRE seem to be in equilibrium in even the most recent clade with at least four species, i.e. tamarin, macaque, chimp, and human ($V/m = 1.6$, $P = 0.19$). This clade has an estimated age of 43 Myr and thus the BOE estimate of half-life time is likely to be less than 10 Myr, i.e. this clade would have an RCA of 4 or higher, based on the simulation results with four taxa. This RCA is consistent with the ML estimate of 7.23 Myr.

The half-life estimates for SREs are of the same order of magnitude as those for TFBSs in yeast (20 to 80 Myr), but situated ,ore to the lower end of the distribution. As expected from the V/m ratios, the half-life time of the RARE is estimated to be longer than that of SRE, $T_{1/2} = 147$ Myr, which is one order of magnitude higher than for SRE. This probably reflects stronger selection against changes in RARE elements because of their central role in vertebrate development. A BOE estimate using the data from the eutherian clade yields $T_{1/2} = 289$ Myr, which is a factor two higher than the ML estimate but still in the same order of magnitude.

The origination rates for the SREs are 0.005 binding sites/(kb $\times$ Myr) for PRE and and 0.008 binding sites/(kb $\times$ Myr) for ERE. In contrast, the origination rate for RARE is only 0.0001 binding sites/(kb $\times$ Myr). There is a general negative relationship between origination rates and half-life times such that the shorter the half-life time, the higher the origination rate

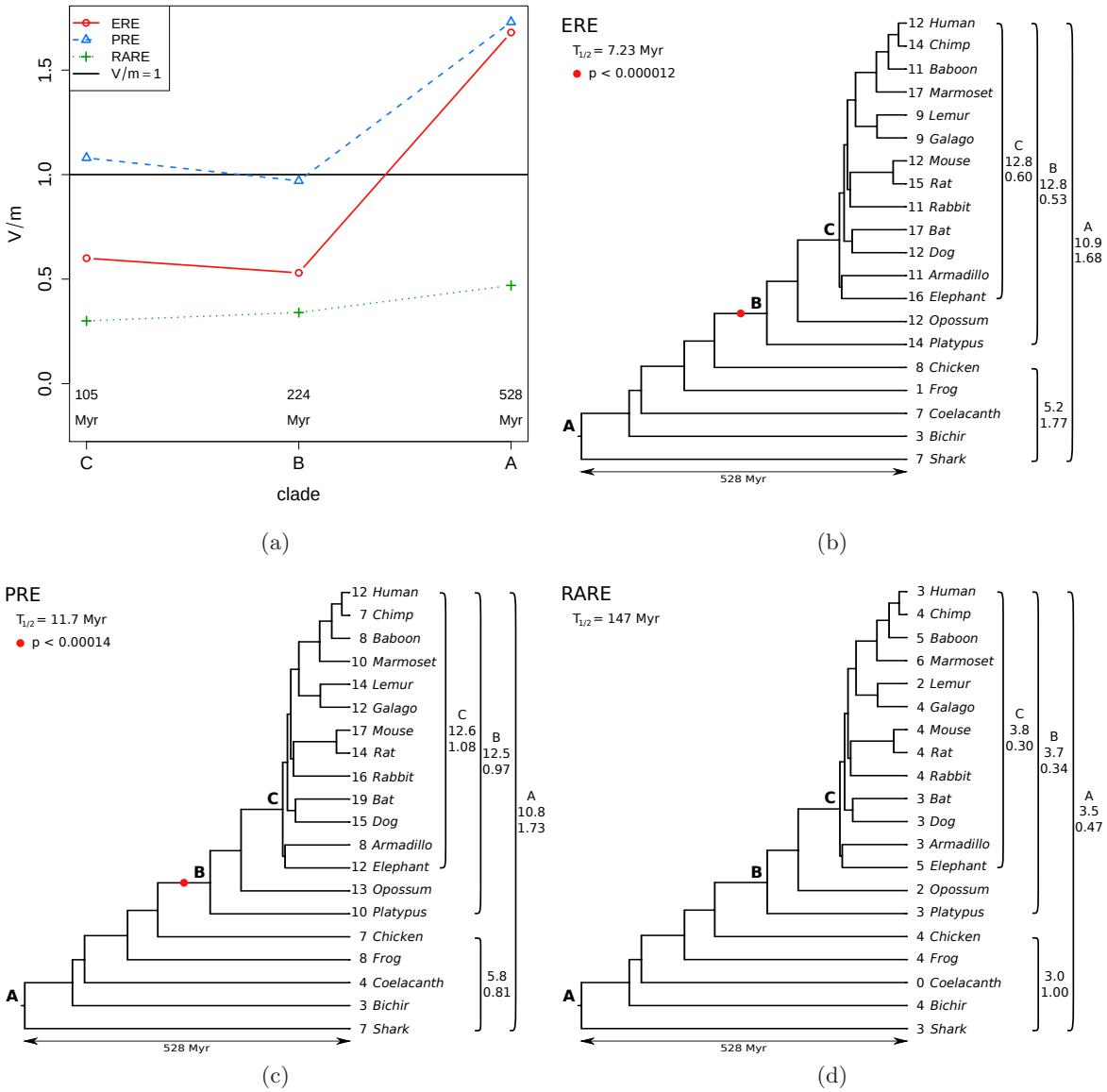


Figure 5.12: Evolution of binding site numbers in the HoxA clusters of vertebrates. The first diagram (a) shows the relationship of the clade age and the V/m ratio of binding sites. The binding site numbers of ERE (b) and PRE (c) are significantly overdispersed caused by differences between the mammalian and non-mammalian taxa. The mammalian clade shows no evidence of heterogeneity (V/m of ERE is 0.53, V/m of PRE is 0.97), which suggests a twofold increase in equilibrium density from about 5 to 10. The P-values at the stem lineage of the mammalian clade are those of the likelihood ratio test, showing that the mammalian and the non-mammalian lineages have different turnover rates and equilibrium densities of ERE and PRE. RARE (d) show a very low density and variation with a V/m ratio of 0.47 suggesting a low turnover rate. There is also no evidence for heterogeneity which is consistent with the ancestral and conserved function of RARE.

(see also the results for yeast-binding sites). Such a negative relationship can be explained if we assume that binding site turnover is due to accidental fixation of slightly deleterious mutations and the selection of compensatory mutations. In equilibrium then, the origination rate is driven by the accidental loss rate, which in turn is determined by the population size and the intensity of stabilizing selection on the binding sites. The stronger the selection, the lower is the fixation probability of deleterious mutations and, as a consequence, the smaller is the need for the fixation of compensatory mutations.

5.3.3 Biological Implications

The exploratory data analysis using this model is based on the prediction that in stochastic equilibrium, the ratio of variance to mean V/m is equal to 1. The data analyzed here is qualitatively consistent with this prediction as younger clades tend to have V/m ratios smaller than 1 and older clades tend to have V/m ratios around 1. In the case of $V/m > 1$, the model predicts that there should be heterogeneity in binding site density among lineages. This is often the case in the data sets we analyzed as clear instances of heterogeneity of mean binding site density are identifiable in clades with $V/m > 1$ (see Figures 5.10 (p.114) and 5.12 (p.119)). Likelihood ratio tests for heterogeneity in the rate of binding site turnover confirm this inference. Hence, comparing V/m ratios between clades can be used as a useful heuristic to identify homogenous binding site dynamics and infer variation in selective constraints acting on TFBSs.

The ML method described here is most useful for testing for heterogeneity in the turnover rate and thus testing for differences in the selective constraints on TFBSs. For example, we found that PRE and ERE densities in HoxA clusters are heterogeneous among gnathostomes, comparison of V/m ratios, and mean binding site densities indicate that the heterogeneity is between mammals and the other gnathostomes. The ERE and PRE densities in mammals are about twice that observed in non-mammalian. A likelihood ratio test indicated that these differences are highly significant (PRE: $P = 1.4 \times 10^{-4}$; ERE: $P = 1.2 \times 10^{-5}$), suggesting a greater involvement of HoxA genes in female reproductive function in mammals than in non-mammalian gnathostomes. Indeed, in humans and other placental mammals, HoxA-13, HoxA-11, HoxA-10, and HoxA-9 have been shown to be involved in female fertility and development of the reproductive tract and mammary gland and directly responsive to hormone signaling (Daftary and Taylor, 2006). Unexpectedly, ERE and PRE density increased in the stem lineage of mammals rather than in the therian or eutherian stem lineage coincident with the evolution of internal development and placentation, respectively. A possible explanation of the early increase SREs is the involvement of HoxA-9 and HoxA-5 in mammary gland development and function (Chen and Capecchi, 1999), which evolved in the stem lineage of mammals. This apparent increase in involvement of steroids in regulating Hox genes might have been preadaptive (exaptive) for the later evolving role of HoxA-11 in placental function

(Lynch *et al.*, 2008). In contrast, there was no heterogeneity in RARE density among the gnathostome taxa sampled, which is consistent with the ancient and conserved role of retinoid acid in Hox gene regulation.

In our yeast data set, heterogeneity in binding site density is most often associated with a decrease in density in the clade including *Candida albicans*. We find this reduced density for CBF1- and MET30/31-binding sites, see clade F in Figure 5.10 (c) and (e), suggesting that these TFs play a diminished role in regulating the methionine pathway genes in these species. A similar pattern applies to GCN4-binding sites in the clade including *Kluyveromyces lactis*, see clade E in Figure 5.10 (d). It would be interesting to test for the function of these TFs in methionine biosynthesis in these species.

Estimates of the model parameters are imprecise in our data, at least as judged by CIs derived from the likelihood functions, suggesting that dense taxon sampling is required for more precise parameter estimation. Indeed, our data indicate that larger taxon sampling in younger clades is necessary to obtain more accurate estimates. Even with these caveats, however, several general trends in binding site turnover are apparent. For example, the half-life of a "typical" TFBS is between 10 and 100 Myr, whereas more constrained binding sites, such as RAREs in mammalian HoxA clusters, have an estimated half-life time of about 150 Myr. At the lower end of the half-life time, distribution are EREs in the mammalian HoxA clusters with an estimated half-life time of about 7 Myr. This indicates that the optimal taxon sampling and phylogenetic depth needed for accurate inferences in binding site density dynamics is variable, such that more constrained sites require deeper taxon sampling, whereas binding sites with higher turnover rates require dense sampling of younger clades. Based on our simulation results, we recommend that studies of binding site dynamics should include at least six species, see Figure 5.6, although the choice of taxon sampling depends heavily on half-life of the binding site in that clade. In studies of genomic regions from species that are closely related, one has the benefit that the number of orthologous binding sites could be determined by sequence alignment, which will further increase the accuracy of the parameter estimates, see Section 4.2.2 (p.90).

Overall, our results show that an ML implementation of the stochastic binding site turnover model derived in Chapter 4 (p.77) allows for the analysis of binding site density in an explicit phylogenetic context. The model can be used to statistically test for relative differences in the turnover rate and equilibrium density of TFBSs as well as gain insights into the actual rate of binding site turnover. The development of this and other models of TFBS evolution has the potential to reveal the rate and pattern of CRE evolution similar to the development of codon-based models of protein evolution.

CHAPTER 6

Conclusion

The Universe is full of ignorance all around and the scientist panned through it like a prospector crouched over a mountain stream, looking for the gold of knowledge among the gravel of unreason, the sand of uncertainty and the little whiskery eight-legged swimming things of superstition.

Occasionally he would straighten up and say things like "Hurrah, I've discovered Boyle's Third Law." And everyone knew where they stood. But the trouble was that ignorance became more interesting, especially big fascinating ignorance about huge and important things like matter and creation, and people stopped patiently building their little houses of rational sticks in the chaos of the universe and started getting interested in the chaos itself—partly because it was a lot easier to be an expert on chaos, but mostly because it made really good patterns that you could put on a t-shirt.

And instead of getting on with proper science, like finding the bloody butterfly whose flapping wings cause all these storms we've having lately and getting it to stop, scientist suddenly went around saying how impossible it was to know anything, and that there wasn't really anything about, and how all this was tremendously exciting, and incidentally did you know there were possibly all these little universes all over the place but no one can see them because they are all curved in on themselves? Incidentally, don't you think this is a rather good t-shirt?

Witches Abroad
Terry Pratchett

In this thesis, we discussed two important aspects of computational analysis of transcriptional regulatory elements: the detection of them based on homology and the examination of their evolutionary history based on a binding site turnover model.

In the first part of the thesis, we formulated the multiple sequence alignment problem as an optimization problem of selecting a maximum collection of consistent local alignments. Based on a detailed theoretical scaffold, we developed a simple but effective heuristic for assembling local pairwise sequence alignments into a local multiple alignment. We then applied the algorithm to artificial and biological data sets. In both cases, we were able to demonstrate the capabilities of our algorithm to solve the problem of finding maximal consistent alignment subsets. Furthermore, we demonstrated that our approach can be used for the computation of multiple alignments based on individual alignment edges. The quality of these alignments is comparable to other state-of-the-art multiple alignment tools.

The motivation for developing this algorithm was to be able to detect phylogenetic footprints based on pairwise local alignments using consistency to detect homologous similarities. In order to perform this task, we developed the `Tracker` algorithm that computes for a given set of sequences an initial collection of local, pairwise alignments and determines on the basis of this data consistent alignment sets. The multiple alignment that is created during the computation of consistent subsets contains all conserved motifs that are consistent to the alignment subset. We then applied `Tracker` to a set of homologous sequences and demonstrated its capability to detect phylogenetic footprints.

In the second part of the thesis, we proposed a simple, but mathematically non-trivial, phenomenological model for binding site number evolution at a genomic locus. The model is based on the assumption that binding sites originate at a constant rate typical of a certain genomic region and have a constant decay rate per binding site. We provided an elementary derivation of the transient probability distribution and compared it to the transient frequency distribution from simulations of a sequence evolution model. The results showed that the phenomenological model fits the results of the sequence evolution model very closely, suggesting that the phenomenological model could be a fair representation of binding site turnover, even though the model is not based on an explicit consideration of sequence evolution. We then applied the model to data concerning the number of estrogen response elements in mammalian HoxA clusters. We showed that this data is consistent with the assumption of a stationary turnover process within the mammals and that we can estimate the phenomenological parameters.

The motivation for developing a phenomenological model of binding site number dynamics is to be able to detect changes in selective constraints acting on transcription factor binding sites (TFBSs). Therefore, we explored the utility of a phenomenological mathematical model of binding site turnover for the analysis of the evolution of cis-regulatory elements (CREs).

Specifically, we wanted to determine whether this model could guide investigators in identifying clades where the selective forces acting on binding sites in CREs are different. We assume that, in the case of poorly conserved but functionally important TFBSs, selection primarily affects the rate of TFBS turnover and thus leads to differences in average binding site density between lineages. We showed that the predictions of the model can be used in two ways. On the one hand, the model can be used for exploratory "back of the envelope" data analysis. On the other hand, our maximum likelihood implementation `Creto` of the model can be used to estimate model parameters and, in likelihood ratio tests, for detecting differences in turnover rates between clades. The latter test is a key tool for identifying functionally important changes in the evolution of cis-regulation and we demonstrated the abilities of the approach on biological data.

Our `Tracker` approach for detecting phylogenetic footprints computes a new form of multiple alignments that consist of local motifs but still satisfy the order conditions. The representation as thick alignment columns is an intuitive way to illustrate homologous motifs that differ by evolutionary events like insertions or deletions. The computation of the initial alignment sets as well as the computation of other alignment steps is completely generic and can be easily adopted to new alignment algorithms. The replacement of the relative old `LASTZ` and `ClustalW2` alignment programs that we used for the development of the method by modern approaches is one of the next tasks. Also the inclusion of filter methods that allow the search for specific motif patterns is important. A further possibility to improve the results of the algorithm could be the usage of phylogenetic information. Alignments between far related sequences are more likely to be based on random similarities, if they are not supported by alignments between more related sequences We could detect false positive alignments this way.

The statistical analysis of binding site density data in order to identify potentially important changes in the selective constraints acting on CREs is also a completely new approach. In contrast to other methods, it does not depend on conserved cis-regulatory sequences and works even with data where turnover changed the location and arrangement of binding sites. The model allows to make predictions for deviations from a constant rate of turnover that can be tested with rigorous statistical methods. In that way `Creto` is meant to set the stage for a sequence analysis tool, similar to the neutral sequence evolution model used to detect natural selection in coding regions. It would certainly be interesting to use the present model to search for two kinds of deviations from the model predictions. One is heterogeneity in the rate of origination and/or decay between different lineages or clades. That could be done by comparing the likelihood of the observed distribution of binding site numbers on the tips of a phylogenetic tree assuming constant rates of origination and decay with a model which allows for different parameters in different parts of the phylogenetic tree. A significant difference in likelihood would constitute evidence for a difference in the selective constraints on binding

sites in different groups of organisms. Another possible application of the model would be to provide evidence for binding sites with different turnover rates. For instance, it is likely that a non-coding region contains binding sites with strong selective constraints and others with weaker constraints. This could be detected by testing for deviations from the exponential loss of homologous binding sites as a function of time since lineage separation.

In summary, both parts of this thesis present new approaches for well known problems concerning the detection and evolutionary analysis of regulatory elements. The software tools `Tracker` and `Creto` were created in order to perform these tasks and is was shown that they perform well. Both programs are freely available for usage and further development. Hopefully, they shed more light on the world of transcriptional regulation.

APPENDIX A

Supplementary Data

A.1 Simulation of Sequence Evolution

The simulation of sequence evolution in Section 4.2.1 (p.86) was performed with different parameters for the mutations and the motifs. The corresponding results are shown in the following figures. In detail we used following parameters:

(a) The motif length l is 5 and the mutation rate m is 10^{-5}. All mutations are equally probable.

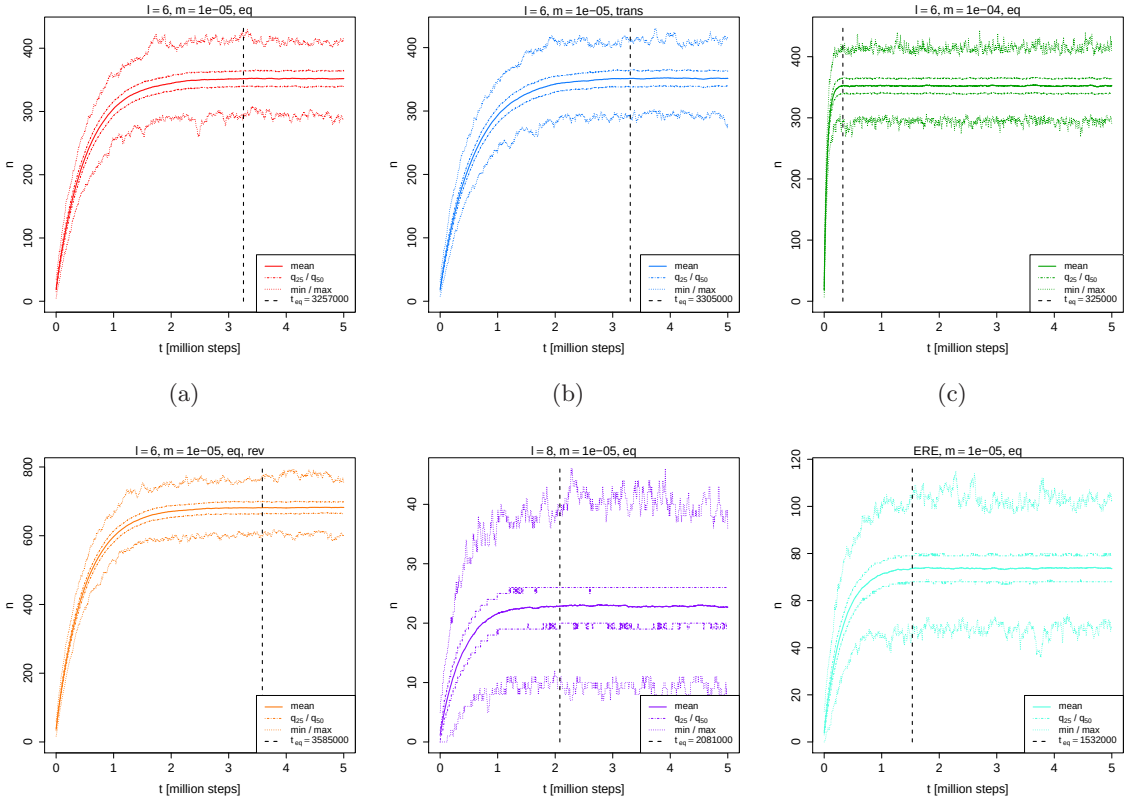
(b) The motif length l is 5 and the mutation rate m is 10^{-5}. The transition/transversion rate ratio for mutations is 2/1.

(c) The motif length l is 5 and the mutation rate is high with $m = 10^{-4}$. All mutations are equally probable.

(d) The motif length l is 5 and the mutation rate m is 10^{-5}. All mutations are equally probable but the motifs are count on both strands.

(e) The motif length is higher with $l = 8$ and the mutation rate m is 10^{-5}. All mutations are equally probable.

(f) The motifs are determined with position weight matrix for the estrogen response element. The mutation rate m is 10^{-5} and all mutations are equally probable.



(a) (b) (c)

(d) (e) (f)

Figure A.1: Spreading of simulated binding site numbers over time.

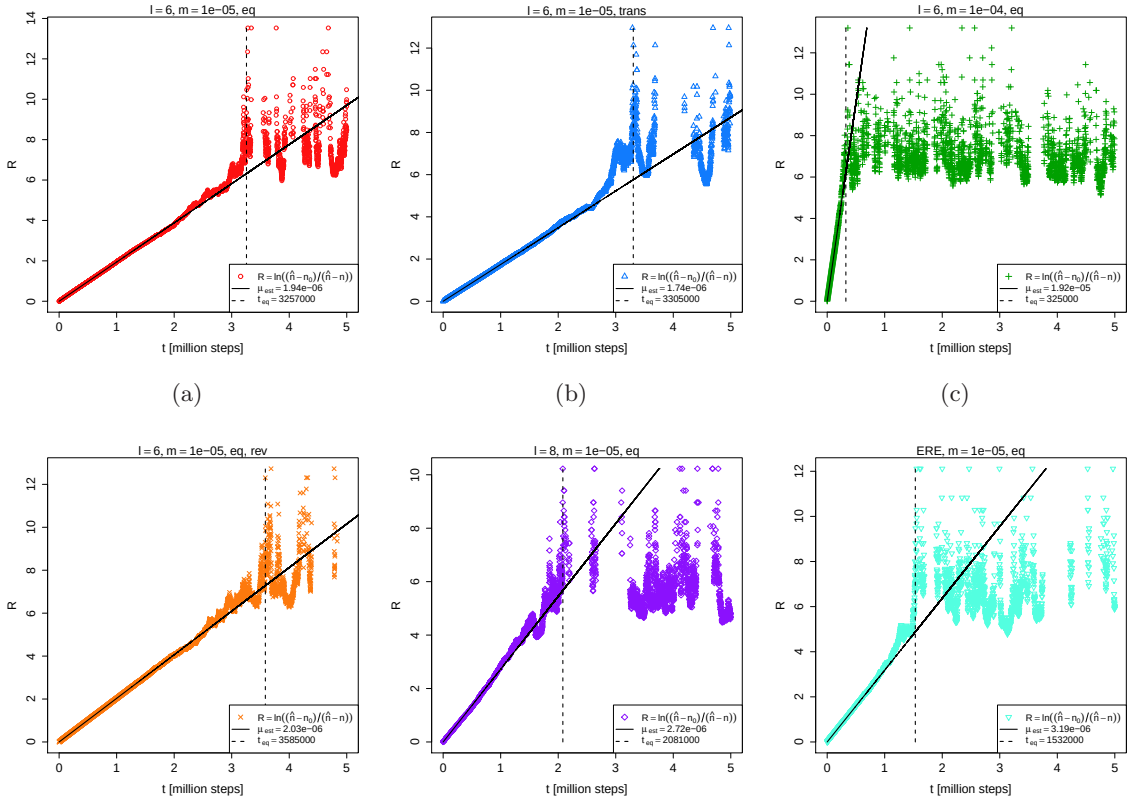


Figure A.2: Linear relationship (solid black line) between time t and the regression variable $R = \ln((\hat{n} - n_0)/(\hat{n} - n))$ in the transient section of the development of the mean binding site number.

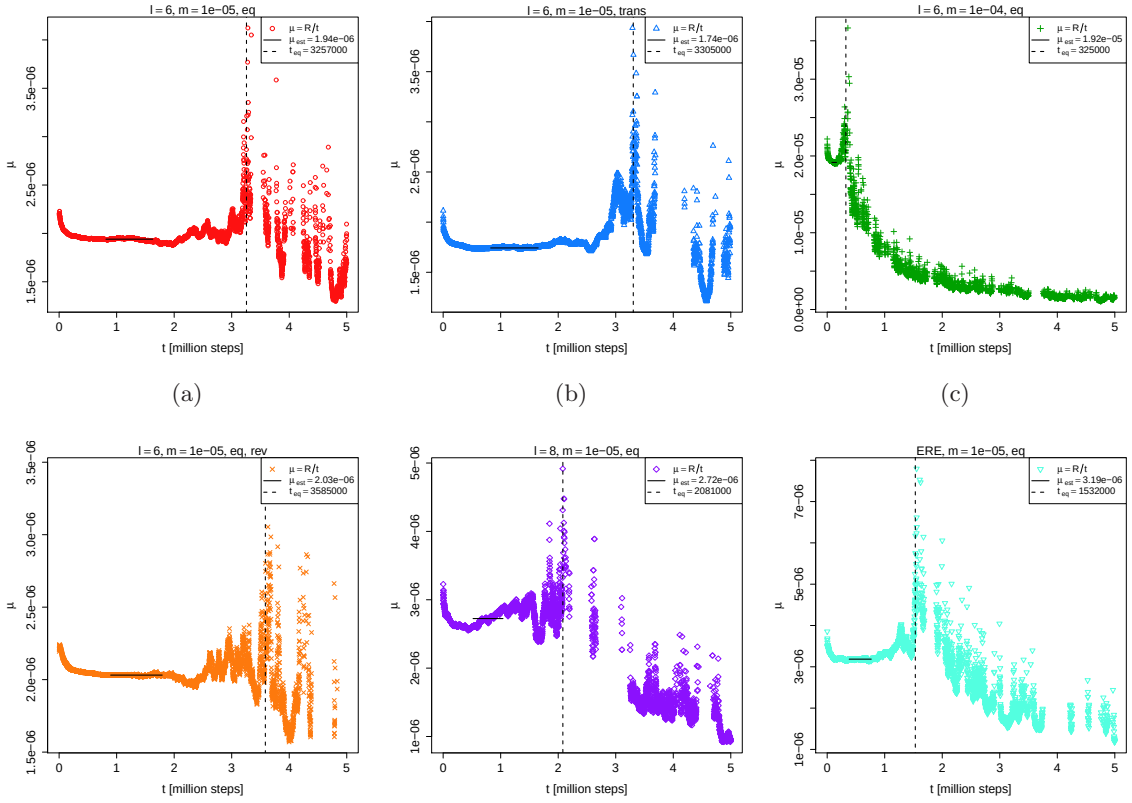
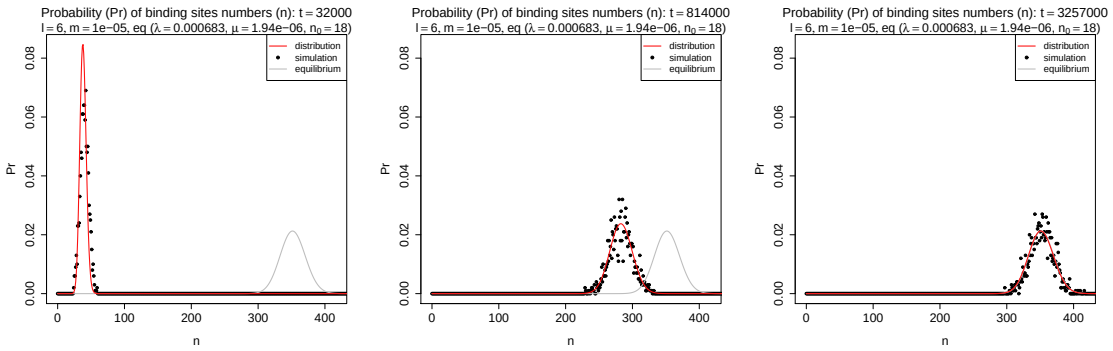
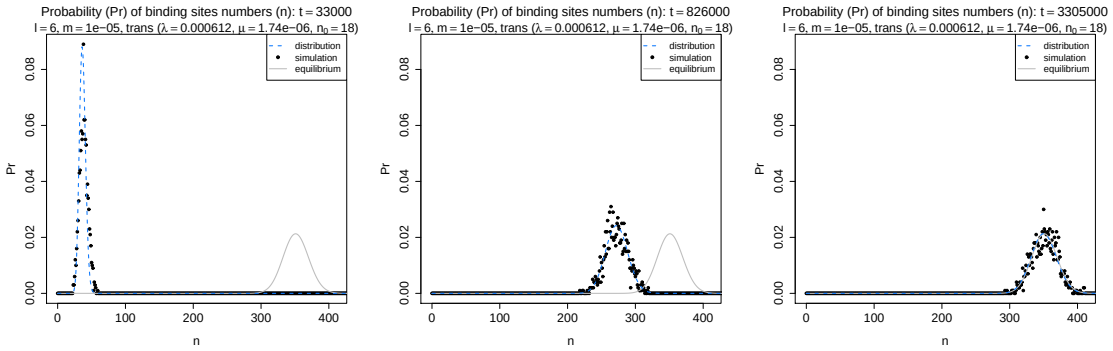


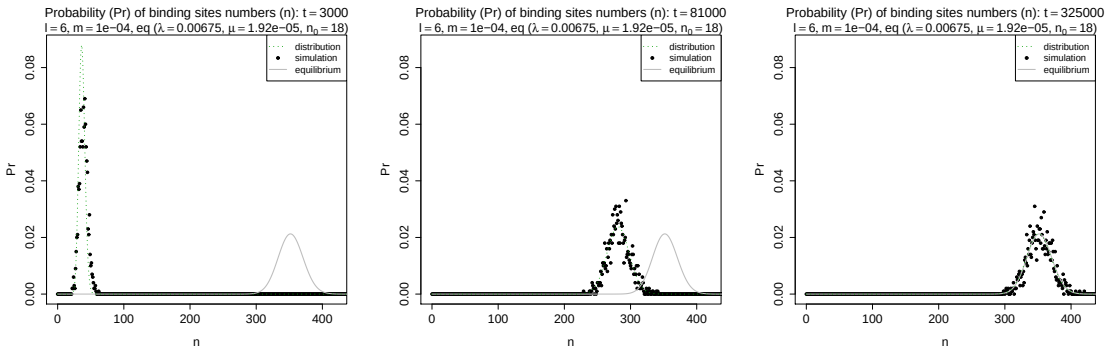
Figure A.3: Estimating the effective decay rate μ of binding sites as a function of the time. The solid black line indicates the area that is used for the estimating of the value of μ.



(a)

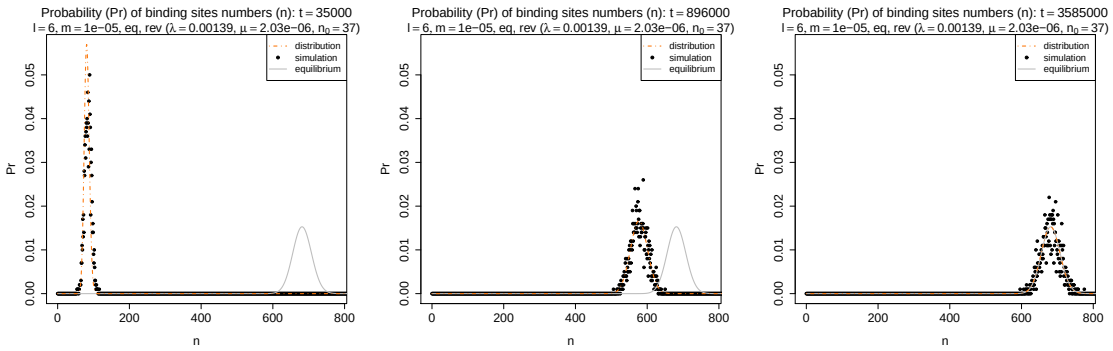


(b)

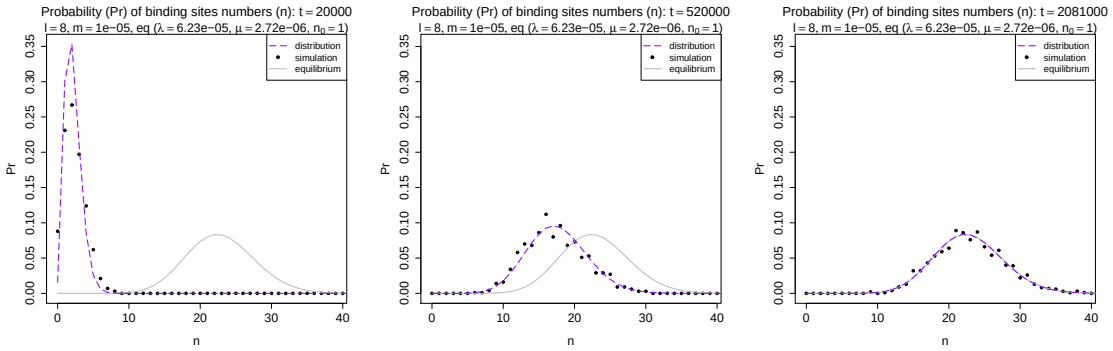


(c)

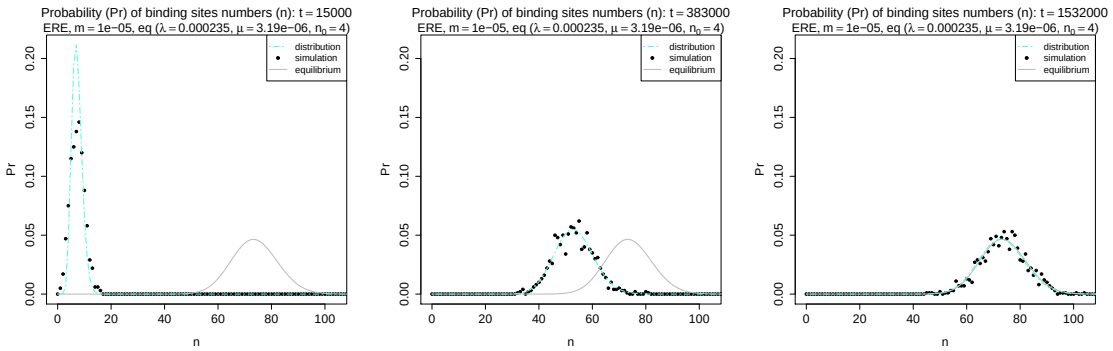
Figure A.4: Comparison of Simulation and Analytical Prediction (A).



(d)



(e)



(f)

Figure A.4: Comparison of Simulation and Analytical Prediction (B). The transient ensemble distribution of binding sites as found by simulation (black points) resembles the analytical prediction (colored line). In equilibrium, the distribution corresponds to the Poisson distribution (gray line).

A.2 Vertebrate Data Set

Species	Scientific Name	Reference
Human	*Homo sapiens*	NT086366
Chimp	*Pan troglodytes*	NT107590
Baboon	*Papio anubis*	NT108362
Marmoset	*Callithrix jacchus*	NT113347
Lemur	*Microcebus murinus*	NT165887
Galago	*Otolemur garnettii*	DP000935
Mouse	*Microcebus murinus*	NT165680
Rat	*Rattus norvegicus*	NT107474
Rabbit	*Oryctolagus cuniculus*	DP001063
Bat	*Rhinolophus ferrumequinum*	DP000727
Dog	*Canis familiaris*	NT107854, NT165635
Armadillo	*Dasypus novemcinctus*	NT108274
Elephant	*Loxodonda africana*	NT161952
Opossum	*Monodelphis domestica*	NT113108, NT112848
Platypus	*Ornithorhynchus anatinus*	NT165289
Chicken	*Gallus gallus*	NT107524
Frog	*Xenopus tropicalis*	AC145789
Coelacanth	*Latimeria menadoensis*	FJ497005
Bichir	*Polypterus senegalus*	AC126321, AC132195
Shark	*Heterodontus francisci*	AF224262, AF224263

Table A.1: NCBI references for the sequences used to determine the nuclear receptor response elements in the vertebrate HoxA clusters.

APPENDIX B

The Tracker Program

B.1 Availability and Installation

The source code of `Tracker` is available at:

- `http://www.bioinf.uni-leipzig.de/Software/tracker2/`

It is written in ANSI C++ and does not need any additional libraries. For installation, copy the downloaded file to an arbitrary directory, extracted the source code and compiled it as follows:

- `tar -xzf tracker2.src.tgz` (extraction)
- `cd tracker` (change into directory)
- `make` (compilation)

An example run can be performed by typing:

- `./tracker example/csb/csb.fa`

After the calculation, the results can be found in the file `results.txt`.

B.2 Input Format

In general, the input of `Tracker` involves three hierarchical file types. The program itself is called with the main file including the sequence table. This table contains in each row an entry for one of the homologous sequences that should be compared. Each row is composed of three columns, separated by a tab character:

1. The first column contains the *id* of the sequence, i.e. an arbitrary but unique name.

2. The second column contains the absolute or relative location of the *region* file for the corresponding sequence. Region files are the second input level and contain the definitions of the regions as described later.

3. The third column contains the *ignore* list, i.e. a list of comma separated ids referring to other sequences that should not be compared with this sequence. This can, for example, be used if sequences are not expected to contain mutual homologous areas in order to prevent alignments caused by random similarities.

Region files are the second input level and each region file includes the fragment table for one of the homologous sequences. This table contains in each row an entry for one fragment consisting of seven columns separated by a tab character:

1. The first column contain a list of *regions* to which the corresponding fragment belongs, separated by comma. Each region is represented by a unique id and encompasses all fragments of all region files that containing this id in their definition.

2. The second column contains the absolute or relative location of the *fasta* file for the corresponding fragment. Fasta files are the last input level and contain multiple nucleotide sequences given in IUPAC notation. Each entry is labeled with a tag by an extra line in front of the sequence that starts with a '>'.

3. The third column contains the *tag* without the leading '>' of the fasta entry that contains the corresponding fragment.

4. The fourth column contains the position of the *begin* of the fragment in the fasta entry whereas the first nucleotide of the fasta entry is at position 1.

5. The fifth column contains the position of the *end* of the fragment.

6. The sixth column contains the *strand* where the fragment is located. The strand given by the fasta entry is denoted by '+' while the reverse compliment strand is denoted by '−'.

7. In the seventh column, a *name* of the fragment can be defined that is used in the output of `Tracker`. If no name is given, an artificial name is created consisting of the fasta file, the tag and the begin and end positions.

Note that it is possible to split the data for a input sequence into multiple fasta entries. In this case, `Tracker` determines the order and the orientation of this entries during the assembly of the multiple alignments.

Additionally to the given input scheme, the input can consist of only one multiple fasta file. In this case, only one region exists and each fasta entry corresponds to one homologous sequence where the fragment for the comparison involves the whole sequence. In other words, each entry in the fasta file is compared to each other.

Furthermore, `Tracker` can be used to determine only consistent alignment subsets. In this case the input can also consist of a single file including an alignment table. This table contain in each row an entry for one pairwise, local alignment and each row is composed of seven columns, separated by a tab character:

1. The first column contains the *id* of the first sequence.
2. The second column contains the *begin* of the first sequence.
3. The third column contains the *end* of the first sequence.
4. The fourth column contains the *id* of the second sequence.
5. The fifth column contains the *begin* of the second sequence.
6. The sixth column contains the *end* of the second sequence.
7. The seventh column contains the *score* of the alignment.

In addition to this description, the distribution contains in the `examples/` sub-directory an example for each of the possible input definitions.

B.3 Parameters

The parameters that are supported by `Tracker` are summarized in Table B.1.

B.4 Output Format

By default, `Tracker` writes all results into the plain text file `results.txt`. If the program is used for the determination of consistent alignment subsets, the file contains all consistent subsets determined by the heuristic. Each solution is described by a single line containing the index of the solution starting with 0 followed by the list of comma separated alignment indices. Alignment indices corresponds to the order of the input alignments whereas the first alignment in the input file has index 0. The results are sorted by the sum of the alignment scores in each solution, i.e. the first column contains the highest scoring solution. For example, see Figure B.1 (p.139).

If `Tracker` is used for the detection of homologous regions, the file contains all results in a hierarchic manner. After general information about the program call and the used parameters,

short	long	type	description
-l	--lastz	string	Options used for calculation of initial alignment set with LASTZ. Default is --seed=half8 --ambiguous=iupac --strand=both --hspthresh=1000.
-s	--strand	flag	Additional flag that causes LASTZ to align only the given strands of the fragments.
-p	--process	flag	Remove unconserved and repetitive parts out of initial alignments. This option improves the quality of the initial alignments. It includes the calculation of exact alignments between segment pairs by ClustalW2 and can take some time for big data sets.
-t	--tolerance	int	Error tolerance for the determination of consistent subsets and the assembly of the multiple alignment. If no value is given, the highest difference between the length of segment pairs is used as indicator for the maximal number of gaps in the initial alignment set and we use this value as allowed tolerance.
-e	--edges	flag	If set, the each alignment of the initial set is broken down to single edges and all edges of all alignments are used as new initial alignment set for the assembly of the multiple alignment. Note that this includes the calculation of exact alignments between segment pairs by ClustalW2, if not already calculated by the pre-processing, and can take some time for big data sets. Also this parameter sets the allowed tolerance to 0.
-i	--indices	flag	If set, Tracker determines only the consistent alignment subsets without calculating the multiple alignment. This parameter is set per default if the input consist of an alignment table.
-h	--html	string	If given, Tracker creates in addition to the text output also a set of html files in the directory given by the parameter. These files contain the results in a more user friendly representation with colors and overview figures.

Table B.1: Parameters supported by the Tracker algorithm.

see Figure B.2 (p.139), the file contains for each result an entry with meta information for the solution, see Figure B.3 (p.140). The order of the results in the file corresponds to the alignment scores. Each result entry is followed by all included columns that describes the homologous motifs, see Figure B.4 (p.141).

The optional HTML output contains in principle the same data, except that it is possible to restrict the shown results to one of the defined regions. Additionally, tabulated data with links simplify the navigation between solutions and columns, the use of nucleotide specific colors for the alignments make it more easy to recognize patterns and graphics give an overview about the relative locations of motifs, see Figure B.5 (p.141).

```
0: 2, 3, 5, 6
1: 0, 1, 2, 4
```

Figure B.1: `Tracker` output for the determination of consistent subsets. The two lines tell us that the heuristic found two consistent subsets. The best scoring solution consists of the alignments 2, 3, 5 and 6 given by the input file, while the second best set consists of the alignments 0, 1, 2 and 4.

```
<DATA>
  call:       ../../tracker csb.fa -a alignments.txt
  version:    v2.6 (2011/04/08 20:03:29)
  time:       2011/05/29 00:17:25
  sequences:  6
  alignments: 1976
  clusters:   1
  cliques:    69
  columns:    50351

<OPTIONS>
  lastz:      --seed=half8 --ambiguous=iupac
              --strand=both --hspthresh=1000
  strand:     0
  process:    1
  tolerance:
  edges:      0
  indices:    0
  html:
```

Figure B.2: `Tracker` output with information about program call and used parameters.

```
<RESULT>
  index:             0
  tolerance:         10
  alignment scores: 1828952
  extended scores:  8957942
  column scores:    188078
  column lengths:   4131
  alignments:       3128
  regions:           csb
  sequences:
    0: 0|+|20-4252|4233
    1: 0|+|0-3240|3241
    2: 0|+|10-3260|3251
    3: 0|+|58-3214|3157
    4: 0|+|187-4002|3816
    5: 0|+|0-2831|2832
```

Figure B.3: Tracker output of results. For each solution, the file contains the index, the used tolerance, the sum of the alignment scores, the sum of the extended scores, the sum of the column scores, the sum of all column lengths, the number of alignments and a list of regions as well as the sequence areas that are covered by the result. For each sequence index, the index of the fasta entry, the strand, the begin and the end of the area as well as the size of the area are given.

```
<MOTIF>
  index:         81
  clique:        0
  species:       6
  seq. length:   11
  algn. length:  12
  score:         361
  alignments:    7
  motif:
    0: 0|+|3542-3552|11   79 (0)   [AGGACACTTGC]   (0) 82
    1: 0|+|2368-2378|11   79 (0)   [GAACAATTTGC]   (0) 82
    2: 0|+|2368-2378|11   79 (0)   [GAACAATTTGC]   (0) 82
    3: 0|+|2423-2432|10   79 (0)   [AACAAAATTG-]   (14) 85
    4: 0|+|3363-3373|11   80 (0)   [AGGACATTTCC]   (0) 82
    5: 0|+|2132-2142|11   80 (0)   [AGGAAATTTCC]   (0) 82
                                   [     * **  ]
```

Figure B.4: `Tracker` output of motifs. Representation of a motif by a thick alignment column. Besides meta information like indices and scores, each motif entry contains a global alignment of a thick column. The first part of each alignment line contains the sequence index followed by the index of the corresponding fasta file entry, the strand, the begin and end position of the interval and the size of the interval. This is followed by two numbers where the first gives the index of the column that contain the predecessor of the corresponding interval and the second gives the distance between this intervals. Behind the subsequent aligned sequence, the same data is also given for the successor interval.

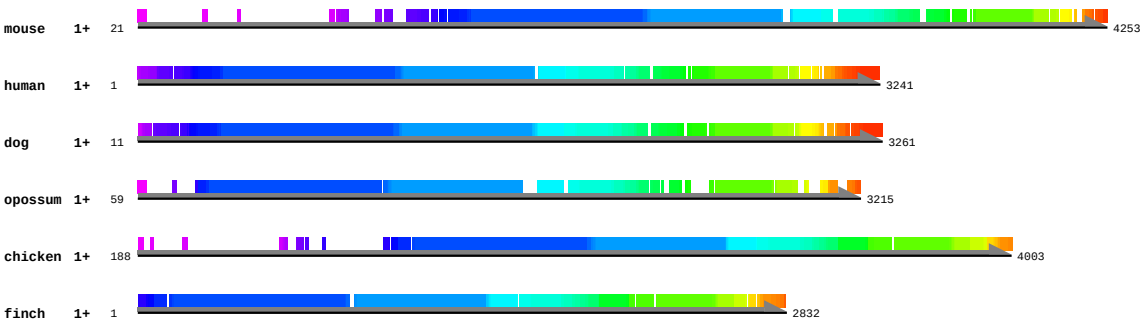


Figure B.5: Graphical representation of a `Tracker` result. The location of motifs relative to the sequences are indicated by colored bars whereas intervals of the same motif have the same color.

APPENDIX C

The Creto Program

C.1 Availability and Installation

The source code of Creto is available at:

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

It is written in ANSI C++ and does not need any additional libraries. For installation, copy the downloaded file to an arbitrary directory, extracted the source code and compiled it as follows:

- tar -xzf creto.src.tgz (extraction)
- cd creto (change into directory)
- make (compilation)

An example run can be performed by typing:

- ./creto example/primate_HoxA_ERE.txt

After the calculation, the results can be found in the file results.txt.

C.2 Input Format

Creto parses the phylogenetic data needed for the computation out of an input file. The data in the file has to be marked with tags and the algorithm parses everything between the line behind the tag and the line in front of the next tag. Thereby, empty lines are allowed and '#' can be used for comments. Following tags are supported:

<**COMMENTS**> (optional) The text behind this tag is printed in result files. This is useful for the transfer of meta information from the input file to the result file.

<**TREE**> (required) The text behind this tag is interpreted as the phylogenetic tree with binding site numbers of terminal taxa in extended Newick grammar:

```
<tree>        ::= <subtree>";"
<subtree>     ::= <leaf> | <internal>
<leaf>        ::= <description>
<internal>    ::= "("<branchlist>")"<description>
<branchlist>  ::= <branch> | <branch>","<branchlist>
<branch>      ::= <subtree><distance>
<description> ::= <empty> | <name> | <name>"|"<bsnumber>
<name>        ::= <empty> | <string>
<bsnumber>   ::= <empty> | <int>
<distance>    ::= <empty> | ":"<double>
```

<**SCALE**> (optional) The text behind this tag is interpreted as double value and is multiplied with the distances in the tree.

<**START**> (optional) The text behind this tag is interpreted as the assumed number of binding sites for the root node. If this tag is not used, the mean of all leaf binding site numbers is used.

<**ROOT**> (optional) The text behind this tag is interpreted as the name of the root note, i.e. `Creto` determines the parameters for the subtree that corresponds to the given name of a node in the tree. Node names have the following syntax:

```
<name> ::= <leaf> | "("<name>"-"<name>")"
```

<**ALTERNATIVES**> (optional) The text behind this tag is interpreted as list of node names. For each subtrees defined by the corresponding name, `Creto` will compute determines alternative turnover rates. Node names have the following syntax:

```
<name> ::= <leaf> | "("<name>"-"<name>")"
```

In addition to this description, the distribution contains in the `examples/` sub-directory an example for a possible input definition.

C.3 Parameters

The parameters that are supported by `Creto` are summarized in Table C.1.

C.4 Output Format

During the optimization, `Creto` displays the actual turnover rates together with the corresponding likelihood. After the optimization, the program creates the plain text file `results.txt` and the postscript file `results.eps`. The text file contains an overview of the input (see Figure C.1 (p.146)) and used parameters (see Figure C.2 (p.146)), the initialization values together with characteristics of the given data like mean binding site numbers, variance of the binding site numbers, ages of

short	long	type	description
`-a`	`--alternatives`	int	Detect this number of subtrees that are most likely to have alternative rates and determine this rates. Default: 0
`-e`	`--equilibrium`	flag	Optimize root parameters under the condition that they are in the equilibrium state.
`-l`	`--lambda`	flag	Optimize only lambda and use root mu for all alternative nodes.
`-m`	`--mu`	flag	Optimize only mu and use root lambda for all alternative nodes.
	`--bound`	float	Fraction of the binding site number density on both boundaries that is not considered by the the calculations. Default: 1e-6
	`--decay`	float	Fraction of the binding site numbers that are exists continuously since root (for the determination of the optimization start values). Default: 0.5
`-d`	`--delta`	int	Limit number of rate value bisections in each improvement step to given value. Default: 20
`-r`	`--repeats`	int	Limit rounds of parameter optimizations to given value
	`--density`	float	Set minimal probability density to this value.

Table C.1: Parameters supported by the `Creto` algorithm.

clades, variance/mean ratios and the final results for each defined subtree with alternative rates (see Figure C.3 (p.147)).

The postscript file contains a graphical representation of the data together with the final results (see Figure C.4 (p.148)).

```
tree:
  ((((Human,Chimp),Baboon),Tamarin),(Lemur,Galago)) [a:0.00e+00, bs:11.06]
  |-(((Human,Chimp),Baboon),Tamarin)                [a:3.07e+07, bs:13.27]
  | |-((Human,Chimp),Baboon)                        [a:4.91e+07, bs:12.16]
  | | |-(Human,Chimp)                               [a:6.64e+07, bs:13.08]
  | | | |-Human                                     [a:7.75e+07, bs:12]
  | | | `-Chimp                                     [a:7.75e+07, bs:14]
  | | `-Baboon                                      [a:7.75e+07, bs:11]
  | `-Tamarin                                       [a:7.75e+07, bs:17]
  `-(Lemur,Galago)                                  [a:2.71e+07, bs:9.15]
    |-Lemur                                         [a:7.75e+07, bs:9]
    `-Galago                                        [a:7.75e+07, bs:9]
```

Figure C.1: Creto output for the input tree with age and expected binding site number for each node.

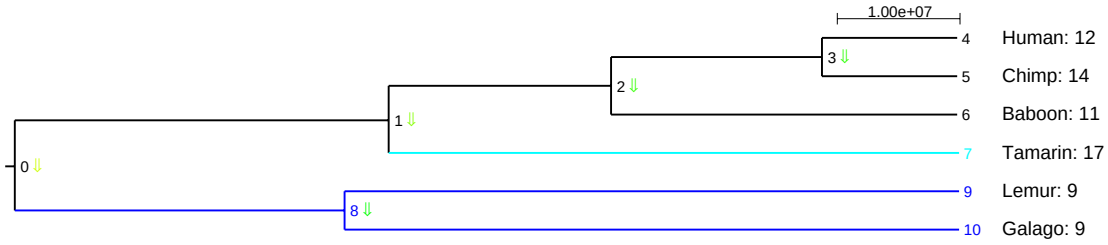
```
parameters:
  tree:          ((((Human|12:0.0035,Chimp|14:0.0035):0.0055,
                    Baboon|11:0.009):0.00582,Tamarin|17:0.01482):0.00971,
                  (Lemur|9:0.01595,Galago|9:0.01595):0.00858);
  scale:         3.1594e+09
  start bs:      12
  n_min:         0
  n_max:         32
  equilibrium:   0
  only lambda:   0
  only mu:       0
  bound:         1e-06
  decay:         0.5
  convergence:   1e-06
  repeats:       inf
  delta:         20
```

Figure C.2: Creto output for the used parameters.

```
results:
  root (id: 0, name: '((((Human,Chimp),Baboon),Tamarin),(Lemur,Galago))',
        mean: 12, Var: 9.6, V/m: 0.8', diff: 0)
    lambda:    6.33607e-08
    mu:        1.96262e-09
    lambda/mu: 32.2837
    min. dens: 0.607467
  alternative* (id: 7, name: 'Tamarin',
                mean: 17, Var: -, V/m: -', diff: 3.5)
    => stopped: lambda/mu -> +inf <=
    lambda:    1.4451e-07 (const)
    mu:        7.60014e-15 (const)
    lambda/mu: 1.90142e+07
    min. dens: 0.00115203
  alternative* (id: 8, name: '(Lemur,Galago)',
                mean: 9, Var: 0, V/m: 0', diff: 3)
    => stopped: rates -> 0 <=
    lambda:    9.59569e-15 (const)
    mu:        4.26475e-15 (const)
    lambda/mu: 2.25
    min. dens: 1
  likelihood:   2.86125e-05
  iterations:   36
  calculations: 2431
```

Figure C.3: `Creto` output for the computed evolutionary rates. For each subtree under consideration, the rates that maximize the likelihood of the tree are given together with meta information like the λ/μ and V/m ratio are given. The maximal likelihood is given at the end together with the number of iterations and likelihood calculations during the optimization.

comments:
binding site number of transcription factor
ERE in the HoxA cluster of primates.



(sub)tree	λ	μ	λ/μ	mean	Var	V/m	Δ	∫
0	6.33607e-08	1.96262e-09	32.28	12.00	9.60	0.80 ⇓	0.00	0.61
7*	1.44510e-07 (∅)	7.60014e-15 (∅)	$\lambda/\mu \to \infty$	17.00	-	-	3.50	0.00
8*	9.59569e-15 (∅)	4.26475e-15 (∅)	$\lambda, \mu \to 0$	9.00	0.00	0.00 ⇓	3.00	1.00
likelihood	2.86125e-05							
iterations	36							
calculations	2431							

Figure C.4: Graphical representation of a Creto result. Subtrees with alternative rates are marked by different colors. Arrows behind node numbers indicate v/m ratios above (red) or below (green) 1.

List of Abbreviations

BOE	back of the envelope
bp	base pairs
CI	confidence interval
CRE	cis-regulatory element
CSB	Conserved Sequence Block B
ERE	estrogen response element
IGR	intergenic region
MCASP	maximal consistent alignment subset problem
miRNA	micro RNA
mRNA	messenger RNA
Myr	million years
ncRNA	non-coding RNA
PDE	partial differential equation
PFM	position frequency matrix
PRE	progesterone response element
PSSM	position specific scoring matrix
PWM	position weight matrix
RARE	retinoic acid response element
RCA	relative clade age
RNA	ribonucleic acid
rRNA	ribosomal RNA
SD	standard deviation
siRNA	small interfering RNA
snoRNA	small nucleolus-restricted RNA
snRNA	small nuclear RNA
SRE	steroid response element
TF	transcription factor
TFBS	transcription factor binding site
tRNA	transfer RNA
UTR	untranslated region

List of Figures

List of Tables

Bibliography

Abdeddaïm, S. (1997). On incremental computation of transitive closure and greedy alignment. In A. Apostolico and J. Hein, editors, *Combinatorial Pattern Matching*, volume 1264 of *Lecture Notes in Computer Science*, pages 167–179. Springer Berlin / Heidelberg.

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002). *Molecular Biology of the Cell*. Garland Science.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*, **25**(17), 3389–3402.

Ameyar, M., Wisniewska, M., and Weitzman, J. B. (2003). A role for ap-1 in apoptosis: the case for and against. *Biochimie*, **85**(8), 747–752.

Aparicio, S., Morrison, A., Gould, A., Gilthorpe, J., Chaudhuri, C., Rigby, P., Krumlauf, R., and Brenner, S. (1995). Detecting conserved regulatory elements with the model genome of the japanese puffer fish, fugu rubripes. *Proc Natl Acad Sci U S A*, **92**(5), 1684–1688.

Avery, O., MacLeod, C. M., and McCarty, M. (1944). Induction of transformation by a desoxyribonucleic acid fraction isolated from pseudomococcus tye iii.

Badis, G., Berger, M. F., Philippakis, A. A., Talukder, S., Gehrke, A. R., Jaeger, S. A., Chan, E. T., Metzler, G., Vedenko, A., Chen, X., Kuznetsov, H., Wang, C.-F., Coburn, D., Newburger, D. E., Morris, Q., Hughes, T. R., and Bulyk, M. L. (2009). Diversity and complexity in dna recognition by transcription factors. *Science*, **324**(5935), 1720–1723.

Bailey, T. L., Williams, N., Misleh, C., and Li, W. W. (2006). Meme: discovering and analyzing dna and protein sequence motifs. *Nucleic Acids Res*, **34**(Web Server issue), W369–W373.

Bajic, V. B., Tan, S. L., Chong, A., Tang, S., Ström, A., Gustafsson, J.-A., Lin, C.-Y., and Liu, E. T. (2003). Dragon ere finder version 2: A tool for accurate detection and analysis of estrogen response elements in vertebrate genomes. *Nucleic Acids Res*, **31**(13), 3605–3607.

Balhoff, J. P. and Wray, G. A. (2005). Evolutionary analysis of the well characterized Endo16 promoter reveals substantial variation within functional sites. *Proc Natl Acad Sci U S A*, **102**(24), 8591–8596.

Beadle, G. W. and Tatum, E. L. (1941). Genes direct the manufacture of proteins that control the basic metabolic functions.

Berg, O. G. and von Hippel, P. H. (1987). Selection of dna binding sites by regulatory proteins. statistical-mechanical theory and application to operators and promoters. *J Mol Biol*, **193**(4), 723–750.

Bird, A. (2007). Perceptions of epigenetics. *Nature*, **447**(7143), 396–398.

Blanchette, M. and Tompa, M. (2002). Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome Res*, **12**(5), 739–748.

Blanchette, M. and Tompa, M. (2003). Footprinter: A program designed for phylogenetic footprinting. *Nucleic Acids Res*, **31**(13), 3840–3842.

Boffelli, D., McAuliffe, J., Ovcharenko, D., Lewis, K. D., Ovcharenko, I., Pachter, L., and Rubin, E. M. (2003). Phylogenetic shadowing of primate sequences to find functional regions of the human genome. *Science*, **299**(5611), 1391–1394.

Bourque, G., Leong, B., Vega, V. B., Chen, X., Lee, Y. L., Srinivasan, K. G., Chew, J.-L., Ruan, Y., Wei, C.-L., Ng, H. H., and Liu, E. T. (2008). Evolution of the mammalian transcription factor binding repertoire via transposable elements. *Genome Res*, **18**(11), 1752–1762.

Bron, C. and Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM*, **16**(9), 575–577.

Brudno, M., Do, C. B., Cooper, G. M., Kim, M. F., Davydov, E., Program, N. I. S. C. C. S., Green, E. D., Sidow, A., and Batzoglou, S. (2003). Lagan and multi-lagan: efficient tools for large-scale multiple alignment of genomic dna. *Genome Res*, **13**(4), 721–731.

Burke, A. C. and Feduccia, A. (1997). Developmental patterns and the identification of homologies in the avian hand. *Science*, **278/5338**, 666–668.

Carroll, S. B. (2008). Evo-devo and an expanding evolutionary synthesis: a genetic theory of morphological evolution. *Cell*, **134**(1), 25–36.

Chan, E. T., Quon, G. T., Chua, G., Babak, T., Trochesset, M., Zirngibl, R. A., Aubin, J., Ratcliffe, M. J. H., Wilde, A., Brudno, M., Morris, Q. D., and Hughes, T. R. (2009). Conservation of core gene expression in vertebrate tissues. *J Biol*, **8**(3), 33.

Chen, F. and Capecchi, M. R. (1999). Paralogous mouse hox genes, hoxa9, hoxb9, and hoxd9, function together to control development of the mammary gland in response to pregnancy. *Proc Natl Acad Sci U S A*, **96**(2), 541–546.

Chiu, C.-H., Amemiya, C., Dewar, K., Kim, C.-B., Ruddle, F. H., and Wagner, G. P. (2002). Molecular evolution of the hoxa cluster in the three major gnathostome lineages. *Proc Natl Acad Sci U S A*, **99**(8), 5492–5497.

Chiu, C.-H., Dewar, K., Wagner, G. P., Takahashi, K., Ruddle, F., Ledje, C., Bartsch, P., Scemama, J.-L., Stellwag, E., Fried, C., Prohaska, S. J., Stadler, P. F., and Amemiya, C. T. (2004). Bichir hoxa cluster sequence reveals surprising trends in ray-finned fish genomic evolution. *Genome Res*, **14**(1), 11–17.

Ciechanover, A. (2005). Early work on the ubiquitin proteasome system, an interview with aaron ciechanover. interview by cdd. *Cell Death Differ*, **12**(9), 1167–1177.

Collas, P. (2010). The current state of chromatin immunoprecipitation. *Mol Biotechnol*, **45**(1), 87–100.

Corel, E., Pitschi, F., and Morgenstern, B. (2010). A min-cut algorithm for the consistency problem in multiple sequence alignment. *Bioinformatics*, **26**(8), 1015–1021.

Crocker, J. and Erives, A. (2008). A closer look at the eve stripe 2 enhancers of drosophila and themira. *PLoS Genet*, **4**(11), e1000276.

Daftary, G. S. and Taylor, H. S. (2006). Endocrine regulation of hox genes. *Endocr Rev*, **27**(4), 331–355.

Davidson, E. H. (2006). *The Regulatory Genome: Gene Regulatory Networks in Development and Evolution.* Academic Press, Amsterdam.

Dayhoff, M. O., Barker, W. C., and Hunt, L. T. (1983). Establishing homologies in protein sequences. *Methods Enzymol*, **91**, 524–545.

De, S., Teichmann, S. A., and Babu, M. M. (2009). The impact of genomic neighborhood on the evolution of human and chimpanzee transcriptome. *Genome Res*, **19**(5), 785–794.

de la Calle-Mustienes, E., Feijóo, C. G., Manzanares, M., Tena, J. J., Rodríguez-Seguel, E., Letizia, A., Allende, M. L., and Gómez-Skarmeta, J. L. (2005). A functional survey of the enhancer activity of conserved non-coding sequences from vertebrate iroquois cluster gene deserts. *Genome Res*, **15**(8), 1061–1072.

Dermitzakis, E. T. and Clark, A. G. (2002). Evolution of transcription factor binding sites in mammalian gene regulatory regions: conservation and turnover. *Mol Biol Evol*, **19**(7), 1114–1121.

Dieterich, C., Cusack, B., Wang, H., Rateitschak, K., Krause, A., and Vingron, M. (2002). Annotating regulatory dna based on man-mouse genomic comparison. *Bioinformatics*, **18 Suppl 2**, S84–S90.

Donath, A., Findeiß, S., Hertel, J., Marz, M., Otto, W., Schulz, C., Stadler, P. F., and Wirth, S. (2010). *Noncoding RNA. In Evolutionary Genomics and Systems Biology*. Wiley-Blackwell, Hoboken (Editor Caetano-Anollés G).

Eddy, S. R. (2001). Non-coding RNA genes and the modern RNA world. *Nat Rev Genet*, **2**(12), 919–929.

Edgar, R. C. (2004). Muscle: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, **5**, 113.

Elias, I. (2006). Settling the intractability of multiple alignment. *J Comput Biol*, **13**(7), 1323–1339.

Elnitski, L., Riemer, C., Petrykowska, H., Florea, L., Schwartz, S., Miller, W., and Hardison, R. (2002). Piptools: a computational toolkit to annotate and analyze pairwise comparisons of genomic sequences. *Genomics*, **80**(6), 681–690.

Elnitski, L., Hardison, R. C., Li, J., Yang, S., Kolbe, D., Eswara, P., O'Connor, M. J., Schwartz, S., Miller, W., and Chiaromonte, F. (2003). Distinguishing regulatory dna from neutral sites. *Genome Res*, **13**(1), 64–72.

Euler, R. (1983). On a classification of independence systems. *Mathematical Methods of Operations Research*, **27**, 123–136. 10.1007/BF01916906.

Felsenstein, J. (2003). *Inferring Phylogenies*. Sinauer, Sunderland, MA.

Fickett, J. W. (1996). Coordinate positioning of mef2 and myogenin binding sites. *Gene*, **172**(1), GC19–GC32.

Fickett, J. W. and Wasserman, W. W. (2000). Discovery and modeling of transcriptional regulatory regions. *Curr Opin Biotechnol*, **11**(1), 19–24.

Fisher, S., Grice, E. A., Vinton, R. M., Bessling, S. L., and McCallion, A. S. (2006). Conservation of ret regulatory function from human to zebrafish without sequence similarity. *Science*, **312**(5771), 276–279.

Floyd, R. W. (1962). Algorithm 97: Shortest path. *Commun. ACM*, **5**, 345–.

Frazer, K. A., Elnitski, L., Church, D. M., Dubchak, I., and Hardison, R. C. (2003). Cross-species sequence comparisons: a review of methods and available resources. *Genome Res*, **13**(1), 1–12.

Frith, M. C., Hansen, U., Spouge, J. L., and Weng, Z. (2004). Finding functional sequence elements by multiple local alignment. *Nucleic Acids Res*, **32**(1), 189–200.

Gardner, P. P., Wilm, A., and Washietl, S. (2005). A benchmark of multiple sequence alignment programs upon structural rnas. *Nucleic Acids Res*, **33**(8), 2433–2439.

Gardner, P. P., Daub, J., Tate, J. G., Nawrocki, E. P., Kolbe, D. L., Lindgreen, S., Wilkinson, A. C., Finn, R. D., Griffiths-Jones, S., Eddy, S. R., and Bateman, A. (2009). Rfam: updates to the RNA families database. *Nucleic Acids Res*, **37**(Database issue), D136–D140.

Gasch, A. P., Moses, A. M., Chiang, D. Y., Fraser, H. B., Berardini, M., and Eisen, M. B. (2004). Conservation and evolution of cis-regulatory systems in ascomycete fungi. *PLoS Biol*, **2**(12), e398.

Glover, J. N. and Harrison, S. C. (1995). Crystal structure of the heterodimeric bzip transcription factor c-fos-c-jun bound to dna. *Nature*, **373**(6511), 257–261.

Gompel, N., Prud'homme, B., Wittkopp, P. J., Kassner, V. A., and Carroll, S. B. (2005). Chance caught on the wing: cis-regulatory evolution and the origin of pigment patterns in drosophila. *Nature*, **433**(7025), 481–487.

Gotoh, O. (1990). Consistency of optimal sequence alignments. *Bulletin of Mathematical Biology*, **52**, 509–525. 10.1007/BF02462264.

Graur, D. and Li, W.-H. (2000). *Fundamentals of Molecular Evolution.* Sinauer Associates, Inc., Sunderland, MA.

Gross, D. and Harris, C. (1998). *Fundamentals of Queueing Theory.* Wiley, New York.

Hahn, M. W. and Wray, G. A. (2002). The g-value paradox. *Evol Dev*, **4**(2), 73–75.

Hare, E. E., Peterson, B. K., Iyer, V. N., Meier, R., and Eisen, M. B. (2008). Sepsid even-skipped enhancers are functionally conserved in drosophila despite lack of sequence conservation. *PLoS Genet*, **4**(6), e1000106.

Haygood, R., Fedrigo, O., Hanson, B., Yokoyama, K.-D., and Wray, G. A. (2007). Promoter regions of many neural- and nutrition-related genes have experienced positive selection during human evolution. *Nat Genet*, **39**(9), 1140–1144.

Hebert, D. N. and Molinari, M. (2007). In and out of the er: protein folding, quality control, degradation, and related human diseases. *Physiol Rev*, **87**(4), 1377–1408.

Helman, P., Moret, B. M. E., and Shapiro, H. D. (1993). An exact characterization of greedy structures. *SIAM J. Discret. Math.*, **6**, 274–283.

Hertz, G. Z. and Stormo, G. D. (1999). Identifying dna and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, **15**(7-8), 563–577.

Hofacker, I. L., Bernhart, S. H. F., and Stadler, P. F. (2004). Alignment of RNA base pairing probability matrices. *Bioinformatics*, **20**(14), 2222–2227.

Hou, Y. and Lin, S. (2009). Distinct gene number-genome size relationships for eukaryotes and non-eukaryotes: gene content estimation for dinoflagellate genomes. *PLoS One*, **4**(9), e6978.

Huang, X. and Miller, W. (1991). A time-efficient linear-space local similarity algorithm. *Adv. Appl. Math.*, **12**, 337–357.

Hughey, R. and Krogh, A. (1996). Hidden markov models for sequence analysis: extension and analysis of the basic method. *Comput Appl Biosci*, **12**(2), 95–107.

Istrail, S. and Davidson, E. H. (2005). Logic functions of the genomic cis-regulatory code. *Proc Natl Acad Sci U S A*, **102**(14), 4954–4959.

Jackson, R. R. P. and Henderson, J. C. (1966). The time-dependent solution to the many-server poisson queue. *Operations Research*, **14**(4), 720–722.

Jacob, F. and Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *J Mol Biol*, **3**, 318–356.

Johnson, D. S., Mortazavi, A., Myers, R. M., and Wold, B. (2007). Genome-wide mapping of in vivo protein-dna interactions. *Science*, **316**(5830), 1497–1502.

Jones, T. A., Otto, W., Marz, M., Eddy, S. R., and Stadler, P. F. (2009). A survey of nematode smy rnas. *RNA Biol*, **6**(1), 5–8.

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

Kim, J., He, X., and Sinha, S. (2009). Evolution of regulatory sequences in 12 drosophila species. *PLoS Genet*, **5**(1), e1000330.

King, O. D. and Roth, F. P. (2003). A non-parametric model for transcription factor binding sites. *Nucleic Acids Res*, **31**(19), e116.

Klinge, C. M. (2001). Estrogen receptor interaction with estrogen response elements. *Nucleic Acids Res*, **29**(14), 2905–2919.

Kozak, M. (1999). Initiation of translation in prokaryotes and eukaryotes. *Gene*, **234**(2), 187–208.

Kumar, S. (2005). Molecular clocks: four decades of evolution. *Nat Rev Genet*, **6**(8), 654–662.

Köhler, A. and Hurt, E. (2007). Exporting RNA from the nucleus to the cytoplasm. *Nat Rev Mol Cell Biol*, **8**(10), 761–773.

Lenhard, B., Sandelin, A., Mendoza, L., Engström, P., Jareborg, N., and Wasserman, W. W. (2003). Identification of conserved regulatory elements by comparative genome analysis. *J Biol*, **2**(2), 13.

Lenhof, H. P., Morgenstern, B., and Reinert, K. (1999). An exact solution for the segment-to-segment multiple sequence alignment problem. *Bioinformatics*, **15**(3), 203–210.

Levy, S., Hannenhalli, S., and Workman, C. (2001). Enrichment of regulatory signals in conserved non-coding genomic sequence. *Bioinformatics*, **17**(10), 871–877.

Lewin, B. (1983). *Genes II*. Wiley, New York.

Lewin, B. (2007). *Genes IX*. Jones & Bartlett Publishers.

Li, H. and Johnson, A. D. (2010). Evolution of transcription networks–lessons from yeasts. *Curr Biol*, **20**(17), R746–R753.

Loots, G. G., Ovcharenko, I., Pachter, L., Dubchak, I., and Rubin, E. M. (2002). rvista for comparative sequence-based discovery of functional transcription factor binding sites. *Genome Res*, **12**(5), 832–839.

Ludwig, M. Z. (2002). Functional evolution of noncoding dna. *Curr Opin Genet Dev*, **12**(6), 634–639.

Ludwig, M. Z. and Kreitman, M. (1995). Evolutionary dynamics of the enhancer region of even-skipped in drosophila. *Mol Biol Evol*, **12**(6), 1002–1011.

Ludwig, M. Z., Patel, N. H., and Kreitman, M. (1998). Functional analysis of eve stripe 2 enhancer evolution in drosophila: rules governing conservation and change. *Development*, **125**(5), 949–958.

Ludwig, M. Z., Bergman, C., Patel, N. H., and Kreitman, M. (2000). Evidence for stabilizing selection in a eukaryotic enhancer element. *Nature*, **403**(6769), 564–567.

Ludwig, M. Z., Palsson, A., Alekseeva, E., Bergman, C. M., Nathan, J., and Kreitman, M. (2005). Functional evolution of a cis-regulatory module. *PLoS Biol*, **3**(4), e93.

Lynch, V. J., Tanzer, A., Wang, Y., Leung, F. C., Gellersen, B., Emera, D., and Wagner, G. P. (2008). Adaptive changes in the transcription factor hoxa-11 are essential for the evolution of pregnancy in mammals. *Proc Natl Acad Sci U S A*, **105**(39), 14928–14933.

Ma, B., Tromp, J., and Li, M. (2002). Patternhunter: faster and more sensitive homology search. *Bioinformatics*, **18**(3), 440–445.

MacArthur, S. and Brookfield, J. F. Y. (2004). Expected rates and modes of evolution of enhancer sequences. *Mol Biol Evol*, **21**(6), 1064–1073.

MacArthur, S., Li, X.-Y., Li, J., Brown, J. B., Chu, H. C., Zeng, L., Grondona, B. P., Hechmer, A., Simirenko, L., Keränen, S. V. E., Knowles, D. W., Stapleton, M., Bickel, P., Biggin, M. D., and Eisen, M. B. (2009). Developmental roles of 21 drosophila transcription factors are determined by quantitative differences in binding to an overlapping set of thousands of genomic regions. *Genome Biol*, **10**(7), R80.

Mainguy, G., der Rieden, P. M. J. I., Berezikov, E., Woltering, J. M., Plasterk, R. H. A., and Durston, A. J. (2003). A position-dependent organisation of retinoid response elements is conserved in the vertebrate hox clusters. *Trends Genet*, **19**(9), 476–479.

Matys, V., Fricke, E., Geffers, R., Gössling, E., Haubrock, M., Hehl, R., Hornischer, K., Karas, D., Kel, A. E., Kel-Margoulis, O. V., Kloos, D.-U., Land, S., Lewicki-Potapov, B., Michael, H., Münch, R., Reuter, I., Rotert, S., Saxel, H., Scheer, M., Thiele, S., and Wingender, E. (2003). Transfac: transcriptional regulation, from patterns to profiles. *Nucleic Acids Res*, **31**(1), 374–378.

Medhi, J. (2003). *Stochastic Models in Queueing Theory*. Academic Press, Amsterdam.

Miranda, I., Silva, R., and Santos, M. A. S. (2006). Evolution of the genetic code in yeasts. *Yeast*, **23**(3), 203–213.

Moon, J. and Moser, L. (1965). On cliques in graphs. *Israel Journal of Mathematics*, **3**, 23–28. 10.1007/BF02760024.

Moreau, P., Brandizzi, F., Hanton, S., Chatre, L., Melser, S., Hawes, C., and Satiat-Jeunemaitre, B. (2007). The plant er-golgi interface: a highly structured and dynamic membrane complex. *J Exp Bot*, **58**(1), 49–64.

Morgenstern, B. (1999). Dialign 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, **15**(3), 211–218.

Morgenstern, B., Dress, A., and Werner, T. (1996). Multiple dna and protein sequence alignment based on segment-to-segment comparison. *Proc Natl Acad Sci U S A*, **93**(22), 12098–12103.

Morgenstern, B., Frech, K., Dress, A., and Werner, T. (1998). Dialign: finding local similarities by multiple sequence alignment. *Bioinformatics*, **14**(3), 290–294.

Morgenstern, B., Stoye, J., and Dress, A. (1999). Consistent equivalence relations: a set-theoretical framework for multiple sequence alignment. preprint.

Moses, A. M., Pollard, D. A., Nix, D. A., Iyer, V. N., Li, X.-Y., Biggin, M. D., and Eisen, M. B. (2006). Large-scale turnover of functional transcription factor binding sites in drosophila. *PLoS Comput Biol*, **2**(10), e130.

Mustonen, V. and Lässig, M. (2005). Evolutionary population genetics of promoters: predicting binding sites and functional phylogenies. *Proc Natl Acad Sci U S A*, **102**(44), 15936–15941.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, **48**(3), 443–453.

Nelson, C. C., Hendy, S. C., Shukin, R. J., Cheng, H., Bruchovsky, N., Koop, B. F., and Rennie, P. S. (1999). Determinants of dna sequence specificity of the androgen, progesterone, and glucocorticoid receptors: evidence for differential steroid receptor response elements. *Mol Endocrinol*, **13**(12), 2090–2107.

Notredame, C., Higgins, D. G., and Heringa, J. (2000). T-coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol*, **302**(1), 205–217.

Odom, D. T., Dowell, R. D., Jacobsen, E. S., Gordon, W., Danford, T. W., MacIsaac, K. D., Rolfe, P. A., Conboy, C. M., Gifford, D. K., and Fraenkel, E. (2007). Tissue-specific transcriptional regulation has diverged significantly between human and mouse. *Nat Genet*, **39**(6), 730–732.

Otto, W., Will, S., and Backofen, R. (2008). Structural local multiple alignment of RNA. In A. Beyer and M. Schroeder, editors, *German Conference on Bioinformatics*, volume 136 of *LNI*, pages 178–177. GI.

Otto, W., Stadler, P. F., López-Giraldéz, F., Townsend, J. P., Lynch, V. J., and Wagner, G. P. (2009). Measuring transcription factor-binding site turnover: a maximum likelihood approach using phylogenies. *Genome Biol Evol*, **2009**, 85–98.

Otto, W., Stadler, P. F., and Prohaska, S. J. (2011). Phylogenetic footprinting and consistent sets of local aligments. In R. Giancarlo and G. Manzini, editors, *CPM 2011*, volume 6661 of *Lecture Notes in Computer Science*, pages 118–131, Heidelberg, Germany. Springer-Verlag. in press.

Prabhakar, S., Noonan, J. P., Pääbo, S., and Rubin, E. M. (2006). Accelerated evolution of conserved noncoding sequences in humans. *Science*, **314**(5800), 786.

Prohaska, S. J., Fried, C., Amemiya, C. T., Ruddle, F. H., Wagner, G. P., and Stadler, P. F. (2004a). The shark hoxn cluster is homologous to the human hoxd cluster. *J Mol Evol*, **58**(2), 212–217.

Prohaska, S. J., Fried, C., Flamm, C., Wagner, G. P., and Stadler, P. F. (2004b). Surveying phylogenetic footprints in large gene clusters: applications to hox cluster duplications. *Mol Phylogenet Evol*, **31**(2), 581–604.

Reik, W. (2007). Stability and flexibility of epigenetic gene regulation in mammalian development. *Nature*, **447**(7143), 425–432.

Reimer, J. J. and Turck, F. (2010). Genome-wide mapping of protein-dna interaction by chromatin immunoprecipitation and dna microarray hybridization (chip-chip). part a: Chip-chip molecular methods. *Methods Mol Biol*, **631**, 139–160.

Reinitz, J. and Sharp, D. H. (1995). Mechanism of eve stripe formation. *Mech Dev*, **49**(1-2), 133–158.

Romano, L. A. and Wray, G. A. (2003). Conservation of endo16 expression in sea urchins despite evolutionary divergence in both cis and trans-acting components of transcriptional regulation. *Development*, **130**(17), 4187–4199.

Roth, F. P., Hughes, J. D., Estep, P. W., and Church, G. M. (1998). Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat Biotechnol*, **16**(10), 939–945.

Saaty, T. L. (1960). Time-dependent solution of the many-server poisson queue. *Operations Research*, **8**(6), 755–772.

Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, **4**(4), 406–425.

Sammeth, M., Morgenstern, B., and Stoye, J. (2003). Divide-and-conquer multiple alignment with segment-based constraints. *Bioinformatics*, **19 Suppl 2**, ii189–ii195.

Sandelin, A., Bailey, P., Bruce, S., Engström, P. G., Klos, J. M., Wasserman, W. W., Ericson, J., and Lenhard, B. (2004a). Arrays of ultraconserved non-coding regions span the loci of key developmental genes in vertebrate genomes. *BMC Genomics*, **5**(1), 99.

Sandelin, A., Alkema, W., Engström, P., Wasserman, W. W., and Lenhard, B. (2004b). Jaspar: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res*, **32**(Database issue), D91–D94.

Sanderson, M. J. (2006). Analysis of rates ("r8s") of evolution. Internet: http://loco.biosci.arizona.edu/r8s/. cited 2009 June 15.

Sankoff, D. (1985). Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM Journal on Applied Mathematics*, **45**(5), 810–825.

Santini, S., Boore, J. L., and Meyer, A. (2003). Evolutionary conservation of regulatory elements in vertebrate hox gene clusters. *Genome Res*, **13**(6A), 1111–1122.

Sayers, E. W., Barrett, T., Benson, D. A., Bolton, E., Bryant, S. H., Canese, K., Chetvernin, V., Church, D. M., DiCuccio, M., Federhen, S., Feolo, M., Fingerman, I. M., Geer, L. Y., Helmberg, W., Kapustin, Y., Landsman, D., Lipman, D. J., Lu, Z., Madden, T. L., Madej, T., Maglott, D. R., Marchler-Bauer, A., Miller, V., Mizrachi, I., Ostell, J., Panchenko, A., Phan, L., Pruitt, K. D., Schuler, G. D., Sequeira, E., Sherry, S. T., Shumway, M., Sirotkin, K., Slotta, D., Souvorov, A., Starchenko, G., Tatusova, T. A., Wagner, L., Wang, Y., Wilbur, W. J., Yaschenko, E., and Ye, J. (2011). Database resources of the national center for biotechnology information. *Nucleic Acids Res*, **39**(Database issue), D38–D51.

Schmidt, D., Wilson, M. D., Ballester, B., Schwalie, P. C., Brown, G. D., Marshall, A., Kutter, C., Watt, S., Martinez-Jimenez, C. P., Mackay, S., Talianidis, I., Flicek, P., and Odom, D. T. (2010). Five-vertebrate chip-seq reveals the evolutionary dynamics of transcription factor binding. *Science*, **328**(5981), 1036–1040.

Schwartz, S., Kent, W. J., Smit, A., Zhang, Z., Baertsch, R., Hardison, R. C., Haussler, D., and Miller, W. (2003). Human-mouse alignments with blastz. *Genome Res*, **13**(1), 103–107.

Seidman, J. G. and Seidman, C. (2002). Transcription factor haploinsufficiency: when half a loaf is not enough. *J Clin Invest*, **109**(4), 451–455.

Sereno, P. C. (1999). The evolution of dinosaurs. *Science*, **284**(5423), 2137–2147.

Siepel, A., Bejerano, G., Pedersen, J. S., Hinrichs, A. S., Hou, M., Rosenbloom, K., Clawson, H., Spieth, J., Hillier, L. W., Richards, S., Weinstock, G. M., Wilson, R. K., Gibbs, R. A., Kent, W. J., Miller, W., and Haussler, D. (2005). Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Res*, **15**(8), 1034–1050.

Steiper, M. E. and Young, N. M. (2006). Primate molecular divergence dates. *Mol Phylogenet Evol*, **41**(2), 384–394.

Storm, C. E. V. and Sonnhammer, E. L. L. (2003). Comprehensive analysis of orthologous protein domains using the hops database. *Genome Res*, **13**(10), 2353–2362.

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

Subramanian, A. R., Weyer-Menkhoff, J., Kaufmann, M., and Morgenstern, B. (2005). Dialign-t: an improved algorithm for segment-based multiple sequence alignment. *BMC Bioinformatics*, **6**, 66.

Subramanian, A. R., Kaufmann, M., and Morgenstern, B. (2008). Dialign-tx: greedy and progressive approaches for segment-based multiple sequence alignment. *Algorithms Mol Biol*, **3**, 6.

Tafer, H., Rose, D., Marz, M., Hertel, J., Bartschat, S., Kehr, S., Otto, W., Donath, A., Tanzer, A., Bermudez-Santana, C., Gruber, A. R., Jühling, F., Engelhardt, J., Busch, A., Hiller, M., Stadler, P. F., and Dieterich, C. (2009). Comparative analysis of non-coding rnas in nematodes. PREPRINT.

Taft, R. J., Pheasant, M., and Mattick, J. S. (2007). The relationship between non-protein-coding dna and eukaryotic complexity. *Bioessays*, **29**(3), 288–299.

Tagle, D. A., Koop, B. F., Goodman, M., Slightom, J. L., Hess, D. L., and Jones, R. T. (1988). Embryonic epsilon and gamma globin genes of a prosimian primate (galago crassicaudatus). nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. *J Mol Biol*, **203**(2), 439–455.

Tarjan, R. E. (1972). Depth first search and linear graph algorithms. *SIAM J. Computing*, **1**, 146–160.

Tatusov, R. L., Fedorova, N. D., Jackson, J. D., Jacobs, A. R., Kiryutin, B., Koonin, E. V., Krylov, D. M., Mazumder, R., Mekhedov, S. L., Nikolskaya, A. N., Rao, B. S., Smirnov, S., Sverdlov, A. V., Vasudevan, S., Wolf, Y. I., Yin, J. J., and Natale, D. A. (2003). The cog database: an updated version includes eukaryotes. *BMC Bioinformatics*, **4**, 41.

Tautz, D. (2000). Evolution of transcriptional regulation. *Curr Opin Genet Dev*, **10**(5), 575–579.

Taylor, H. M. and Karlin, S. (1998). *An Introduction to Stochastic Modelling.* Academic Press, San Diego.

Teng, G. and Papavasiliou, F. N. (2007). Immunoglobulin somatic hypermutation. *Annu Rev Genet*, **41**, 107–120.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, **22**(22), 4673–4680.

Thompson, J. D., Plewniak, F., and Poch, O. (1999). A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res*, **27**(13), 2682–2690.

Tirosh, I., Reikhav, S., Levy, A. A., and Barkai, N. (2009). A yeast hybrid provides insight into the evolution of gene expression regulation. *Science*, **324**(5927), 659–662.

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

Tronche, F., Ringeisen, F., Blumenfeld, M., Yaniv, M., and Pontoglio, M. (1997). Analysis of the distribution of binding sites for a tissue-specific transcription factor in the vertebrate genome. *J Mol Biol*, **266**(2), 231–245.

Tsong, A. E., Tuch, B. B., Li, H., and Johnson, A. D. (2006). Evolution of alternative transcriptional circuits with identical logic. *Nature*, **443**(7110), 415–420.

Tuch, B. B., Galgoczy, D. J., Hernday, A. D., Li, H., and Johnson, A. D. (2008). The evolution of combinatorial gene regulation in fungi. *PLoS Biol*, **6**(2), e38.

Ureta-Vidal, A., Ettwiller, L., and Birney, E. (2003). Comparative genomics: genome-wide analysis in metazoan eukaryotes. *Nat Rev Genet*, **4**(4), 251–262.

Veitia, R. A. (2008). One thousand and one ways of making functionally similar transcriptional enhancers. *Bioessays*, **30**(11-12), 1052–1057.

Vickaryous, M. K. and Hall, B. K. (2006). Human cell type diversity, evolution, development, and classification with special reference to cells derived from the neural crest. *Biol Rev Camb Philos Soc*, **81**(3), 425–455.

Vingron, M. and Argos, P. (1991). Motif recognition and alignment for many sequences by comparison of dot-matrices. *J Mol Biol*, **218**(1), 33–43.

Wagner, G. P. (2005). The developmental evolution of avian digit homology: an update. *Theory Biosci*, **124**(2), 165–183.

Wagner, G. P. and Lynch, V. J. (2008). The gene regulatory logic of transcription factor evolution. *Trends Ecol Evol*, **23**(7), 377–385.

Wagner, G. P., Fried, C., Prohaska, S. J., and Stadler, P. F. (2004). Divergence of conserved non-coding sequences: rate estimates and relative rate tests. *Mol Biol Evol*, **21**(11), 2116–2121.

Wagner, G. P., Otto, W., Lynch, V., and Stadler, P. F. (2007). A stochastic model for the evolution of transcription factor binding site abundance. *J Theor Biol*, **247**(3), 544–553.

Wapinski, I., Pfeffer, A., Friedman, N., and Regev, A. (2007). Natural history and evolutionary principles of gene duplication in fungi. *Nature*, **449**(7158), 54–61.

Warshall, S. (1962). A theorem on boolean matrices. *J. ACM*, **9**, 11–12.

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

Wasserman, W. W., Palumbo, M., Thompson, W., Fickett, J. W., and Lawrence, C. E. (2000). Human-mouse genome comparisons to locate regulatory sites. *Nat Genet*, **26**(2), 225–228.

Watson, J. D. and Crick, F. H. C. (1953). Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, **171**(4356), 737–738.

Wei, W. and Yu, X.-D. (2007). Comparative analysis of regulatory motif discovery tools for transcription factor binding sites. *Genomics Proteomics Bioinformatics*, **5**(2), 131–142.

Will, S., Reiche, K., Hofacker, I. L., Stadler, P. F., and Backofen, R. (2007). Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput Biol*, **3**(4), e65.

Wilson, M. D. and Odom, D. T. (2009). Evolution of transcriptional control in mammals. *Curr Opin Genet Dev*, **19**(6), 579–585.

Wilson, M. D., Barbosa-Morais, N. L., Schmidt, D., Conboy, C. M., Vanes, L., Tybulewicz, V. L. J., Fisher, E. M. C., Tavaré, S., and Odom, D. T. (2008). Species-specific transcription in mice carrying human chromosome 21. *Science*, **322**(5900), 434–438.

Wong, W. S. W. and Nielsen, R. (2004). Detecting selection in noncoding regions of nucleotide sequences. *Genetics*, **167**(2), 949–958.

Wray, G. A. (2001). Resolving the hox paradox. *Science*, **292**, 2256–2257.

Wray, G. A. (2007). The evolutionary significance of cis-regulatory mutations. *Nat Rev Genet*, **8**(3), 206–216.

Wray, G. A., Hahn, M. W., Abouheif, E., Balhoff, J. P., Pizer, M., Rockman, M. V., and Romano, L. A. (2003). The evolution of transcriptional regulation in eukaryotes. *Mol Biol Evol*, **20**(9), 1377–1419.

Young, R. L., Caputo, V., Giovannotti, M., Kohlsdorf, T., Vargas, A. O., May, G. E., and Wagner, G. P. (2009). Evolution of digit identity in the three-toed italian skink chalcides chalcides: a new case of digit identity frame shift. *Evol Dev*, **11**(6), 647–658.

Zhou, Q., Brown, J., Kanarek, A., Rajagopal, J., and Melton, D. A. (2008). In vivo reprogramming of adult pancreatic exocrine cells to beta-cells. *Nature*, **455**(7213), 627–632.

Curriculum Scientiae

Education:

05/2006 – 05/2011	PhD student at University of Leipzig • Group of Prof. Peter F. Stadler, Chair of Bioinformatics • Thesis: *Transcriptional Regulatory Elements – Detection and Evolutionary Analysis*
10/2000 – 11/2005	Diploma student in Bioinformatics at Friedrich-Schiller-University, Jena • Diploma; Grade: Excellent • Thesis: *Multiple, Local Sequence-Structure-Alignments*
10/1999 – 11/2000	Diploma student in Informatics at Friedrich-Schiller-University, Jena

Working Experience:

03/2009 – 05/2009	Research project at Yale University, New Haven, USA • Lab of Prof. Günter Wagner, Chair of Ecology and Evolutionary Biology • Project: *Development of a maximum likelihood approach for measuring transcription factor binding site turnover using phylogenies*
01/2006 – 04/2006	Internship at Bayer-AG, Wuppertal • Lab of Dr. Michael Seewald, Bayer HealthCare, Section Pharmaceutical Research • Project: *Development of a content analysis pipeline for assigning inhibitors to pharmalogical target proteins based on scientific literature*
10/2002 – 10/2003	Research Assistant at Friedrich-Schiller-University, Jena • Group of Prof. Rolf Backofen, Chair of Bioinformatics

10/2001 – 10/2002	Tutor at Friedrich-Schiller-University, Jena • Group of Prof. Gerd Wechsung, Chair of Theoretical Informatics

Awards:

since 05/2006	PhD Fellowship of the Max-Planck-Society for Mathematics in the Sciences
05/2006 – 04/2009	PhD Fellowship of the Konrad-Adenauer-Foundation
2005	Best Diploma Award
04/2003 – 07/2005	Scholarship of the Konrad-Adenauer-Foundation
2003	Best Intermediate Diploma Award

IT-Knowledge:

Operating systems:	UNIX, Linux, Windows
Programming:	C/C++ (incl. Qt), Java, Perl, Ada, R, PHP, PostScript, Prolog, Pascal, BASIC
Markup languages:	Latex, HTML, XML
Database systems:	IBM DB2 (MySQL)
Applications:	Matlab, Office-Packages (e.g. Excel, Word, PowerPoint)

Language Skills:

German:	native speaker
English:	fluent
Russian:	basic knowledge

Publications

Journals:

Otto W, Stadler PF, Prohaska SJ (2011).
Local, Multiple Alignments Based on Consistent Subsets of Pairwise Alignment Collections. in prep. for *Theoretical Computer Science*

Otto W, Stadler PF, Prohaska SJ (2011).
Phylogenetic Footprinting and Consistent Sets of Local Aligments. In R. Giancarlo and G. Manzini, editors, *CPM 2011*, volume 6661 of *Lecture Notes in Computer Science*, pages 118–131, Heidelberg, Germany. Springer-Verlag. in press.

Tafer H, Rose D, Marz M, Hertel J, Bartschat S, Kehr S, Otto W, Donath W, Tanzer A, Bermudez-Santana C, Gruber AR, Juhling F, Engelhardt J, Busch A, Hiller M, Stadler PF, Dieterich C (2009).
Comparative Analysis of Non-Coding RNAs in Nematodes. Submitted to *Genome Consortium*

Otto W, Stadler PF, López-Giraldéz F, Townsend JP, Lynch VJ, Wagner GP (2009).
Measuring Transcription Factor-Binding Site Turnover: A Maximum Likelihood Approach Using Phylogenies. *Genome Biol Evol.* 2009 May 25;1:85–98.

Jones TA*, Otto W*, Marz M, Eddy SR, Stadler PF (2009).
A Survey of Nematode SmY RNAs. *RNA Biol.* 2009 Jan-Mar;6(1):5–8.

* equal authorship

Otto W*, Will S*, Backofen R (2008).
Structural Local Multiple Alignment of RNA. In A. Beyer and M. Schroeder, editors, *German Conference on Bioinformatics*, volume 136 of LNI, pages 178–177. GI.

* equal authorship

Wagner GP, Otto W, Lynch V, Stadler PF (2007).
A Stochastic Model for the Evolution of Transcription Factor Binding Site Abundance. *J Theor Biol.* 2007 Aug 7;247(3):544–53.

Books:

Donath A, Findeiß S, Hertel J, Marz M, Otto W, Schulz C, Stadler PF, Wirth S (2010).
Chapter 14: **Noncoding RNA** In *Evolutionary Genomics and Systems Biology*; ed Caetano-Anollés, G (Wiley-Blackwell, Hoboken NJ); 2010; pp.251–293.

Conferences / Seminars:

22nd Annual Symposium on Combinatorial Pattern Matching (Presenter)
Otto W, Stadler PF, Prohaska SJ: *Phylogenetic Footprinting and Consistent Sets of Local Aligments*
06/2011; Palermo, Italy

IMPRS Mathematics in the Sciences Workshop (Presenter)
Otto W: *Tracker - The Search for Footprints in the Sand of Evolution*
02/2010; Leipzig, Germany

JCB Workshop (Attendee)
11/2009; Jena, Germany

7th Bioinformatics Leipzig Autumn Seminar (Presenter)
Otto W: *RNA Meets Promoter - Annotation Based on Regulation*
10/2009; Děcín, Czech Republic

6th Bioinformatics Leipzig Autumn Seminar (Presenter)
Otto W: *Take the Maximum Likelihood Walk on the TFB-Site of Life*
10/2008; Děcín, Czech Republic

German Conference on Bioinformatics 2008 (Presenter)
Otto W, Will S, Backofen R: *Structural Local Multiple Alignment of RNA*
09/2008; Dresden, Germany

IMPRS Mathematics in the Sciences Workshop (Presenter)
Otto W: *Tracker - The Search for Footprints in the Sand of Evolution*
07/2008; Leipzig, Germany

Bioinformatics Freiburg Spring Seminar (Presenter)
Otto W: *Life and Let Die - A Stochastic TFBS-Evolution Model*
04/2008; Freiburg, Germany

23th TBI Vienna Winter Seminar (Attendee)
02/2008; Bled, Slovenia

Alignment Meeting (Attendee)
01/2008; Göttingen, Germany

4th Bioinformatics Leipzig Autumn Seminar (Presenter)
Otto W: *Transcription Factor Binding Site Abundance*
10/2007; Děcín, Czech Republic

Bioinformatics Freiburg Winter Seminar (Presenter)
Otto W: *Mulora - An Approach for Multiple, Local Sequence-Structure-Alignments*
01/2007; Freiburg, Germany

3th Bioinformatics Leipzig Autumn Seminar (Attendee)
10/2006; Děcín, Czech Republic

Collaborations:

Prof. Günter Wagner, Department of Ecology and Evolutionary Biology
Yale University; New Haven (CT), USA

Prof. Rolf Backofen, Chair for Bioinformatics
University of Freiburg; Freiburg, Germany

Prof. Ivo L. Hofacker, Theoretical Biochemistry Group
University of Vienna; Vienna, Austria

Prof. Stefan Schuster, Department of Bioinformatics
Friedrich-Schiller-University; Jena, Germany

Reviewer Activities:

Journal of Experimental Zoology Part B: Molecular and Developmental Evolution
ed. Wagner, G (Wiley-Liss, Inc.)

f

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder Dienstleistungen als solche gekennzeichnet.

Leipzig, den 12. Juli 2011

(Wolfgang Otto)