Bioinformatics Approaches to Large Scale Genome Comparison, Including the Identification of Conserved Noncoding Regions

Jomuna Veronica Choudhuri

M.Sc. Jomuna Veronica Choudhuri
AG Praktische Informatik
Technische Fakultät
Universität Bielefeld

email: jomuna@techfak.uni-bielefeld.de

Genehmigte Dissertation zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften (Dr. rer. nat.).
Von Jomuna Veronica Choudhuri am 07. Mai 2003
der Technischen Fakultät an der Universität Bielefeld vorgelegt.
Am 15. Juli 2003 verteidigt und genehmigt.

Prüfungsausschuß:

Prof. Dr. rer. nat. Robert Giegerich, Universität Bielefeld
Dr. rer. nat. Thomas Schmitt-John, Universität Bielefeld
Prof. Dr. rer. nat. Jens Stoye, Universität Bielefeld
Dr. rer. nat. Carsten Voß, Universität Bielefeld

Gedruckt auf alterungsbeständigem Papier nach ISO 9706.

Bioinformatics Approaches to Large Scale Genome Comparison, Including the Identification of Conserved Noncoding Regions

Dissertation zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften (Dr. rer. nat.)

der Technischen Fakultät der Universität Bielefeld
vorgelegt von

Jomuna Veronica Choudhuri

7. Mai 2003

Acknowledgments

I would like to express my gratitude to my two supervisors, Prof. Dr. Robert Giegerich and Dr. Thomas Schmitt-John, for their careful guidance in these years, for their keen sense of timing and, above all, for giving me the opportunity to build this bridge between biology and computer science.

I am sincerely thankful to Alex Sczyrba, who always found some time to patiently answer all my questions, and helped me master innumerable technical challenges. His continuous support was very important over the course of this thesis. I also profited very much from the close collaboration with Stefan Kurtz and Chris Schleiermacher, the developers of `REPuter` and `GenAlyzer` – we were a good team.

I would like to acknowledge all members of the Practical Computer Science group, and, in particular, my loyal officemate Thomas Töller, who tolerated my brazilian temperament, and had fun when I started cursing in portuguese... I also enjoyed very much being in the "Graduiertenkolleg Bioinformatik", and I thank all the scholars for contributing to such a pleasant atmosphere during all these years. Furthermore, I thank the members of the Department of Developmental Biology and Molecular Pathology, specially Jutta Wirth, Daniela Ehling, and Sonja Fuchs, who actively contributed to the development of this thesis. I am thankful to Michael Beckstette, who offered me valuable clues on the *database* subject.... and to Alex Sczyrba and Jörn Clausen who provided me with new ideas and constructive criticism while reading the draft of my thesis.

I am deeply grateful to my parents and my brother, for always encouraging me to be an independent thinker, and for having confidence in my abilities to go after new things that inspired me. I can not imagine being the person I am without such a great family. A very special thanks to my best friends Patricia, Fernanda and Graciela, so far... but so close. Last, but not least, I will always remember the support and the constant encouragement of my boyfriend Jannik Fritsch. I thank him for enjoying life with me, and for showing me the strength of a smile.

Abstract

The repeatitive pattern of genomic DNA reveals much about the structure and organization of genomes. `REPuter` is a bioinformatics tool that efficiently finds repeatitive substrings in large genomic sequences. The identification of repeats enables a wide range of biological interpretations. Here we describe how this single software is applied to many different biological problems: assembly check, localization of low copy repeats, identification of string uniqueness, matching cDNAs onto genomic sequences, and comparison of gene structure. The adaptation of `REPuter` to the era of comparative genomics is described considering its biologically meaningful improvements, generating the new visualization `GenAlyzer`.

The ability to compare different genomes enables researchers to look for conservation and functionality of regulatory regions. Considering that sequences containing vital information are under greater evolutionary pressure than sequences without function, the former are expected to be more conserved during evolution. This clearly holds for protein coding sequences and can also be exploited in the study of regulatory or other noncoding functional sequences. Using comparative genomics, these noncoding functional sequences can be identified as conserved noncoding sequences (CNSs). Today, the functionality of CNSs is determined by very elaborate and time consuming experimental methods. Depending on the background level of similarity between the organisms being compared, the amount of conserved noncoding sequences can be very large and almost impracticable to handle in the wet-lab. Making use of available tools for genome annotation and comparison, we have developed *Connosseur*, a Conserved Noncoding Sequences Repository Generator. It provides bioinformatics support for generating and screening the set of CNSs between two genomic sequences. *Connosseur* automates several tools in a computational cascade, finally returning a repository of CNSs and associated information. For the data storage, we used the relational database system PostgreSQL. Further analyses can be carried out upon selected pCNSs. This includes determining uniqueness, overrepresented words analysis and comparison to known functional CNSs. This approach allowed us to identify several potential CNSs between mouse and human genomic sequences. Overall, *Connosseur* provides a flexible and extensible basis for in-depth studies of pCNSs.

Contents

List of Figures

List of Tables

1 Introduction

The scientific community is celebrating fifty years of the molecular structure of nucleic acids discovery by Watson and Crick [117]. Three-dimension models of DNA were built to look for the energetically most favorable configurations compatible with the helical parameters provided by X-ray analysis. This combination of data led Watson and Crick to describe the double stranded helical structure of the DNA as we know it today. Those results triggered many other experiments, demonstrating the significant role of DNA for information transfer in living organisms, due to its capacity of semiconservative self-replication. At that time, those findings contributed to the understanding of many mysteries regarding phenotypes, inheritance, and evolution. From then on, progress in molecular biology research was impressive. Back in 1953, Watson and Crick could not even imagine that they would live to see the completion of the sequence of the human genome, exactly fifty years later.

In the past decade, remarkable advances in DNA sequencing technologies, as well as in data and information processing systems, have increased the content and quality knowledge of genome biology. Up to now, biological mechanisms have been elucidated basically by studying the effect of evolution on specific sequences and functions. With the sequencing of whole genomes, these studies are being extended to the analysis of entire genomes of different organisms. This large-scale approach demands the development of new technologies, appropriate to cope with ever increasing sequences, delivering significant results to enable a meaningful biological interpretation. The last years have been marked by the employment of computational methods to extract information out of such raw sequencing data. A kind of *bridge* is being established between biology and computer science. Originally, the concept of *bioinformatics* was to bring together problems and programs, joining those who develop the tools for those who can best interpret the results. Today, the success of such an interdisciplinary approach is demonstrated by the establishment of graduate programs in bioinformatics at several universities and research institutes. A new kind of professional is being created, with skills in both disciplines. Computational biology is evolving fast, and more and more researchers with different backgrounds are speaking about *doing bioinformatics*. Recently, the wide range of computational applications in biology is occupying not only biologists (coming from a variety of fields themselves) and computer scientists, but also mathematicians, physicists, medical doctors, and, sometimes, even engineers. The original interdisciplinarity is changing into multidisciplinarity, and the term *bioinformatics* is becoming as broad as biology.

To store, retrieve, analyze, and predict the composition or the structure of biomolecules

– nucleic acids as well as proteins – is the *classical way to do bioinformatics*. The main reason to handle biological information in computers lies in the basic characteristics of biomolecules. Large polymers are nothing but a chain of simpler molecular modules (monomers), which share common features. Usually, each monomer is of the same general class, even considering that it keeps its own well-defined set of characteristics. This facilitates the modeling of such molecules in silico, and even their interactions. Accordingly, the monomers of DNA or protein can be treated computationally as simple letters over an alphabet. This relatively simple approach justifies the explosion of available bioinformatics tools observed in the last 10 years [16, 113]. Several programs aiming at similar goals have been developed, but employing different computational strategies. It has to be carefully decided which tool is the most suitable for the research purpose of an individual user. This selection can make a large difference in the quality of the obtained results and the effort required. Since the beginning of genome sequencing projects, the huge amount of data generated requires the constant improvement of bioinformatics tools in order to cope with large-scale sequence analysis. Faster algorithms are being developed and old ones adapted to the new era and needs. In parallel, wet-lab biologists are improving also experimental techniques, in order to confirm the information provided by the in silico results. Satisfying the developing requirements and increasing necessities of genomic analysis is one of the major challenges which keeps bioinformatics up-to-date.

Genome Analysis

The classical genomic era started with the launch of the Human Genome Project in 1990. At that time, the available sequence information was based only on results of experimental work. With the publication of the draft sequences [70], it turned out that the gene content of the human genome was about five fold overestimated by biological scientists, who based their assumption on the quantity of proteins coded in our genome. Today, about 5% of the whole genome is supposed to code for proteins, suggesting a larger amount of noncoding sequences than was originally thought. This indicates that the human genome is constituted of roughly two distinct DNA fractions: repetitive and unique sequences. In general, the unique fraction comprises the functional constituents of the genomes, such as coding regions and regulatory elements. Yet the functional significance of the majority of repeated elements is less clear. These sequences are usually transposon-derived, presenting various gradations of repetitiveness, based on their copy number and degree of sequence similarity. These mobile elements are found in multiple copies in eukaryotic genomes, and can be subdivided into two basic groups: the tandemly arrayed ones, such as microsatellites and telomeres, and the interspersed ones, such as short or long interspersed repeats (SINEs and LINEs, respectively) [5]. However, the *uniqueness* of DNA is relative, since many genes are present in clusters (rRNA genes, immunoglobulin genes segments), or in duplicates, as a consequence of segmental duplications of larger regions containing these genes. Mathematically, the

term *repeated substring* refers to any region which has suffered at least one duplication, being inserted anywhere else in the genome. These segments can be larger than LINEs or SINEs, and may contain genes, as well as all kinds of repeated elements [34].

The presence of general repetitive sequences has been subject of several studies regarding human diseases and genome disorders. The jumping of mobile elements from one site to another may lead to the disruption of the structural integrity of a gene, resulting either in the extinction of its function, or even in the activation of genes that were previously silent [59]. In contrast to the classical mechanism of genetic diseases, where the abnormal phenotype is usually a result of point mutations, a variety of human diseases is also a consequence of segmental duplications of large genomic regions. These often result in rearrangements of the genome architecture. In this case, the complete loss or gain of a gene sensitive to a dosage effect may occur. Furthermore, the genomic structure can also be altered by homologous recombination between low copy repeated sequences during meiosis [76, 11, 12]. Depending on the relative orientation of these sequences to each other, the resulting rearrangement may be a deletion, a duplication, or an inversion. According to the genes affected by the genomic reorganization, the phenotypes are more or less severe.

The fact that eukaryotic genomes are full of repeated substrings has complicated their sequencing and assembly, and is still a subject of discussion. The fewer and less complex the repeats, the easier a genome is to sequence [34]. The reason that a genome is actually possible to sequence and assemble is the presence of unique fractions interdigitated among the repetitive regions. With the sequencing of genomes, bioinformatics was challenged to create and adapt algorithms that should support the problems biology has been faced by the sequencing results. Furthermore, a variety of computational molecular biology and genomic databases has been created in the last years, in order to collect the enormous amount of data generated by the genome research projects. They provide integrated data management and analysis systems for structural and functional annotations.

One of the most important methods in bioinformatics to carry out the analysis of biological sequences is the *sequence similarity search*. Hints to the understanding of structure and function of a molecular sequence often arise from homologies to other, previously studied molecules. Local sequence similarities are computed by several database search algorithms. The use of a current comprehensive sequence database is essential to any similarity search. The identification of repeated elements, and repetitive substrings in general, as well as gene discovery, or regulatory elements prediction are examples of applications based on sequence similarity search. However, the utilization of only this method could be inappropriate when predicting genes or functional units, because it relies on information derived from known sequence features. The tendency is to find only elements that are similar to those already known. With technological advances and the sequencing of different organisms' genomes, the detection of functional regions by comparing evolutionary related genomic sequences with each other has gained attention [96]. In the evolutionary process under selective pressure, functional sequences tend to

evolve at a slower rate than non-functional sequences. So, the comparison of genomic sequences between distinct species is a promising approach for an accurate detection of genes or regulatory regions.

Comparative Genomics

The nature and priorities of bioinformatics research for genome applications are changing. The sequence availability of multiple genomes offers the possibility to analyze the differences and similarities between all genes of different species. Since the beginning of the sequencing of organisms' genomes, researchers have recognized that a single genome taken in isolation does not reveal much by itself. Consequently, the necessity of comparing the sequences between different species emerged, taking into consideration their evolutionary background. A subfield within bioinformatics has appeared, called *comparative genomics*.

Cross-species sequence comparison has already shown to be a powerful tool for genome analysis and annotation [55, 73]. As mentioned before, functionally important parts of a genome are under selective pressure during evolution. Therefore, they tend to be more conserved than non-functional parts that are primarily subject to random genetic drift. Consequently, *local sequence conservations* are expected to indicate biological functionality. Again, the detection of protein-coding regions or regulatory sites is carried out by sequence similarity search, but this time, looking for similarities between sequences of two or more *distinct* species. The different focus of research aims is continuously supported by modern computational tools. Many recent studies have successfully used the comparative genomics approach to identify novel genes and regulatory elements [48, 49, 92].

Comparative genomics has also become an essential tool in biomedical research. The intuitive use of bioinformatics in disease investigations is contributing to the discovery of new genes with medical relevance to humans. Biomedical researchers have usually utilized mutant organisms in order to get clues about protein function. Today, technological advances and DNA sequence information provide the necessary tools to create specific and directed mutants. For instance, the insertion of an altered or a non-functional copy of a gene into a living organism enables the observation of changes in behavior or development. Since mice breed quickly and share more than 95% of their genes with humans, they are the most used animal model for large-scale functional studies. Similar phenotypes in mouse and human diseases suggest that the same genetic pathways are disrupted by the corresponding mutation in both species. The first assumption of a genetic abnormality is a defect in protein coding regions. It is easier to detect such sequences, and wet-lab experimental methods are well characterized. However, sometimes a mutation in a regulatory element is the cause of the disease rather than the coding region itself, resulting in a damage in the gene expression. There are many different kinds of regulatory sequences, and they lie in noncoding regions of a genome, making it difficult to identify the defect, both in silico and in the wet-lab. But the genome comparison

between different organisms at the sequence level aids this detection, when looking for conserved noncoding sequences. With the available bioinformatics tools, a pre-selection of such conserved, potential functional noncoding regions is possible, considering them the remaining sequences after the subtraction of conserved repeated elements or protein coding regions. However, it is important to take into account some practical problems. Two genomes to be compared should have enough similarities enabling the identification of homologous regions. But, since the divergence from the common ancestor, a significant amount of mutations has accumulated, and selection has occurred. It becomes difficult to distinguish between conservation of noncoding sequences due to functional constraints or insufficient divergence time. Thus, the choice of the genomes used for comparison has to be done carefully, and experimental analyses have to be carried out in addition to computer-assisted evaluation.

Some researchers define these functional genomic investigations as the *post-genomic* era. In parallel to the improvement of experimental methods, such as high-throughput sequence analysis and micro-array technologies, bioinformatics tools have also to be developed according to the needs of the new era. This includes accurate data mining, with more precise prediction tools. In fact, existing sequence analysis programs that can be *adapted* to comparative genomics, proteomics, and satisfying the future demands, are the tools with greater acceptance in the biological community. Moreover, flexible software that permits its utilization in several different aspects and open questions in biology will gain ground in the future. The end-user biologists prefer integrated systems that are easy to use, and attend the vast majority of their investigation purposes.

1.1 Contribution

Motivated by the ever increasing data resulting from several sequencing projects, this thesis concentrates on the various interpretations of repetitive sequences in whole genomes. We analyze the potential of bioinformatics approaches to large scale genome comparison and describe the integration of several programs in a pipeline for the automatic identification of conserved noncoding sequences.

For sequence analysis, we propose the use of `REPuter`, a previously developed tool for the identification of repeated substrings in large sequences [68, 67]. It is a flexible and user-friendly software, and its properties and characteristics led us to employ it in several different types of sequence investigations. We describe in detail five examples of how `REPuter` can successfully be used in biological applications: sequence assembly checks, low copy repeats localization, string uniqueness identification, cDNA matching onto genomic sequences, and gene structure comparison.

Whole genome comparison approaches have been developed in the last years as a consequence of the availability of different organisms' genomic sequences. Today, it is a widely used method in order to understand the regulatory aspects of gene expression, as well as the evolution among species. We have collaborated with the developers of

REPuter, to improve the implementation and design of this software from a biological point of view. We describe how REPuter has been adapted to the comparative genomics era, resulting in the computational tool called vmatch, and its interactive, easy-to-use interface GenAlyzer. These tools are being used worldwide for many other sequence analysis tasks than REPuter was originally thought for. We delineate the biological motivations to improve REPuter, describing the main differences of its successors vmatch and GenAlyzer.

Based on these enhancements for genomic comparisons, we have also applied vmatch and GenAlyzer to biomedical research. We concentrate here on the mutation responsible for the *wobbler* phenotype in mice. It is known that the affected mice suffer an autosomal recessive mutation producing severe motoneuron degeneration and astrogliosis in the spinal cord. This mouse is used as model organism for human spinal muscleatrophy (SMA) and amyotrophic lateral sclerosis (ALS). In this thesis, we present the analysis of the *wobbler* critical region at the sequence level, making use of the mouse draft genomic sequences. We check the assemblies of the mouse contigs using vmatch, analyze the results in GenAlyzer's match graphs, and compare them with the final assembly, published in December 2002 [116]. Furthermore, we compare the *wobbler* genomic region to its homologous segment in the human genome, as highly conserved sequences among species are expected to contain vital information. Wet-lab experiments prior to this work have demonstrated that none of those candidate genes lying in the *wobbler* critical region holds the mutation, suggesting that a regulatory element could be affected.

Motivated by this hypothesis, and maintaining the line of computational approaches to sequence analysis, we developed the *Conserved Noncoding Sequences Repository Generator* (*Connosseur*). In this tool we implemented a chain of established bioinformatics programs, providing exhaustive automatic analysis of genomic sequences, from annotation to whole genome comparisons. Such comparisons have already been shown to aid the identification of conserved noncoding sequences with potential regulatory roles, due to the evolutionary positive selections on sequences with biological function. In this thesis, we describe the different levels of sequence analysis and the utilization of comparative genomics in *Connosseur*, aiming at the in silico identification of conserved noncoding sequences prior to wet-lab experiments. The product of *Connosseur* is a Repository of pCNSs, managed by a Relational Database Management System (RDBMS), facilitating the access and retrieval of intermediate and final computed information.

This pre-selection of conserved noncoding sequences facilitates the end-user biologists to verify their functionality in the wet-lab. Furthermore, *Connosseur* is constructed in a flexible way, allowing its extension by integrating other tools for the in-depth in silico analysis of conserved noncoding sequences in the future.

1.2 Outline

To establish a background of essential knowledge of the computational resources utilized in this work, we start, in Chapter 2, describing the `REPuter` software and its visualization facilities. Afterwards, we turn to the analysis of the several biological applications we have carried out with `REPuter`.

Chapter 3 introduces the mouse and human genome sequencing projects, providing an insight into the topic of cross-species sequence comparison. With the justified necessity to compare the genomic sequences of distinct species in Chapter 3, we delineate the initiated improvements of `REPuter` to adapt it to the comparative genomics era in Chapter 4. Here we present the main differences between `REPuter` and `GenAlyzer`, the enhanced interactive visualization interface. The computational improvements of `REPuter` are implemented in the `vmatch` program, also delineated in Chapter 4. We demonstrate that `vmatch` and its visualization `GenAlyzer` are an excellent approach for pairwise comparative genomics.

Based on the `vmatch` and `GenAlyzer` programs, Chapter 5 describes the intense sequence analysis of the *wobbler* critical region both in mouse and human genomic sequences. The progress of the sequencing projects is analyzed by checking the assembly and localizing the candidate genes for the mutation in the genomic sequence. The surprising outcome of comparative genomics in the *wobbler* critical region leads us to redirect the approach for the mutation detection. Besides coding sequences, also regulatory elements may be affected, generating a genetic disease by damaging the gene expression. Extracting conserved noncoding sequences from raw DNA sequences is the first step for the identification of regions with regulatory functions. In this context, we introduce *Connosseur* in Chapter 6, the developed cascade of established bioinformatics tools, supporting the annotation and comparison of genomic sequences. In addition to other features of *Connosseur*, all computational steps towards the extraction of potential conserved noncoding sequences with possible functional significance are described in Chapter 6. The resulting repository of potential conserved noncoding sequences is delineated in detail, explaining its structured architecture supported by a relational database.

Finally, Chapter 7 summarizes the conclusions we draw from this thesis, providing motivations for future work.

2 The REPuter Software

The REPuter software was developed to compute and analyze repeats, i.e., repetitive substrings in large DNA sequences. It finds all repeats above a given level of significance in whole genomes [67]. This automated task provides a visualization of the repeat structure of the given sequence, facilitating biologists to analyze and understand the structure of genomes. Therefore, REPuter is a suitable tool to be applied for initial approaches for genomic scale studies. Once the repeat pattern of a DNA sequence is computed, the repeats can be further investigated by other methods, like database searches, looking for any known characteristics of the detected repeated sequences.

This chapter gives an overview of the repetitive sequence types in genomes (Section 2.1). Section 2.2 delineates four criteria to successfully analyze repeats in large sequences. Furthermore, repeats terminologies used in this work are listed in Section 2.3, introducing the concept of *repeats* in both biological and mathematical meaning. Afterwards, the REPuter family of programs is defined in Section 2.4, explaining each step of repeat search. Finally, we discuss how the intensive work between computer scientists and biologists led to the recognition that repeat analysis can be interpreted and used in a wide range of biological applications. This versatility of REPuter is demonstrated in detail in Section 2.5, showing that REPuter is used for many other sequence analysis tasks than it was originally thought for [67].

2.1 Repetitive Sequences in Genomes

The wide range of size diversity of different organism's genomes has already been known in the molecular biology community for years. It even led researchers to suggest that the amount of DNA should have a correlation with organismal complexity. This theory lost its meaning after observing that our genome is about 200 times larger than that of the yeast, *S. cerevisae*, but also 200 times smaller than that of *Amoeba dubia*. Today, it is known that the different amounts of DNA between species is suggested by the different quantities of repeated sequences in their genomes.

In experimental approaches, researchers had already observed large quantities of such repeated sequences in the different genomes. With the availability of the sequences of those genomes, such observations are not only confirmed, but also the analysis and recognition of new repeats can be done more accurately. In *Homo sapiens*, for instance, the coding sequences hold about 5% of the genomes, while repeated sequences account for 50% or more [70]. This large discrepancy is well demonstrated in Figure 2.1, showing a 1 megabase pairs (Mb) segment on chromosomes 22 and 2, where the interspersed

repeat content (red) relative to coding regions (blue) can be clearly visualized.

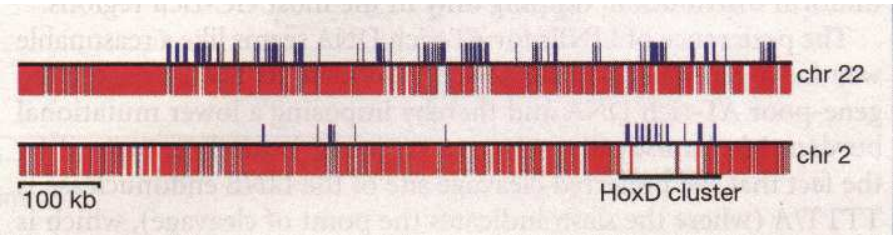

Figure 2.1: Regions of about 1 Mb on chromosomes 22 and 2. The red lines represent interspersed repeats and the blue ones exons from known genes. (Taken from [70])

The word *repeat* has a very broad range of employment. Some of the classes they fall in are shortly introduced below:

1. Interspersed Repeats: transposon-derived elements, which inserts themselves all over the genome are called interspersed repeats. They account for the most abundant repeat type in the human genome (for instance, Long Interspersed Repeats (LINEs), Short Interspersed Repeats (SINEs) and Long Terminal Repeats (LTRs));

2. Simple Sequence Repeats: direct repetitions of short k-mers, like micro- or minisatellites are denominated simple sequence repeats;

3. Pseudogenes: copies of cellular genes, which have been only partially retroposed may have lost their function, therefore being called pseudogenes;

4. Segmental duplications: when blocks of about 10-300 kb genomic sequence are copied from one region into another in the genome, they are generalized as segmental duplications;

5. Tandem Repeats: repeated sequences which are arranged successively in a row are called tandem repeats, like the ones found in centromeres and telomeres, for instance.

This classification into different repeat types is done exclusively under a biological point of view. Their constitution, the way they arise and spread, their functions and possible structural consequences for the genome lead biologists to divide them into these categories promoting a better understanding and information exchange. More detailed definitions of the term *repeat* will be given in Section 2.3, distinguishing between different kinds of repeated substrings.

2.2 Requirements for Repeat Analysis

`REPuter` was implemented to fulfill all requirements of a software which systematically searches for repetitions in large sequences. On a genomic scale, the great challenge is to satisfy the following criteria:

1. *Efficiency*: the tool must be able to cope with whole genomes, i.e., up to 3 - 4 billion bp, in realistic time and space consumption. `REPuter` is linear in time with respect to sequence length, a consequence of the underlying suffix tree data structure [68].

2. *Flexibility*: to represent a biological realistic and significant model, the tool must recognize not only exact, but also degenerate repeats, allowing a certain amount of error. In biological sequences, it is also important to analyze not only direct (forward) repeats, but also reverse complemented ones (palindromic).

3. *Interactive Visualization*: as large amounts of data are generated by such a computation, the tool has to provide a visualization of the whole genomic repetitive structure, allowing the user to get not only an overview, but also to zoom into details of particular segments, extracting the sequence of interest for further investigations (e.g., database searches).

4. *Compositionality*: as the repeat finding is considered a basic and initial step in genome structure analysis, the program has to provide a simple interface enabling compositions with other tools.

`REPuter` meets these requirements in several ways, which are demonstrated in detail in the next sections.

2.3 Terminology of Repeats

The word *repeat* is defined in different ways by biologists and computer scientists. From now on, the terminology will be distinguished as explained below. The different categories of repeats can be also visualized in Figure 2.2.

Repeated Element This terminology is generally employed by biologists, with respect to the various DNA sequences that are present in multiple copies in the genomes. The most abundant repetitive elements found in the human genome are *interspersed* and *simple sequence repeats*, as mentioned before, in Section 2.1.

Repeat or Repeated Substring Under a computer scientist's point of view, a repeat is a mathematically simple object – namely a substring w of a sequence S occurring twice in S. Considering a DNA sequence being a string S of length n, a substring $S[i, j]$ contained in the string is represented by the pair of nucleotide positions (i, j). A pair of substrings is called a *repeat* or a *repeated substring* if it fits in one of the different categories described in this section, satisfying the parameters set by the user.

Unique Substring The absolute mathematical observation of a repeat leads to the consequence that a substring w is *unique* in S when it is not a repeat in S (Figure 2.14).

Match If two sequences S_1 and S_2 have a common substring w, this means that this substring is a repeat from the concatenated sequence S_1S_2. In this case, the repeat is called a *match* between both different sequences.

Exact Repeat (or Match) Closely related sequences, which have not suffered yet great evolutionary pressure because of recent divergence or vital functionality, are often found as exact repetitions. Formally, this is defined by a pair of substrings $S = ((i_1, j_1), (i_2, j_2))$, if and only if $(i_1, j_1) \neq (i_2, j_2)$ and $S[i_1, j_1] = S[i_2, j_2]$.

Containment A pair of exact repeated substrings has evidently embedded repeats of shorter length. That is, a pair of positions (i_1, j_1), $i_1 \leq j_1$ contains the pair of positions (i_2, j_2), $i_2 \leq j_2$ if and only if $i_1 \leq i_2$ and $j_2 \leq j_1$.

Maximality It would not make sense, biologically, to study all the repeats that are embedded in other, larger repeats. This would generate a large amount of data which is redundant and also has to be further analyzed. For the purpose of avoiding such consequences, REPuter reports only exact maximal repeats. A repeat is called *maximal*, iff it is not contained in any other repeat. The maximality of a repeat is given by its surrounding characters: they have to be unequal in both repeat instances. Formally, an exact repeat is called *maximal* if and only if $S[i_1 - 1] \neq S[i_2 - 1]$ and $S[j_1 + 1] \neq S[j_2 + 1]$ [52].

Degenerate Repeat (or Match) During evolution, sequences that have been repeated are under different selective pressures. Mutations can accumulate in distinct rates between bases and segments, usually depending on their functional meaning. Those mutations include base substitutions, resulting in *mismatches* and generating a *k-mismatch* repeat, but also insertions and deletions, generating a *k-difference* repeat. Both kind of degenerate repeats are described in the following.

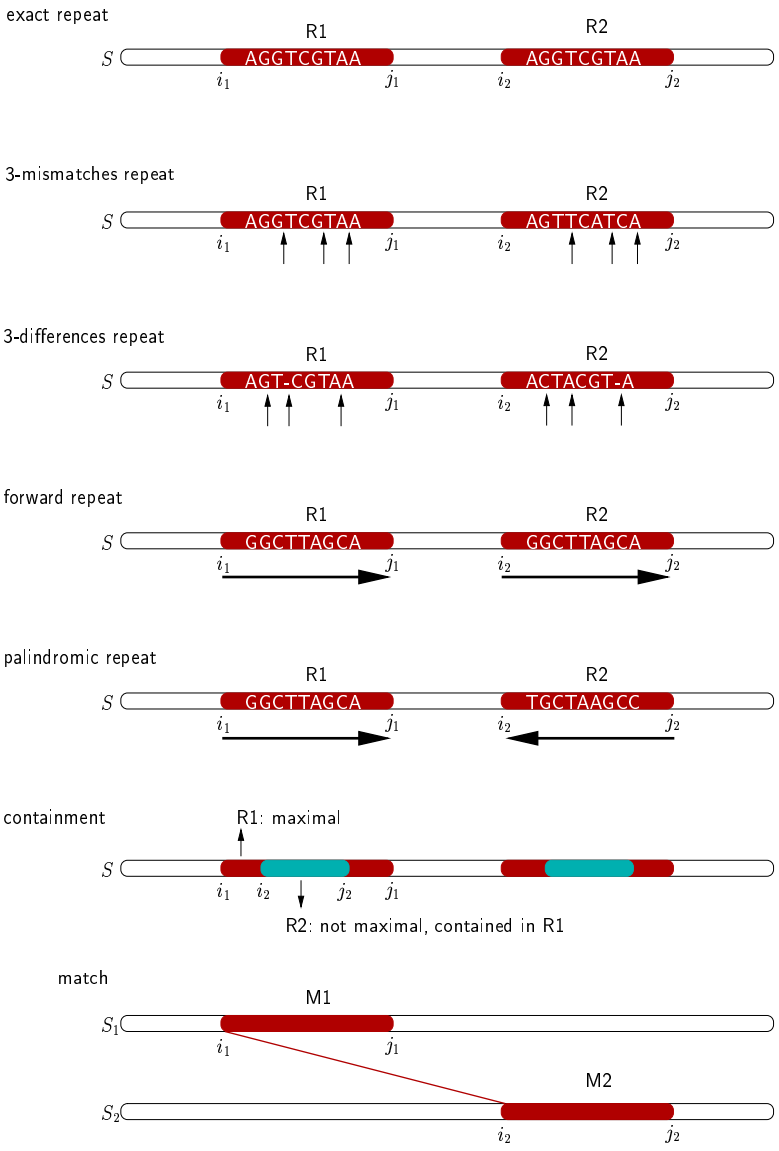

Figure 2.2: Graphical representation of the different categories of repeats. A repeat between two different sequences is called a *match*, which can also be exact, degenerate, forward, and/or palindromic.

K-mismatch Repeat (or Match) When both substrings $S[i_1, j_1]$ and $S[i_2, j_2]$ have the same length, but not the same content, i.e., they include mismatches of single characters, they are called *k-mismatch* repeats. The number of positions where $S[i_1, j_1]$ differs from $S[i_2, j_2]$ is called the *Hamming distance* [52].

K-difference Repeat (or Match) Insertions and deletions in two repeat instances result in two nonequal-length strings $S[i_1, j_1]$ and $S[i_2, j_2]$ called *k-difference* repeats. In this case, all three types of edit operations are possible: mismatches, deletions and insertions. This kind of repeat search makes more sense when analyzing biological sequences, since chromosomal rearrangements and evolution among species include deletions or insertions of segments. Once considering insertions and deletions (indels) between two instances of a repeat, gap costs have to be taken into account. The minimum number of edit operations needed to transform $S[i_1, j_1]$ in $S[i_2, j_2]$ is called the *Edit distance* [52].

Direct or Forward Repeat (or Match) When a segment of genomic DNA is duplicated elsewhere maintaining its original orientation, it is said to be a *direct* repeat, also called *forward* repeat.

Palindromic or Reverse-complemented Repeat (or Match) In case of an inverted insertion of substrings, both instances of the repeat are said to be the *reverse complement* of each other (*palindromes*). Usually, base pairs form between bases on opposing strands, but in case of inverted repetitious sequences, the bases can pair within single chains, forming hydrogen-bonded hairpin loops [57]. This formation may happen during momentary denaturation of such palindromic regions, facilitating the interaction with specific DNA-binding proteins. However, such structures can also promote chromosomal rearrangements, leading to human diseases (see Subsection 2.5.3).

2.4 The REPuter Family of Programs

REPuter fulfills all criteria needed for repeat analysis tasks, as expected for a tool that systematically searches for repetitive structures in large sequences (see Section 2.2). The identification of different types of repeated substrings in DNA sequences is done in a flexible, and user specific way. The user chooses between the categories to be identified (see Section 2.3), according to investigation purposes. The diversity of repeats computed by REPuter explains its versatility. The possibility of searching not only for exact, but also degenerate repeats in DNA sequences is biological relevant when looking for recent or ancient duplications, when comparing closely or distantly related species or even interpreting genomic disorders. Basically, the repeat search is done in three main computational steps, subdividing the REPuter family of programs in 3 members: *REPfind*, finding repeated substrings, *REPselect*, which selects user-specific repeats from the *REPfind* output, and *REPvis*, the graphical visualization of the repetitive structure. All three steps are relatively independent from each other, i.e, they present only chronological dependencies, but do not necessarily need to be used together.

2.4.1 REPfind

The search engine of REPuter is called *REPfind*, which uses an efficient and compact implementation of suffix trees in order to locate exact repeats in linear space and time [68]. These exact repeats are also called *seeds*. The construction of degenerate repeats proceeds by extending these *seeds*, using a dynamic programming approach, to the left and to the right until the limit of degeneracy is achieved [68, 98]. The allowance of errors between both repeat instances is here determined by the user. In this first step of repeat finding, besides the degeneracy, the user also defines other parameters for the search, like type (forward or palindromic), and minimal repeat length. It is important to emphasize that REPuter is not heuristic: it guarantees to find all repeats according to the specified parameters. Additionally, REPuter reports the significance scores of the repeats by means of the expectation value (*E-value*) of the alignment. The *E-value* is defined as the number of different alignments with scores equivalent to or better than a given least score s that are expected to occur in a database search by chance. The lower the E value, the more significant the score [103, 62].

One of the most striking features of REPuter is the fact that it is fast. This is a very important feature considering that whole genomes can be composed of up to 3-4 billion base pairs. It facilitates the biological investigation on such large sequence data, once the results can be rapidly recalculated after an update of the sequencing status, for instance. Table 2.1 shows the time and space required by REPuter for computing all exact or degenerate repeats of a given length l, in different DNA input sequences. Clearly, it confirms our previous statement that REPuter runs in linear time and space relative to the input sequence length. Analyzing the fifth column, where an edit distance of 10 has been allowed for each search, we observe that looking for degenerate repeats rather than for exact ones causes only a small computational overhead. The explanation above demonstrates how *REPfind* accomplishes the requirements of *flexibility* and *efficiency*, described in Section 2.2.

Genome	size (Mb)	l (bp)	Edist=0 (sec)	Edist=10 (sec)	space (MB)
H. influenzae	1.75	140	7	32	24
E. coli	4.42	150	20	44	61
S. cerevisae	11.50	180	58	103	159
H. sapiens, Chr 22	32.06	670	186	191	443
D. melanogaster	114.00	700	1047	1125	1581

Table 2.1: Time (in seconds) and space (in MB) efficiency of REPuter on different input sequences, and under given parameters (the computation was run on a Sun UltraSparc II 400MHz).

2.4.2 REPselect

Dealing with large genomic sequences, the computation of all repeats may generate a very large amount of data, depending on the given parameters. Taking this fact into account, the second member of the REPuter family of programs, *REPselect*, permits the user to *select*, as the name suggests, repeats from the output of *REPfind*. The choice of selecting repeats can be done, again, according to the user's defined criteria. It delivers repeats of chosen length, degeneracy, or significance into further analyses routines. Consequently, the user has a more restricted overview of the repeat structure of the segment in question. This is an interesting option when the initial output is very large, instead of running *REPfind* again with more restricted parameters. However, this selection step is not *necessary*, the user just runs it to adapt the output to individual needs. Moreover, *REPselect* allows to sort the repeats according to different criteria, like length or position in the input sequences. Finally, *REPselect* provides an open interface to other programs, or scripts, permitting the user to automate the steps of selection and further investigations, for instance. In this way, REPuter fulfills the requisite of *compositionality*, mentioned in Section 2.2.

2.4.3 REPvis

The visualization of data is essential for their evaluation by human inspection. *REPvis* provides an easy-to-use interactive visualization of the repeat structures computed by *REPfind*, completing the list of requirements for a repeat analysis tool (Section 2.2). Being the third member of REPuter family of programs does not necessarily mean that it is also the third step in the computation. As mentioned before, the dependencies of the three programs is relative to the analysis' purposes. Chronologically, the visualization of repeats occurs after their localization by *REPfind*. But one may want to pass through the selection step (with *REPselect*) before visualizing the results.

As the output of *REPfind* is actually a list of repeat positions, and this list can be very long under certain circumstances, it is incontestable the necessity of a good visualization of the calculated repetitions. This includes the ability of getting an overview of the whole repeat structure as well as zooming into regions of particular interest. In the following, we present how the visualization of the repeat graph is built up and the input sequences can be annotated.

The Repeat Graph Layout

The input DNA sequence is first represented as a horizontal line near the top of the repeat graph, which is duplicated in the bottom of the graph (Figure 2.3). A diagonal line joins the beginning of the first instance of the repeat in the top line with the beginning of its respective second instance in the bottom line. *Projecting* the second instance of the repeat onto the top line would give the representation of the repeats in

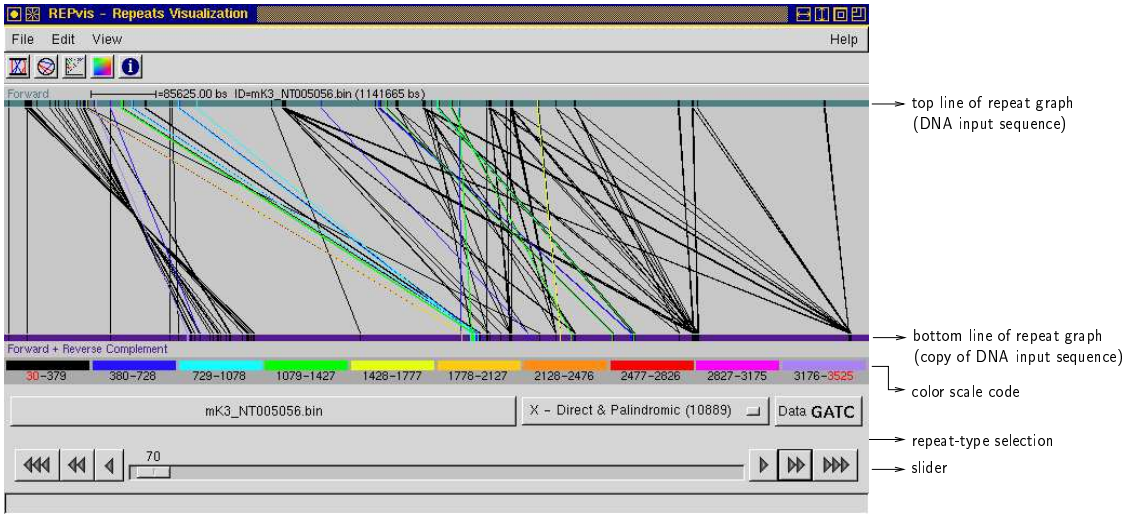

Figure 2.3: *REPvis* visualization of the repeat graph of REPuter.

the input sequence. By shifting the slider, the user can determine the minimal length of the repeats shown in the graph. There is a 10-color scale code which helps the user to identify the repeat lengths displayed in the graph. The shortest, or less significant repeats are coded in black. If the parameters were set below a certain threshold, going all the way down with the slider will bring the user to hit the noise level (as it can be seen later in Figure 2.15). Even though, the larger, or more significant repeats can be seen shining up in colors before the black background noise.

The graph allows the user to get an overview of the repeat structure along the whole input sequence. Observing a specific region with interesting repetitions, it is possible to bring up an *inspector window* with a mouse click in the graph region (see Figures 2.6 and 2.16). In this window, the user can zoom in and out the structure with the left and right mouse button, respectively. Selecting a repeat by clicking on its position on the input sequence line, the respective features of this repeat will be shown in a browser box below the graph. Information about the repeat can be visualized in both data and annotation browsers in Figure 2.6. This feature allows biologists to use the specific information of the repetitive sequences for further analysis. They can be directly submitted to programs like FASTA and BLAST through the *inspector window* interface. In order to investigate a larger region which contains one or more repeats, this subsequence can also be extracted and saved into a file by opening the *Save Subsequence* dialog and defining the start and end positions of the substring. This approach facilitates the search for other attributes in the region of interest, rather than looking only at the repeat itself. Moreover, it permits the user to recalculate the repeats in the selected region with lower thresholds, avoiding unnecessary waste of time and space when using such low thresholds for the entire input sequence.

The Annotation Graph

The repeat graph displayed by *REPvis* can be annotated with additional lines and symbols. These are called the *annotation graph*, where the user can display sequence specific annotations, like gene predictions, localization of LINEs, SINEs, ALUs, etc., as well as extra features of particular interest, such as markers or even contig positions (Figure 2.4). These attributes are specified as colored arcs, blocks and several other symbols, together with their respective start and stop positions, listed in an *annotation file*, which can be uploaded in the *inspector window* (for more details see [98]). The *annotation graph* will be shown together with the *repeat graph*, permitting the user to verify or hypothesize correspondences between the sequence annotation and the particular repeat structure found.

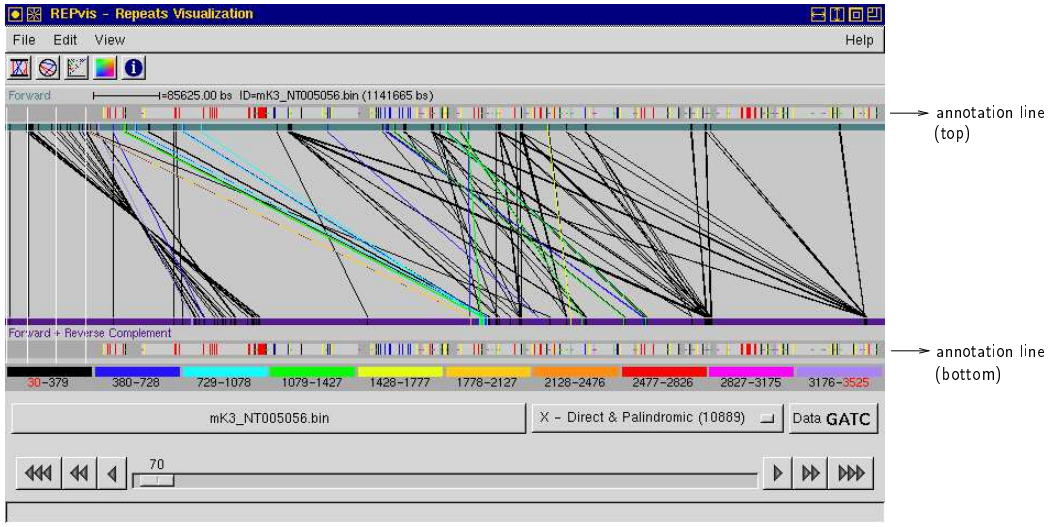

Figure 2.4: *REPvis* visualization of the repeat graph including the annotation graph.

2.5 Development of Manifold Applications

During the development of REPuter, it became clear that a software which systematically searches for repeat structures is actually multi-tasking, as it can be appointed to different fields of application. Recovering strings in larger strings, or finding repetitions in sequences can be interpreted biologically in a variety of manners. This renders REPuter a very suitable tool for such purposes, covering all needs for repeat analysis in large genomic scale. In the following, we explain how repeats arise and how they may behave in the genome, mentioning some of the consequences of such behavior. Moreover, we demonstrate the broad range of repeat analysis offered by REPuter in five applications: checking assemblies, localizing low copy repeats (LCRs), identifying

unique strings, matching cDNAs or ESTs onto genomic sequences, and last, but not least, comparing gene structures. It will get clear that from the above mentioned applications, only the identification of LCRs is related to the traditional kind of repeat analysis. The other ones represent the wide range of sequence analysis tasks based only on the repeat structures of genomic data.

2.5.1 The Biological Meaning of Repeats

It is well known that repeated sequences are involved in some biological mutational mechanisms. Most of the repeated elements are transposon-derived, that is, they can "jump" around in the genome, inserting themselves several times in different regions. Sometimes, it can be demonstrated that such insertions are directed into defined target regions. It has been described that LINEs, for instance, occur at much higher density in AT-rich regions. In the contrary, SINEs seem to target preferentially GC-rich DNA for insertion. The mechanisms of such accumulations is still unclear [107]. These transposable elements can cause extinction of gene function by insertion into coding or regulatory sequences, resulting in deleterious mutations. On the other hand, LTRs, for instance, have the property of sometimes activating genes that where previously silent [59].

Repeated substrings can also enhance chromosomal rearrangements. Segmental duplications of short or large genomic regions occur as well as translocations, inversions and other chromosomal abnormalities. This leads to a shuffling of large sections of chromosomes, bringing together previously unlinked genes and modifying recombination frequencies. There are basically two categories of chromosomal rearrangements: inter- and intrachromosomal rearrangements [70, 104]. The first ones involve two different chromosomes and may occur between nonhomologous chromosomes. These include Robertsonian translocations, where whole arms of acrocentric chromosomes are exchanged, and reciprocal translocations, which result from a single break in each of the two participating chromosomes. Intrachromosomal rearrangements are aberrations that involve a single chromosome. It includes interstitial and terminal deletions, duplications and inversions. In this case, rearrangements can occur between a single homologue (exchanges of sister chromatids) or involve both homologous chromosomes. These changes in the genomic organization are often observed linked to human malformations and genetic diseases. This topic is further explored in Subsection 2.5.3. There are also hypotheses suggesting that the presence of palindromic repeats (i.e., reverse complemented) hints to the formation of hairpin structures [57]. These may be responsible for inversions or deletions during replication (see Section 2.5.3).

2.5.2 Assembly Check

The assembly of genomes aims to place fragments of sequenced DNA in the proper order and orientation in the chromosomes. When BACs (Bacterial Artificial Chromosomes) or contigs are sorted into regions along the chromosomes, they sometimes show large

overlapping sequences, inferring that they should have been concatenated or that one BAC contains a smaller one. Considering the subsequent annotation of the genome with the analysis of its gene contents as well as its regulatory sequences, it is important to withdraw superfluous regions, otherwise the true frequency of such features could be under- or overestimated [70].

A simple plausibility check of the assembly of a sequence can be done by applying REPuter to it. Overlapping regions between BAC sequences or contigs can be identified through the visualization of very long repeats. These may indicate assembly errors. If those large repeats are palindromic, it could suggest that one of the repeat instances may have been assembled in the wrong orientation. This approach was applied to two different sequences: first, the 11 concatenated contigs of human chromosome 22[1], shown in Figures 2.5 and 2.6; second, the draft sequences of mouse contigs sequenced in the the Rummage project[2], depicted in Figure 2.7 (more about this sequencing project in Chapter 5).

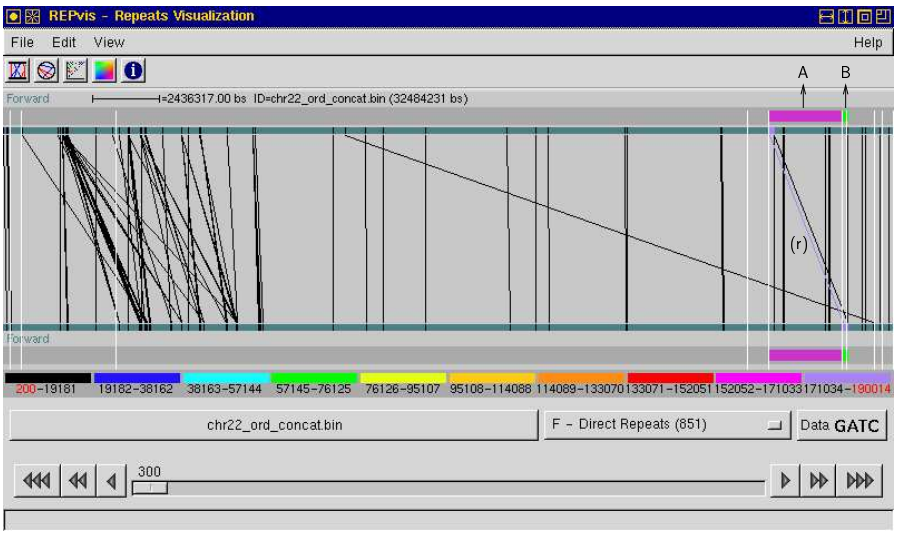

Figure 2.5: Assembly check of human chromosome 22. The repeat graph displays exact and direct repeats with a minimum length of 300 bp. The chromosome's contigs are separated from each other by vertical white lines. The color code points to an unexpected long repeat (r) of 190 kb (purple). Furthermore, in the beginning of the sequence, a quite confusing repeat structure is observed. This indicates the localization of low copy repeats in this chromosome (see Section 2.5.3).

[1] Accession numbers: NT_011516.3, NT_011517.2, NT_011519.4, NT_025937.1, NT_011520.5, NT_011521.1, NT_011523.4, NT_011524.2, NT_011525.3, NT_019197.2, NT_011526.3 (GenBank, March 2001).

[2] Rummage numbers: mK5-185K22, mK11-48H20. These contigs were sequenced in the Genome Sequencing Center in Jena, Germany. These designations refer to the sequences in the draft stage, before being published in GenBank.

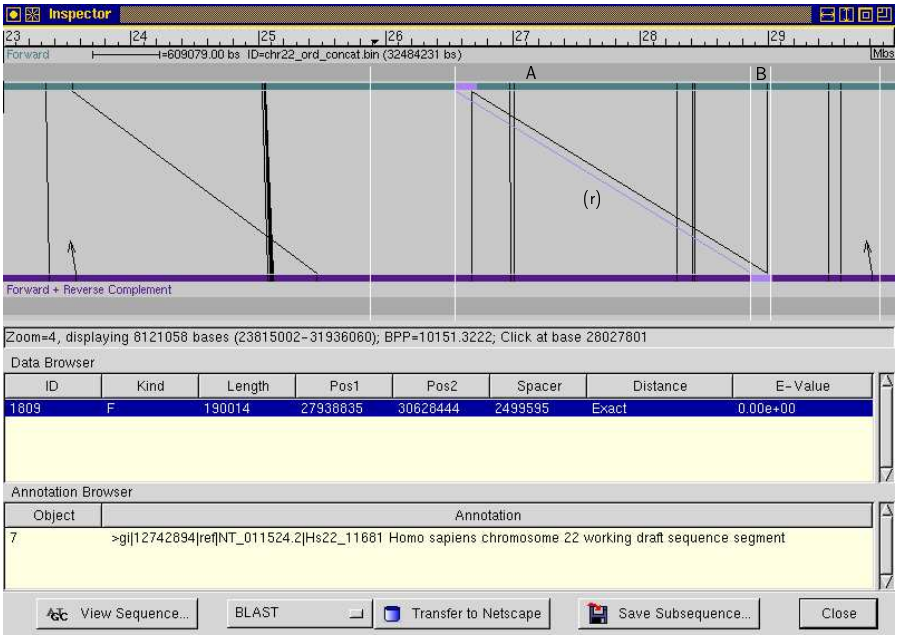

Figure 2.6: Assembly check of human chromosome 22 (ctd.). Enlarged view of the region from Figure 2.5 containing the contigs 7 (A) and 8 (B). The focus is set on the large repeat (r), in purple. These zoomed view clearly enables the user to recognize an erroneous assembly of contig 8 (B), which had already been assembled in the beginning of contig 7 (A).

Analyzing the repeat structure of chromosome 22 in Figure 2.5, an overlapping region of 190 kb is observed. As this unexpected exact repeat corresponds to the usual size of BACs, it clearly indicates an assembly error. This region was zoomed in for a better visualization (Figure 2.6). The data browser shows the features of the repeated sequence and the annotation browser gives the information about the contig the repeat belongs to. It turned out that the complete contig 8 (B) had already been assembled in the beginning of contig 7 (A). This error has been corrected in the current version of chromosome 22. Another example of assembly check is depicted in Figure 2.7. Both mouse contigs mK5-185K22 and mK11-48H20 were about to be published when we looked at their positioning relative to each other. Results of the `REPuter` analysis show that the whole contig mK11-48H20 (B) was contained in one large piece of contig mK5-185K22 (A). The vertical white line in the annotation, and crossing the repeat graph represent small gaps in the genomic sequences. Consequently, contig mK11-48H20 was discarded from the mouse genomic assembly.

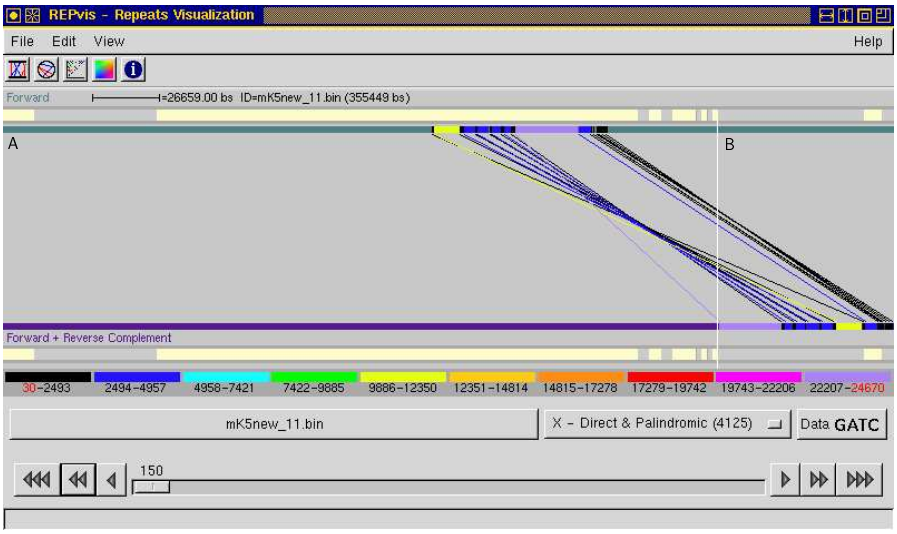

Figure 2.7: Assembly check of mouse contigs mK5-185K22 and MK11-48H20. The boundaries of the colored bars in the annotation graph represent the localization of small gaps in the sequence. The second sequence, mK11-48H20 (B), is completely contained in the larger sequence mK5-185K22 (A), indicating that it can be excluded from the subsequent assembly step.

2.5.3 Localization of Low Copy Repeats Associated with Human Malformations

Diseases that are caused by chromosomal rearrangements involving one to several megabase pairs are generically called *genomic disorders* [76]. Those rearrangements can generate interstitial or terminal deletions, duplications or even unbalanced translocations. Such reorganizations may result in imbalanced gene dosage, leading to several human malformations and syndromes associated with the consequent haploinsufficiency of at least some genes in the affected region. Genomic rearrangements generating interstitial deletions are suggested to have a preferential site for recombination on chromosomes [106]. The existence of *repeat gene clusters* flanking such common deletion sites has been described for several syndromes [104]. These are called *low copy repeats* or, LCRs for short. The presence of LCRs suggest that they function as breakpoints leading to homologous recombination. There are two models that explain the results of such rearrangements. The first model involves modules of LCRs that are directly oriented to each other. This could lead to interchromosomal misalignments between two homologous chromosomes, resulting in an unequal crossing-over. The following consequences would be reciprocal deletion and duplication events from the sequence inbetween those LCRs (Figure 2.8). The second model regards to LCRs that are palindromic to each other. In this case, the repeated modules might form a “stem-loop”-like

structure, leading to either the inversion or the deletion of the intervening DNA present within the "loop" [104, 105, 78] (Figure 2.9).

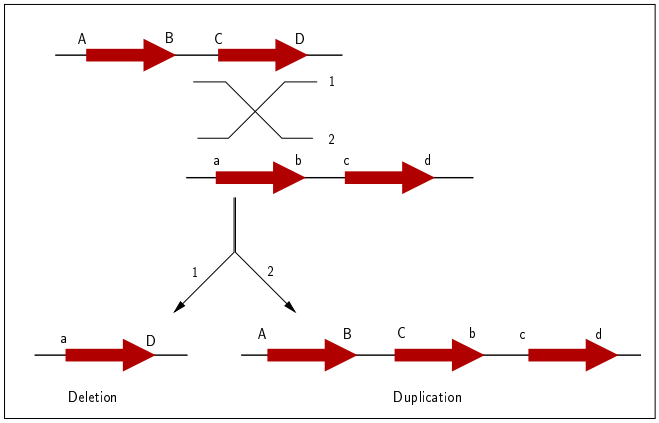

Figure 2.8: Mechanisms for chromosomal rearrangements. Model for interchromosomal rearrangement between LCRs with direct orientation to each other (indicated by thick horizontal arrows). Unequal crossing-over occurs resulting in deletion (1) and duplication (2) rearrangements. (Taken from [104] and [105])

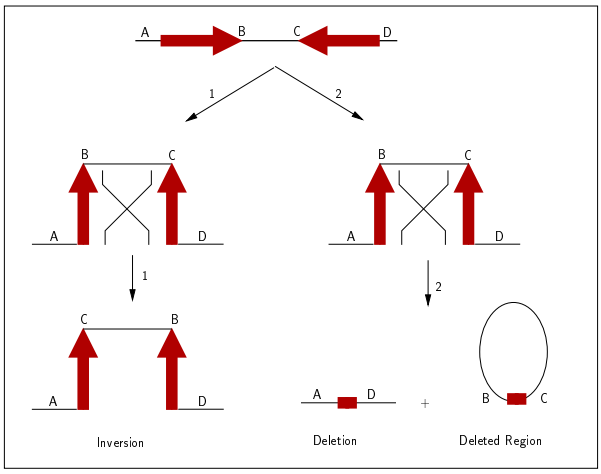

Figure 2.9: Mechanisms for chromosomal rearrangements (ctd.). Model for intrachromosomal rearrangement between LCRs with palindromic orientation to each other (indicated by thick horizontal arrows). The intervening sequence is susceptible for either an inversion (1) or a deletion (2). (Taken from [104] and [105])

It has been observed that phenotypes related to deletions are more severe than those related to duplications [11, 12]. In general, it has been assumed that haploinsufficiency

for at least some of the genes in the deleted region is responsible for direct effects on specific developmental processes. The human chromosome 22, for example, is very rich in gene content, although being the smallest of the human chromosomes. It has been reported in detail that several malignant diseases and developmental abnormalities are associated with genomic rearrangements of this chromosome [21, 30, 106].

In the beginning of the 90s, specific low copy repeat elements have been identified in the q11 region of chromosome 22 [54]. These findings suggested genomic instability of this part of the human genome. Most of the rearrangements observed in this chromosome refer to large 3 Mb deletions, causing various anomalies including mental retardation, like the very well known DiGeorge/Velo-cardio-facial Syndrome, localized at 22q11.2. This syndrome is characterized by craniofacial anomalies, heart defects and immunological deficiencies, besides learning disabilities and behavioral irregularities [31, 14, 33, 21, 104, 60]. The hypothesis that LCRs are the responsible elements for the deletion of this region was fortified by the localization of those repeats flanking the 3 Mb Typically Deleted Region (TDR) (Figure 2.10).

Some smaller variant deletions have been reported, which breakpoint regions have been localized within the 3 Mb TDR. Inside this large segment of chromosome 22, 2 more copies of the LCRs were found, explaining the occurrence of patients with distinct, smaller deletions. Analyzing the graph in Figure 2.10, 87% of the patients present the large, 3 Mb deletion. The smaller deletions, resulted from recombinations involving the

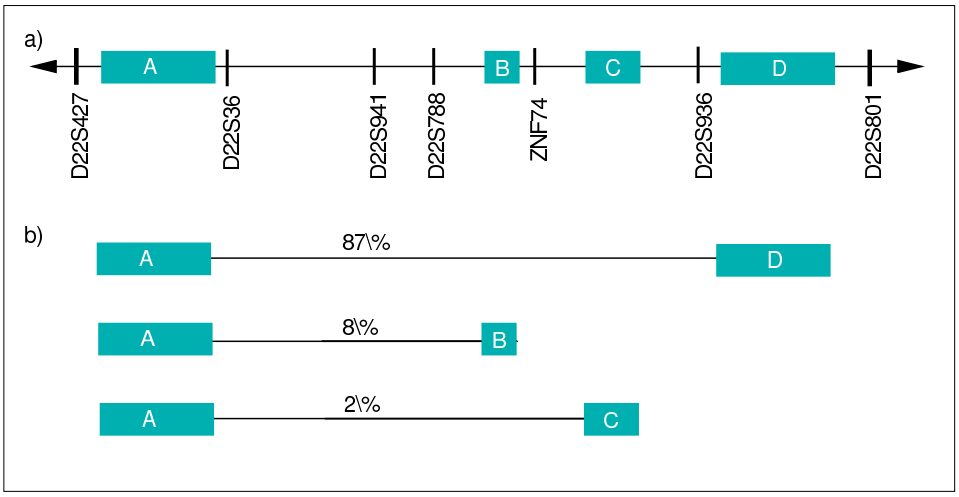

Figure 2.10: Low copy repeats localization on human chromosome 22. a) The 3 Mb Typical Deleted Region involved in DiGeorge/Velo-cardio-facial syndrome is shown between the markers D22S427 and D22S801. The filled blocks represent the LCRs responsible for the rearrangements leading to different deletions. b) The percentage of patients identified with those deletion boundaries. The remaining 3% represent unique deletions. (Taken from [106])

nested LCRs, are found in 10% of the patients analyzed by T. Shaik *et al.* [106]. The findings of the same-sized deletions in the majority of the patients points to a specific mechanism which gives rise to most of those structural rearrangements. Deletions resulted from interchromosomal recombinations on this chromosome's region are frequently seen, although the reciprocal duplication event is rarely observed [31].

Results of the 3 Mb TDR Analysis with `REPuter`

Before the sequencing completion of the human chromosome 22, `REPuter` analysis done in cooperation with Dr. Schleiermacher (Artemis Pharmaceuticals) revealed an interesting repeat structure in one of the draft contigs (Figure 2.11). The positions on the DNA strand confirmed that one main block was repeated three times, constituted of embedded direct and palindromic repeats. As this analyzed subsequence was only 240 kb long, it raised the question whether the repetitive pattern generated by the sequence of this repeating module would further extend itself to the left and right on the chromosome.

After the publishing of the whole sequence of chromosome 22 [29], its entire sequence has been analyzed with `REPuter`, searching for all direct and palindromic exact repeats of 300 bp minimal length. The overview of the repetitive structure of this chromosome is depicted in the *REPvis* visualization graph in Figure 2.5. The specified search adjustments generated a relative homogeneous repeat pattern, except for a quite confusing structure in the first quarter of the sequence. This subsequence was extracted and searched for repeats using a lower threshold. Figure 2.12 shows an overview of this

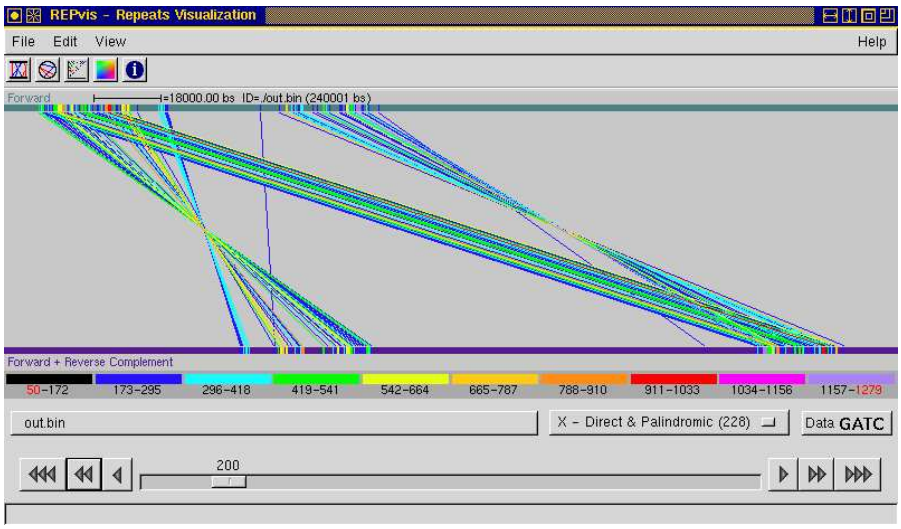

Figure 2.11: Repeat structure of a 240 kb region on human chromosome 22. The direct and palindromic repeats of 200 bp minimum length are displayed in the repeat graph. The main sequence repeats itself sometimes direct, others reverse-complemented (seen in the visualization by the typical cross-pattern of palindromic repeats).

area, revealing an interesting net-like structure of direct and palindromic repeats.

It turned out that the small module in Figure 2.11 represents only a substructure of a larger repeat pattern on this region of chromosome 22, as it had been hypothesized. The main sequence module is repeated four times, being comprised of smaller repeated units which are present in direct and palindromic orientation to each other. Strikingly, this zoomed region is exactly 3 Mb long, suggesting that those repeated modules found with REPuter could refer to the LCRs on chromosome 22. Analyzing the LCRs scheme described by T. Shaik *et al.* [106] in Figure 2.10, the repeated blocks located at the end-points of the TDR are the largest in size. The authors also comment that they are the most similar to each other when comparing their sequence to the inner LCRs. Moreover, the four LCR are placed between the markers D22S427 and D22S801. The localization of these markers in the DNA region in Figure 2.12 confirmed the hypothesis that the repeat structure found with REPuter refers to the 3 Mb Typical Deleted Region on chromosome 22, which causes DiGeorge/Velo-cardio-facial syndrome. The *REPvis* visualization of the repetitive pattern in question is overlapped by the LCRs graph from T. Shaik *et al.* [106] in Figure 2.13.

Generalizing the observations described above, an analysis of the repeat structure of different chromosomes using REPuter is helpful to identify such breakpoint regions regarding the localization of low copy repeats, without any experimental approach.

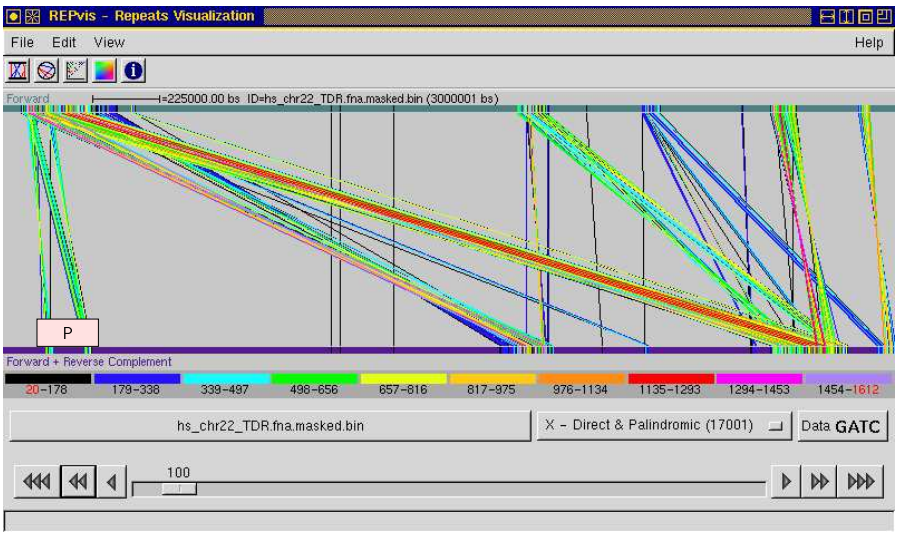

Figure 2.12: Net-like pattern of low copy repeats on human chromosome 22. The repetitive structure extends over a 3 Mb region on the chromosome. Displayed in the graph are direct and palindromic repeats of minimum length 100 bp, with at most 2 errors. The net-like pattern reveals the typical structure of low copy repeats. The block "P" corresponds to the structure previously shown in Figure 2.11.

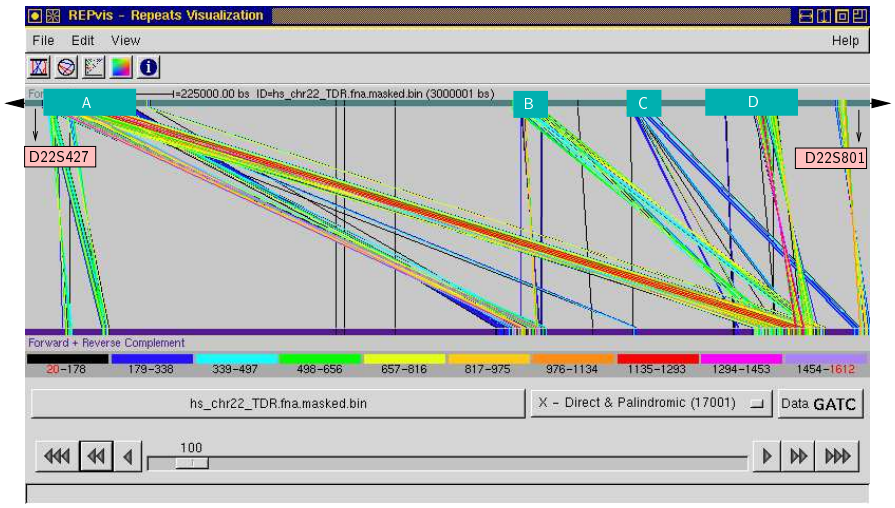

Figure 2.13: Net-like pattern of low copy repeats on human chromosome 22 (ctd.). The graph in Figure 2.12 is overlapped by the one in Figure 2.10, demonstrating the correspondence of the schematical view of the LCRs localization and their identification at the sequence level.

2.5.4 Identification of String Uniqueness

It is well known that chromosomal endings, regions around the telomeres, have a significant role in genetic disorders and genomic rearrangements. In vertebrates, these telomeres consist of a tandemly repeated sequence of (TTAGGG)n, which can extend from 2 up to 15 kb in length [46]. Adjacent to this sequence, repetitive DNA constituting the subtelomeric region is found in a number of other chromosomes. However, the subtelomeric sequences do have many unique and functional genes, too. This could be one reason for many mental and other disorders generated by chromosomal rearrangements in these regions. The detection of such rearrangements is made by Fluorescent *in Situ* Hybridization (FISH) [89].

Hybridization techniques are very effective tools in molecular biology, used in a variety of experiments, including pre-natal diagnosis and microarray technology. These techniques make use of nucleic acid probes to detect their complementary targets present in biological fluids or tissues. The success of hybridization experiments depends on the specificity of the probes.

In collaboration with Dr. Wirth and Dr. Ehling (Bielefeld University), we were interested in finding BACs for such *in situ* hybridization in the *Down-Syndrome Critical Region* (DSCR) of human chromosome 21q22.2 [33]. In order to search for rearrangements in the subtelomeric region of this chromosome, we established probes localized at the end of contig 4[3] for FISH analysis. As the last 5th contig is only 26076 bp long,

[3]GenBank accession number: NT_003534

taking exactly this region could produce undesired cross hybridization with other chromosomes, considering that it is very near to the telomeric region. The establishment of DNA probes for the FISH experiments was accomplished with the utilization of BACs.

Specific BAC clones corresponding to the region of interest on chromosome 21 were searched in a Human BAC Library containing *Upper* and *Plate* pools (GenomeSystems, Inc). The search begins with the amplification via Polymerase Chain Reaction (PCR) of the DNA contained in the *Upper* pool with primers that are specific for the defined region. The coordinates of the positively amplified well reveal the corresponding *Plate* pool holding the next set of BACs to be searched. Further PCRs are done until the last step, *Down-to-the-Well*, where the well inclosing only the BAC in question is identified. To successfully perform each step, the primers needed for screening have to be as specific as possible for the region of interest. Targeting the probes to non-unique sequences result in cross-hybridization generating false-positive amplification, and, consequently, increasing the experimental effort.

Results of Unique String Detection with REPuter

We define *finding unique sequences* as the mathematical complement of *finding repeats*. As REPuter is not heuristic, identifying all repeats according to specified parameters, it can also be successfully utilized to solve the *unique sequence finding problem*. Assume that the probe should have length l and be unique up to k errors. Finding and discarding all repeats of length at least l with maximal k errors, the remaining fragments are guaranteed to be unique for substrings of length l and k errors everywhere in the original input sequence (Figure 2.14).

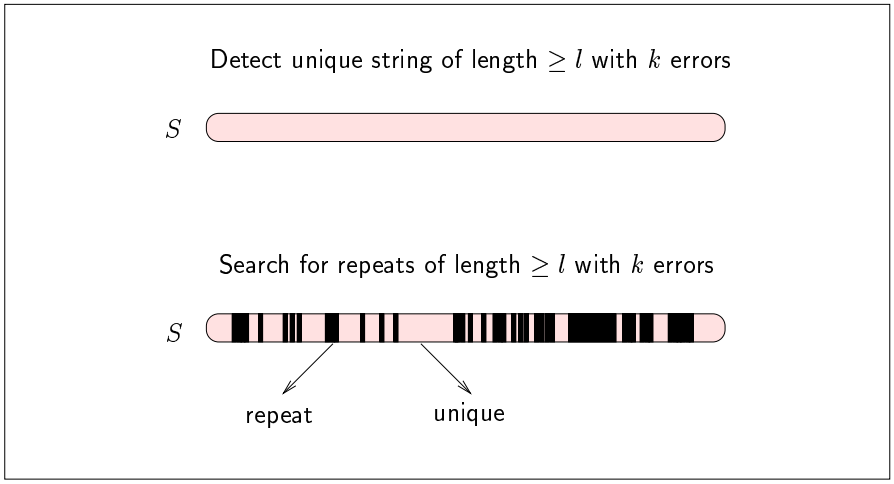

Figure 2.14: The unique sequence finding problem. As REPuter is not heuristic, finding all repeats of length l with at most k errors guarantees that the remaining sequences are unique for these parameters.

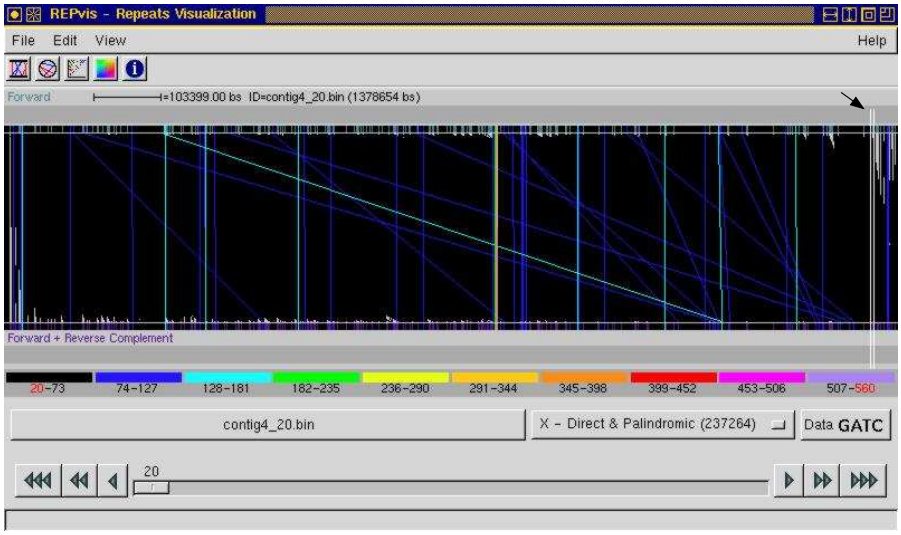

Figure 2.15: Identification of unique sequences. Direct and palindromic repeats of contig 4 in human chromosome 21. The sequence was searched for repeats of length at least 20 bp and an allowance of 2 errors. The black background reveals that the noise level is reached with this threshold. Even though, some regions free of repeated substrings can be identified, like the one between both vertical white lines (indicated by an arrow).

As primers are usually around 20 bp long, in our application example of chromosome 21 we searched the fourth contig for repeats of length at least 20 bp. The error allowance of 2 mismatches, insertions or deletions was chosen to guarantee the absence of mispriming of the primers taken from the unique sequences. This threshold reached the black background noise level, as it can be observed in Figure 2.15. Nevertheless, zooming into this graph, segments without any repetitive substrings can be localized (see Figure 2.16). These fragments were used for the primer design. In addition, in order to avoid setting primers in regions with repeated elements (see Section 2.3), these were masked out by submitting the sequences to `RepeatMasker` (Smit, A. & Green, P., unpublished). Finally, the primers were designed within the remaining unique sequences, using the `GeneFisher` program [98].

The usage of these primers in the BAC library screening produced very specific amplification, generating sharp bands in the gel (data not shown), and leading us to rapidly identify the BAC corresponding to the region of our interest on chromosome 21. The same approach was used to find BACs in the DiGeorge/Velo-cardio-facial syndrome region on chromosome 22q11. The fluorescent *in situ* hybridization experiments using those BACs as probes were run by D. Ehling in her PhD thesis [33]. Just to illustrate the FISH results, Figure 2.17 shows sharp fluorescent signals on chromosomes 21 and 22. Although the specificity of the primer probes contributed to the targeted selection of BACs, the absence of unspecific background or cross-hybridization in the FISH neither

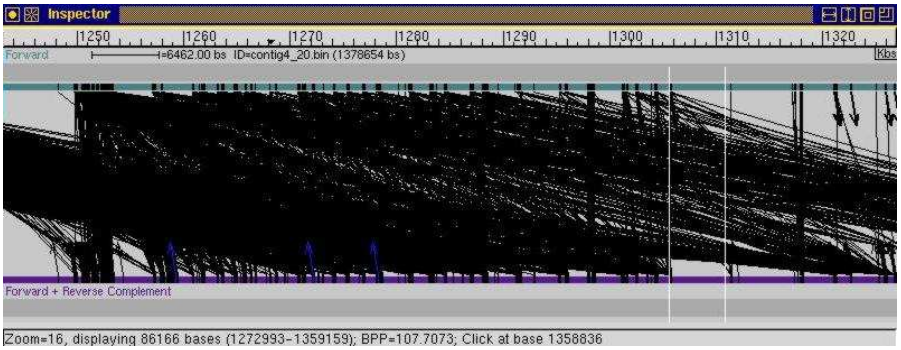

Figure 2.16: Identification of unique sequences (ctd.). Zooming into the repeat graph of Figure 2.15, we can observe the absence of repeated substrings in the segment marked between two vertical white lines. After extracting this subsequence, it was submitted to `RepeatMasker`, eliminating also repeated elements in the region. The remaining sequence is repeat-free and thus optimal for the design of specific primers.

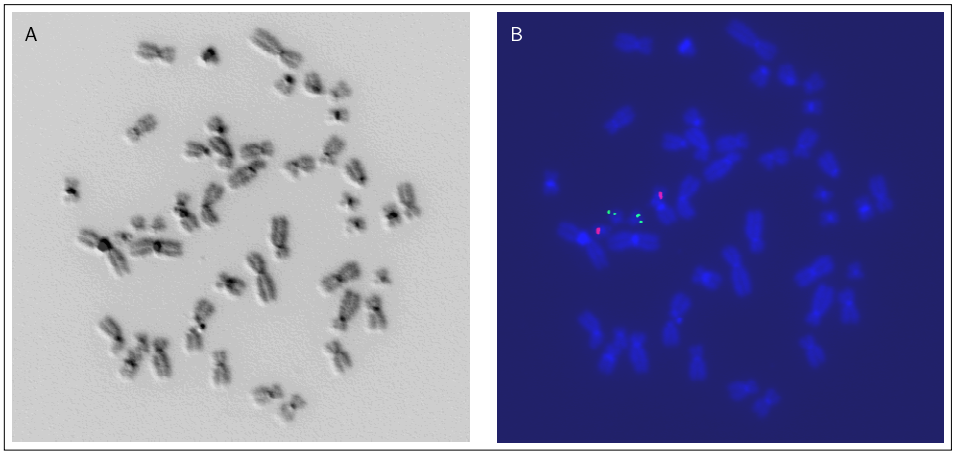

Figure 2.17: Fluorescent *in situ* hybridization on metaphase lymphocytes. The BACs used as probes were identified in a Human BAC Library by PCR screening. Red signals represent the specific hybridization on the human chromosome 22q11 between LCRs *A* and *B* from the graph in Figure 2.10. Green signals indicate the probe specificity to the chromosome 21q22, in the *Down Syndrome Critical Region*. A) The inverted picture. B) The picture with color filters.

gives an evidence about the amount of repeats in the BAC, nor is guaranteed by this explained method. This kind of approach to set specific primers in repeat-free regions showed to be also economically very advantageous, considering that the upper pools from GenomeSystems, Inc. supply enough DNA for testing 25 or 50 different primer pairs. In our case, only one pair per sequence was necessary to get specific amplification.

2.5.5 Matching Complementary DNA or Expressed Sequence Tags onto Genomic Sequences

Expressed Sequence Tags (ESTs) are generated by partial DNA sequencing on complementary DNA (cDNA) clones. As short stretches of transcribed regions [2], they are used as a rapid and reliable method for gene discovery in the genome. They serve as markers for analyzing the localization and expression of known and unknown genes. Not only by experimental approaches, but also through local similarity search, ESTs and cDNAs can be easily localized in genomic sequences. Given a cDNA sequence, for instance, it can be *unspliced* onto its corresponding genomic sequence using `REPuter`, by concatenating both strings and searching for repeats. This strategy allows the user to investigate the exon/intron structure of the respective gene as well as to check for the gene structures predicted by other software tools (see also Subsection 2.5.6).

In our application example, we looked for a mouse gene similar to the human *KIAA0903* gene (see Figure 2.18). In humans, this gene is localized on chromosome 2, in a region known to be homologous to the mouse chromosome 11 [45, 95] (see Chapter

Figure 2.18: Matching cDNAs onto genomic sequences. Human *KIAA0903* cDNA (A) is concatenated with the homologous mouse region on chromosome 11, contig AC091423 (B). Looking for a mouse gene similar to the human *KIAA0903*, the sequence was searched for all direct repeats of length at least 100 bp. As coding regions are more conserved between both species than noncoding ones, up to 2 errors were allowed. The unsplicing of the *KIAA0903* cDNA onto the mouse genomic sequence shows the exons separated by very long introns. The 5' and 3' ends of the gene are either missing in this contig or are not conserved enough to be detected with the chosen parameters.

5). The sequence of the cDNA *KIAA0903*[4] represented the string S_1, and was concatenated with the corresponding mouse contig[5] S_2, in order to recover the cDNA sequence in the mouse genome. Using *REPfind*, the concatenated string S_1S_2 was searched for direct and palindromic repeats of minimal length 100 bp. We allowed 2 mistakes in the finding of repeated substrings. This threshold shows enough error allowance to recover the parts of the cDNA that were contained in the searched contig. The resulting repeat graph is depicted in Figure 2.18. Making use of the annotation graph, a vertical white line was utilized to visualize the separation of the cDNA sequence (A) from the mouse contig (B). As it can be clearly observed, the exons of *KIAA0903* gene are separated by very long introns in the mouse genome. The 5' and 3' ends are not shown to match anywhere in the contig. This indicates that those ends are either missing in the mouse region or not enough conserved to be identified under the used threshold.

2.5.6 Comparison of Gene Structure - Part I

Comparative genomics is one of the major reasons for sequencing whole genomes of different organisms. Throughout evolution, vital genes and regulatory regions have been conserved to guarantee the organism's basic functions. The mouse is a very well known model organism for studies of human biology and medicine. The access to its genomic sequence allows researchers to make important discoveries in the regulation of human genes based on common structures, and mechanisms shared with mouse genes (see Chapter 3). Today, these similarities across species can be identified at the sequence level with computational programs like REPuter.

Besides the procedure of concatenating different sequences and looking for repeats, REPuter allows us to consider many grades of similarities between the repeated substrings by controlling their error rates. This is the approach used in the first part of the example application *Comparison of Gene Structure*. We considered the genomic sequence of the 5' region from *Mus musculus Peli1* gene[6] as string S_1. The genomic sequence covering the complete human *Peli1* gene[7] S_2, was concatenated to S_1. Matches of least length 20 bp and 2 errors were searched by *REPfind*. Figure 2.19 depicts all matches above 35 bp in the concatenated string S_1S_2.

The annotation graph is useful to separate S_1 from S_2 with a vertical white line for a better visualization. Moreover, in the first annotation line, the exons predicted by the program GENSCAN [15] are represented by colored bars. One of the predicted mouse exons, (a), coincide with a prediction in the human genomic sequence (a'). The other intersequence match (b) also corresponds to a predicted exon in mouse, but does not show the same prediction accuracy in the human region. However, it has been demonstrated that the match refers to a correct exon, by analyzing the unspliced cDNA

[4]GenBank accession number: AX030068
[5]GenBank accession number: AC091423
[6]GenBank accession number: AC091421
[7]GenBank accession number: NT_005326

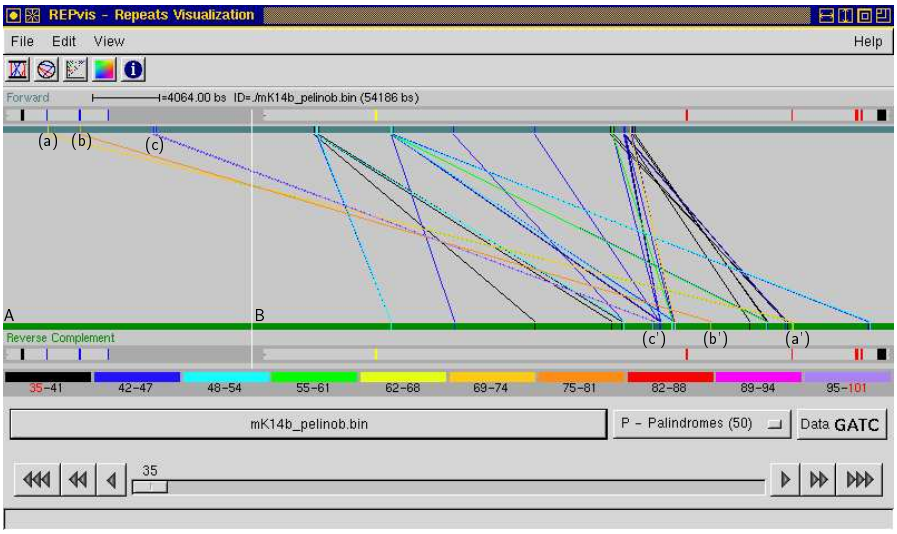

Figure 2.19: Comparison of gene structure -Part I. The concatenation of the mouse genomic sequence containing the 5' region of *Peli1* gene (A) with the human corresponding genomic sequence (B) is considered. The repeat graph shows palindromic repeats of length at least 35 bp with at most 2 errors. The annotation lines represent the `GENSCAN` predicted exons in colored bars and a vertical white line separating the mouse (A) from the human (B) sequence. Analyzing only the intersequence matches, we identify similarities and conserved regions between the *Peli1* gene in both species. Matches (a) and (a') have been correctly predicted by `GENSCAN`. Matches (b) and (b') represents less accuracy of the gene prediction tool regarding the human sequence. Matches (c) and (c') was not predicted at all, although Figure 2.20 confirms that is indeed an exon.

onto the genomic region (data not shown). An even more striking observation regards to the match (c). It refers to a very well conserved string between mouse and human (of about 98% similarity) which was not detected by the gene prediction tool in either of the organisms. Again, the strong conservation and the matching with the corresponding cDNA sequence suggests that it is indeed an exon. Furthermore, wet-lab experimental analysis done by Dr. Fuchs (Düsseldorf University) confirmed our expectations. The intention was to knock-out the *Peli1* gene, generating a mutant mouse for this gene. The sequence of the target exon chosen for the experiment was delivered to be compared with the gene structure of Figure 2.19. The sequence in question was matched with the mouse genomic region containing the *Peli1* gene. Figure 2.20 shows the exact localization of the target exon, where the letter (c) is placed in the region corresponding to match (c) in Figure 2.19.

The prediction of coding sequences is very dependent upon the gene identification algorithm used to infer what segments of the genomic sequence actively code for genes [118]. With the approach described above, `REPuter` provides a simple plausibility check

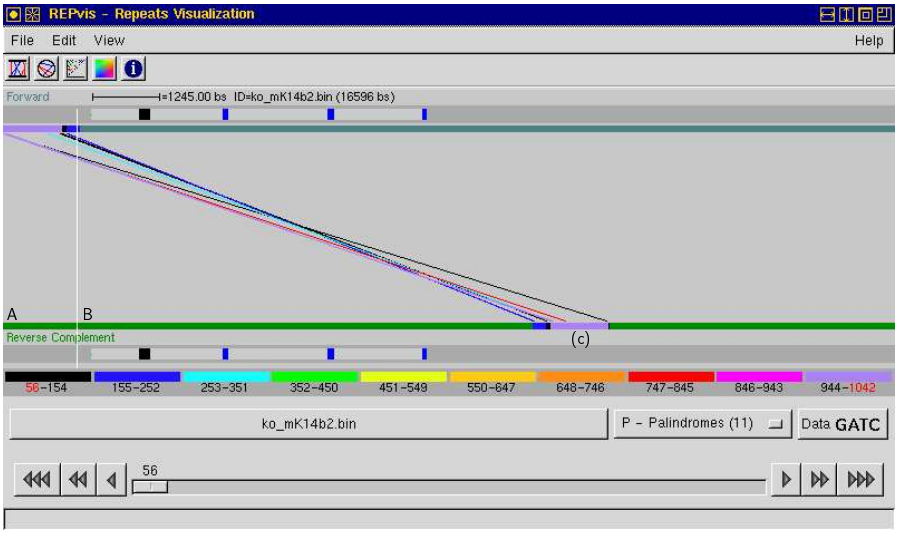

Figure 2.20: Comparison of gene structure - Part I (ctd.). The sequence of the target exon chosen for the knock-out experiment was concatenated with the mouse genomic sequence containing the 5' region of *Peli1* gene. The palindromic repeats of length at least 56 bp are displayed in the graph, localizing the exact position of the exon in question. The letter (c) indicates the same region of match (c) in Figure 2.19, which has not been identified as an exon by the gene prediction tool.

for gene structures predicted by other software tools. Besides, it delivers an accurate and significance-dependent analysis of similarities between species, either in protein coding or regulatory regions.

2.6 Limitations

A very important technical comment regarding the comparison of gene structures using REPuter is the inconvenience of concatenating sequences. The program treats the concatenation of S_1S_2 as a new string Z. Without the annotation graph, it would be very difficult for the user to recognize if an interesting match is localized actually in S_1 or in S_2. Even using annotations to visually separate both sequences, the absolute positions of the matches are concealed by the concatenation and consequently depending on the length of S_1. Furthermore, the non-intended repeats found within S_1 and/or S_2 are also shown in the repeat graph, generating superfluous data for this kind of application. The visualization is obscured, causing difficulties when interpreting the results of the comparison between species. In Chapter 4, we present some solutions for this problems and come back to the analysis and comparison of the *Peli1* gene structure in mice and humans (Section 4.4).

2.7 Summary

The term *repeat analysis* has a broad range of biological interpretations. `REPuter` is a suitable program for the computation of repeated substrings in large DNA sequences, delivering the necessary data for investigating the repeat structures of such strings. Although `REPuter` satisfies all criteria for a tool which systematically searches for repeats, being applicable in a variety of biological problems, it also presents limitations. Chapter 4 describes how the challenge to overcome the weaknesses of `REPuter` is successfully achieved.

3 Comparative Genomics

Questions in sequence analysis change in the course of time. In the past, researchers were interested in finding out what genes, and therefore proteins, are shared among organisms. Sequencing the entire genomes of several bacteria and fungi have shown to be useful for identifying such similarities. These genomes consist primarily of coding regions, with little sequence between genes and almost no introns within a gene. This means that a large amount of genetic information is present in relative small segments [55]. This concept can not be easily transfered when sequencing larger genomes of more complex organisms. They usually have much more DNA between genes and introns interrupting coding sequences. So the initial interest in finding common proteins among species brought the question whether it was useful to determine the sequence of all that noncoding regions. This aspect led some researchers to argue that the sequencing of cDNAs, representing the vast majority of the coding information content in the genome, should have priority over genomic sequencing [10].

The concentrated efforts to determine the sequence of all reverse transcribed mRNAs and ESTs have generated rich and useful databases, but they also have limitations. The comparison of genomic regions with ESTs is helpful for identifying many exons. Although, it is difficult to comprise the complement to an entire mRNA, so that the assignment of all exons can not be obtained only from EST information [2]. Moreover, the observation that closely related species show very high similarities in those coding sequences led researchers to redirect their questioning in sequence analysis. The interrogation now was regarding the interspecies similarities and differences in the regions inbetween coding sequences. Today, we know that many differences between organisms lie in the way genes are regulated. As control and regulatory sequences are in the noncoding regions of the genomes, the cDNA sequencing missed much valuable information from intronic and intergenic regions.

This chapter gives an overview of the two genome sequencing projects considered in this work, human and mouse (Section 3.1). Afterwards, in Section 3.2, some insights into comparative genomics are given, describing general concepts of sequence conservation. A few literature examples show how approaches using comparative genomics can be valuable for finding regulatory elements in genomes, and finally, three of the most popular computational tools for comparative genomics are presented.

3.1 Genome Sequencing Projects

It is not so long ago (1995) that we witnessed the completion of the first free-living organism's genome sequence - the bacterium *Haemophilus influenzae Rd* [39]. Since then, through large-scale DNA sequencing efforts of public and private organizations, further complete genomes have been published: about 50 different bacteria (in addition to many different strains), 11 archae and 11 eukaryota.

In our work, we focus mainly on human and mouse genome sequencing projects and their contributions to the scientific community. An overview of both projects is given in the following Subsections. We have also developed a web-site containing links to the mouse and human genome projects, described in Appendix A. The list of corresponding web-sites is outlined in Appendix B.

3.1.1 The Human Genome Project

The Human Genome Project (HGP) is an international and collaborative research program with the goals to sequence the whole genome of *Homo sapiens*, map all genes and understand how they work. To achieve their purposes, the HGP includes the development of new technologies for genomic analysis, and train scientists to use the tools and resources generated by the program in a multidisciplinary approach. Furthermore, the HGP is also supposed to examine the ethical, legal and social implications of human genetic research. Numerous universities and several research facilities from all over the world are concentrating their efforts, participating the so called *International Human Genome Project Consortium*. The deciphering of the human genome occurs in three major ways: determining the sequence, i.e., the order of all 3 billion bases in the genome; mapping the locations of genes; and producing linkage maps, through which inherited traits can be tracked over generations.

In 1998, a five-year plan to sequence the entire human genome was announced, and in February 2001, the draft of 90% of the whole sequence was published in Nature [70]. A complete and more accurate genome sequence is available since April this year (see the Sanger Center web-site in Appendix B). A detailed analysis of this final version of the human genome sequence is being done by the HGP Consortium. The main surprise accompanying the draft sequence publication was the small number of human genes, about 30000 to 40000 protein-coding genes, relative to what researchers had expected, which ranged from 50000 to as many as 140000 genes [70]. This large discrepancy led to the conclusion that the human genome also presents much more repetitive elements than it was originally believed. Furthermore, these results imply that the genome contains many additional features in noncoding regions that may play a role in regulating the expression of those genes.

Since the beginning of the Human Genome Sequencing Project, researchers were aware of the challenges they would face in deciphering the information contained within these DNA sequences. A clever resolution was to include the sequencing of other organisms'

genomes in parallel to the human genome sequencing. The comparison of the human genome with the genomes of different species brings insights into evolutionary changes and provides a powerful tool for a better understanding of structure and function of human genes.

3.1.2 The Mouse Genome Project

The laboratory mouse offers great opportunities in studying the genetic causes and pathological progress of diseases, because of its small size, high fertility rate, and experimental manipulability. The fact that mice and humans share almost the same set of genes indicate that laboratory experiments with mice can provide insights into the genetic role of disease susceptibility in humans. These features explain why the laboratory mouse is the most important model organism and is widely used in biomedical studies (see Chapter 5). This was also one of the decisive arguments to sequence the mouse genome.

Similar to the Human Genome Project, there were joint efforts of an international team of scientists to sequence and assemble the mouse genome. In December 2002, a high-quality draft sequence was published and made publicly available [116]. The mouse genome contains 2.5 billion bases, almost 20 percent shorter than the human genome. In corresponding conserved regions, the mouse genome is about 10% smaller than the human genome. This is argued to be due to the lower content of DNA repeated elements in mice [83]. Toyoda *et al.* [112] confirmed the lower fraction of repetitive elements in the mouse, while analyzing the homologous segment to the human chromosome 21 *Down Syndrome Critical Region*. In this study, the authors quantified the 'uniqueness' in these regions, that is, the remaining amount of base pairs after the exclusion of all interspersed repeats and other repeated elements. They observed that the 'uniqueness' in humans represented 1.04 Mb, whereas the same region in mouse presented 0.97 Mb 'uniqueness'. The relative paucity of repeated elements in the mouse genome might be explained, in part, by reduced transposition activity of repetitive elements [86].

Since humans and mice diverged, about 75 to 80 million years ago, there has been a considerable shuffling of the DNA order within and between chromosomes. Still, gene order between both genomes is often preserved over large regions. At the nucleotide level, 5% of the genome contain segments that are conserved between mouse and human. This proportion is about three times as much as can be explained by protein-coding genes alone. These observations imply that there are many other features under selection for biological function, such as regulatory elements, chromosomal structural elements or untranslated regions. In this way, the comparison of genomic sequences between different organisms demonstrates its importance, supporting the initial goals of the genome sequencing projects.

The development of an integrated physical and genetic map for the mouse, the generation of additional mouse cDNA resources, and the completion of the mouse genome's sequence are goals expected to be achieved by 2008.

3.2 Sequence Conservation

From the outset of genome sequencing, it was clear that a single organism's genome taken in isolation does not reveal much by itself. The whole genome and its content need to be analyzed in comparison with other species, in the known phylogenetic context of evolution [20]. The evolutionary history of genomic rearrangements, mutations and resulting functional specializations of genes offer an insight into similarities and differences between genomes. Besides, this genome comparative approach highlights the unique features of individual genomes. Regarding gene prediction, two main approaches were used in the *pre-genomic* era: the *intrinsic* methods (or *ab initio*), which use statistical features to distinguish coding from noncoding regions, and the *extrinsic* methods, which try to find similarities between genomic sequences and known proteins. Both approaches are limited, since they rely on information derived from already known genes [96]. With the availability of the human genomic sequence, some authors have began the interspecies gene analysis by comparing full-length mouse cDNAs with human genomic sequences [64]. The joint efforts to sequence the mouse genome permitted those researches to accomplish comparisons between both genomic sequences. At this point, comparative genomics complements the gene prediction methods described above, as it is not biased towards finding genes similar to known ones.

In the beginning of the 90's, molecular biologists made use of sequences of multigene families for large-scale DNA comparisons. They represent regions with high information content and defined size. Those were the first steps towards whole genome comparisons. At that time, researchers had already observed different patterns of conservation between organisms. High levels of sequence similarities were found in coding regions, while noncoding regions showed a more mixed pattern of conservation. This means that conserved and divergent sequences were localized adjacent to one another. These observations suggest different divergence rates within noncoding sequences, and therefore, a mosaic model of genomic evolution [65].

Today, it is known that sequences coding for proteins are more conserved between species than noncoding regions. In general, intronic regions have weak identity, where 30% similarity is near the background sequence identity rate for random sequences [6]. Nevertheless, confirming the mosaic model of genomic evolution proposed by B. Koop [65], there are also sequences in noncoding regions that have been under evolutionary pressure as they are in coding sequences. This kind of positive selection is discussed to be either due to functional constraints or insufficient divergence time [27, 41, 115], presuming that selective pressure causes regulatory elements to evolve at a slower rate than that of nonregulatory sequences in the noncoding regions. The high similarity of both biology and sequence between human and mouse led researches to use the mouse genome as a tool for genome comparison (Subsection 3.2.2). However, it is important to realize that different regions of the human genome will have a different background level of conserved sequences between humans and mice. When choosing threshold criteria for implying that an element has been conserved because of functional constraints, it is

important to take into consideration the background level of similarity for the sequences being compared (Frazer K., personal communication). For this reason, with the sequence of genomes of many organisms available, comparisons between more than two species are also successful in finding regulatory elements. This multi-comparison strategy covers a broader range of evolutionary relationships. In general, sequences which have been conserved between species in noncoding regions are called *conserved noncoding sequences*, or CNS for short. Chapters 5 and 6 discuss more about CNS and regulatory elements.

This section describes how cross-species sequence comparison is applied to investigate conservation during evolution between related species. First, in Subsection 3.2.1, terminologies used in sequence comparisons are defined. Secondly, some recent contributions described in the literature are delineated in Subsection 3.2.2. Finally, the most commonly used computational tools for local and global alignments in sequence comparison are shortly introduced (Subsection 3.2.3).

3.2.1 Definitions

Comparative genomic approaches use some specific terms to describe similarities between evolutionary related gene sequences and chromosomal segments in different species [38, 70, 45, 42, 51]. The appropriate terminology is defined as follows:

Homology Genes that are derived from a common ancestral gene are called *homologs*, and the level of similarity in their sequences usually reflects the divergence time.

Orthology When genes in different species are homologous, and they have emerged by a speciation event, they reflect the history of the species and are called *orthologous*.

Paralogy Homologous genes can also be generated by the duplication of a chromosomal segment, rather by a speciation event, producing then *paralogs*.

Similarity Different from homology, similarity is what we can measure from the alignment of sequences or structures. It may be used as evidence of homology, but it does not imply homology.

Synteny and Conserved Synteny *Synteny* is only relevant within a species, as it refers to two or more genes located at the same chromosome. *Conserved synteny* indicates that at least two genes that reside on a common chromosome in one species are also located on a common chromosome in the other species (regardless to their order).

Conserved Segment If multiple orthologs in the conserved syntenic region are found in the same order in both species, the genomic intervals are referred to as *conserved segments*, or *conserved linkages*.

3.2.2 Cross-Species Sequence Comparison Examples

The idea of applying comparative genomics to identify potential regulatory regions has been under development since the sequencing of the organisms' genomes. Several authors have described the effective detection of control and regulatory sequences by cross-species genomic comparisons [55, 49, 90, 75, 88, 63, 58]. Their approach made use of in silico techniques to recognize conserved regions, but their functional evaluation has always been done with considerable wet-lab experimental efforts. Thus, the joint work of computer scientists, biologists and wet-lab researches is essential to achieve concrete results.

Loots *et al.* [73] applied a comparative sequence-based approach to identify regulatory sequences involved in controlling the expression patterns of the interleukin (IL) 4, 5 and 13 on human chromosome 5q31. This region was known to be conserved in humans and mice. From 245 observed conserved elements fitting the empiric criteria of 100 bp minimal length and 70% identity, 90 were defined as noncoding. The largest among those CNSs, about 400 bp long with a percent identity of 84%, was located between IL13 and IL4. After several wet-lab experiments, the sequence was confirmed to be a regulatory element, which regulated not only IL13 and IL4, but many other genes in that chromosome 5q31 region.

Another example in which regulatory elements have arisen from the identification of conserved regions concerns the human gene SOX9. It is located at chromosome 17q24, and encodes a transcription factor of the SOX family of proteins. It has been described that haploinsufficiency for SOX9 causes campomelic dysplasia (CD), a skeletal malformation syndrome [8, 40]. Some CD patients with intact SOX9 alleles showed rearrangements in the vicinity of SOX9, pointing to a possible disruption of the *cis*-acting regulation of the gene, caused by translocations and inversions. The genomic region of human and mouse SOX9 was compared at the sequence level [4]. Many similar noncoding regions were found, so that it was unlikely that all of them would have functional significance. For a better focus on conserved regions with more functional relevance, the homologous region in *Fugu rubripes* was also compared. This procedure takes the advantage that the pufferfish is evolutionary more distant from humans than mice. This was one of the first approaches of multi-species sequence comparison, assuming that the most essential regulatory regions are spared from sequence divergence over time. The analysis revealed five conserved elements, E1-E5, up to 290 kb 5' to human SOX9 and 18 kb 5' to *F. rubripes* SOX9. Transgenic experiments indicated that elements E3-E5 are candidate enhancers for the expression of SOX9 in specific tissues, once translocation breakpoints of CD patients result in the separation of those elements from SOX9.

Recently, researchers from Perlegen Science and Lawrence Berkeley National Laboratory, USA [41], have developed a promising approach for high-throughput comparisons of human genomic sequences with the DNA of multiple species. The technology was based on high-density oligonucleotide arrays, on which coding and noncoding regions of human chromosome 21q were represented. Those arrays were hybridized with labeled syntenic

mouse and dog bacterial artificial chromosome sequences to identify evolutionary conserved regions. Based on the array data, they used an empirically derived criterion to define a conserved sequence as a 30 nucleotide window with $\geq 60\%$ conformance, i.e., similarity. As the sequences on the chip were nonrepetitive, it remained to classify all conserved sequences detected on whether or not they overlapped known exons. It turned out that less than half of the conserved sequences on chromosome 21 between mice and humans constituted known exons. Moreover, human-dog analysis identified considerably more conserved elements (both exonic and non-exonic) than the human-mouse analysis. Some of them were represented in all three species. The authors concluded that elements conserved in two species that are also conserved in a third species are more likely due to functional active conservation rather than shared ancestry. Furthermore, classifying the human-mouse elements based on length, the probability that it was also conserved in the dog increased with the increase of the element's length. All in all, these results indicate that comparison of human sequence with the DNA of multiple species will be valuable for generating a list of potential functional elements in the human genome.

The three literature examples of cross-species comparison described here show only a fraction of the wide range of different approaches for comparative genomics. From pairwise alignments to pairwise alignments between all pairs of sequences (i.e, *multiple pairwise alignments*), passing through expensive high-throughput array experiments, all of them indicate how important sequence comparison approaches are for identifying gene regulatory elements.

3.2.3 Commonly Used Sequence Comparison Tools

The comparison of orthologous sequences is usually done by an alignment of the conserved segments. An alignment is defined as a mapping of one DNA sequence onto another evolutionary related DNA sequence in order to identify regions that have been conserved [42]. There are basically two ways of aligning two sequences: either locally or globally. Local alignments produce optimal similarity scores between subregions of both sequences. This method is used when long sequences have conserved synteny, but present genes in different order and/or orientation in the two sequences. Thus, separating the sequence in segments to be aligned (local) may be more accurate. On the other hand, if the purpose is to detect highly diverged, but orthologous regions in a long sequence, a global alignment would be more appropriate. It produces optimal similarity scores over the entire length of both compared sequences. Here we present `PipMaker` [101] as an example for a local alignment tool and `VISTA` [77], for a tool delivering global aligments. We also introduce `vmatch`, which also finds local similarities between two distinct sequences [66], presenting some advantages over the first and second tools.

`PipMaker` This tool is used to compare two genomic sequences identifying conserved segments between them. The underlying local alignment software, BLASTZ [100] is based on the *gapped BLAST* family of programs. Both input sequences have to be

delivered in fasta format. `PipMaker` should be used with masked sequences, i.e., marking the repeated elements so that they are ignored in the alignment. Thus, the output file from the `RepeatMasker` program is also delivered for the first input sequence. `PipMaker` does not call `RepeatMasker` by itself. In addition, the user can also provide a gene annotation file in order to specify the location the conserved regions relative to the exons in the first sequence. Another feature is the ability to depict regions of interest as shaded blocks, such as highly conserved elements with potential regulatory functions. This information is delivered in an *underlay* file. The inconvenience of this option is that if the user does not know where the interesting conserved regions are located, only after running `PipMaker` once, those regions may appear. Then the user has to run the program a second time, entering the *underlay* file to have all regions with specific identities shaded on the background.

`PipMaker` returns a percent identity plot (PIP) displaying the position, length and percent identity (from 50% to 100%) of each gap-free segment in the BLASTZ alignment. The first reference sequence is represented along the horizontal axis. The matches are depicted as horizontal lines within the plot, and their heights represent their identity. On top of the graph, repeated elements and exon positions are indicated according to the annotation files provided by the user. As it has been well observed by P. Chain *et al.* [16], the positions of the corresponding alignments in the second sequence is not shown in the graph. Those local alignments may be in a different genomic context as the reference sequence, or even in the opposite direction. Moreover, the graphic is static, without the ability to zoom into interesting regions or to link to other information or databases. The program is only available online, restricting the input sequences to at most 2 Mb length.

`VISTA` In order to compare two or more large genomic sequences (input length limitation is 4 Mb), C. Mayor *et al.* developed a tool called `VISTA` [77]. In this program, the multiple alignments are accomplished with the *GLASS* algorithm and more recently, *AVID*, also a global alignment program [9]. In case of more than two input sequences, a *multiple pairwise alignment* is followed by the output processing with intersection/union analysis to statistically identify common regions in all compared sequences. The length and percent identity thresholds are similar to `PipMaker`. Also similar to the previously described software is the ability to annotate the first input sequence based on gene annotation files provided by the user, with the difference that `VISTA` runs `RepeatMasker` automatically. As a second module of the software, `VISTA` provides the visualization of the conserved regions like PIP, but as a continuous curve instead of horizontal lines. Peaks in those curves represent segments of higher identity. Another similarity between both softwares is the fact that the graph is static, with position information available only for the first input sequence. Unlike `PipMaker`, `VISTA` has the ability of handling gaps in the collinear sequences, but it can not display rearrangements or non-collinear regions [16].

`Vmatch` The *REPfind* member of the `REPuter` family of programs has been adapted to the new era of comparative genomics in an improved tool called `vmatch`. In general, it is a index-based large scale matching software for sequence comparison. `Vmatch` copes with whole chromosomes or genomes, maintaining the genomic context of both reference sequences, even representing the relative orientation to each other. Exact conserved regions are found as well as degenerate ones, which include not only the option of gap-free alignments, but also insertions and deletions are allowed in the search. This is a biologically meaningful feature, as sequences can diverge not only by substitutional mutations, but also by insertions and deletions that can be detected among conserved segments. The output of `vmatch` can be visualized by `GenAlyzer`, an improved interactive interface of *REPvis*. In contrast to `PipMaker` and `VISTA`, the graph provided by `GenAlyzer` is not static. It can be zoomed into regions of particular interest, extracting sequence information by clicking on the conserved regions. The alignment information is not only present in percent identity, but in case of `vmatch`, the statistical significance is also taken into account, by means of E-values [26]. Moreover, it allows the user to directly submit matches for database searches, like BLAST or FASTA. `Vmatch` is described in detail in Chapter 4, showing all features which were implemented to overcome the limitations of `REPuter`.

3.3 Summary

Genomic information is being generated at an increasing pace, result of several genome sequencing projects. Thus, the scientific community is challenged to develop computational tools and approaches to convert the raw data into meaningful information. Since the beginning of the human genome sequencing project, researchers were aware of the difficulties in extracting biological data only from one very long sequence of 4 basic letters. Following the general approach to compare new things to known ones, to generate insights from this, molecular biologists and evolutionary biologists were interested in identifying sequence similarities between humans and other species. This was the key to the decision to sequence the mouse genome. The ability to compare genomes of different organisms gives valuable insights into genomic structure and evolution. Moreover, not only similarities in protein-coding regions, but also in noncoding segments have been detected. Making use of the *comparative genomics* approach, the joining efforts between computer scientists and biologists in the wet-lab successfully identified regulatory elements within those conserved noncoding regions. Besides more improved sequence annotation tools, new computational support is being developed to handle these new kind of information. *Comparative genomics* is the prevailing hope to get an accurate knowledge about similarities and differences between species, as well as a better understanding of the origins of genes, and how they are regulated.

4 Adapting REPuter to Comparative Genomics

The REPuter software, described in detail in Chapter 2, was presented as a tool used in a wide span of different biological applications for the simple reason of its versatility and flexibility (see Section 2.5). However, when searching for similarities between two concatenated sequences S_1S_2, the program showed some limitations which were discussed in detail in Section 2.7. Taking the idea of comparing gene structures, the same approach can be applied for comparing *genome* structures. By analyzing genomes of two distinct species, the computation of all repeats (in this case *matches* (see Section 2.3)) is too slow considering that whole genomes reach 3 to 4 billion base pairs. Every time the user changes the parameters for the match finding between the same sequences S_1S_2, first the suffix trees are reconstructed, then the users' criteria are adjusted. This leads to large space requirements and too long computational time, regards in which REPuter definitively had to be adapted to be used in large genomic scale analysis.

Another important adaptation for comparative genomics refers to the visualization of the *REPfind* output. We saw in Chapter 2 the nice interactive visualization provided by *REPvis*, where the calculated repeats are shown in a color coded repeat graph as well as the sequence annotations in the annotation graph (Section 2.4.3). Again, in the application Section 2.5.6, the sequence concatenation S_1S_2 brought some confusion in the visualization of the matches, as repeats found within S_1 or S_2 were also shown (Figure 2.19). The clever solution was to use the available repeat graph architecture and adapt it in order to display one sequence in the top line of the graph, and the other one in the bottom.

This Chapter delineates the main differences between REPuter and GenAlyzer in Section 4.1, presenting the new repeat graph in Section 4.2. Furthermore, additional matching tasks and output options are described in Section 4.3, and, in Section 4.4, *Comparison of Gene Structure - Part II*, the comparison between mouse and human *Peli1* genes is reconsidered using the new visualization.

4.1 GenAlyzer vs REPuter: the Main Differences

The second version of the REPuter family of programs is divided in four basic computational steps, instead of originally three. The *repeat-find engine* is now subdivided in 2 programs: mkvtree and vmatch. The option of selecting and sorting the repeats from the vmatch output is maintained in the program vmatchselect. GenAlyzer provides the visualization of the matching tasks via the new repeat graph and presents also other features, described later in Section 4.2.

Finding matches between two sequences S_1 and S_2 is facilitated by splitting the computational task into two different steps. The sequences do not need to be concatenated anymore, being handled as distinct strings. However, both sequences S_1 and S_2 may still consist of several concatenated sequences (like contigs in a multiple fasta file, for instance). The basic concept is to preprocess S_1, called the *database* sequence, or *database set* of sequences, into an enhanced suffix array (or virtual suffix tree) [66]. The advantage of suffix arrays over suffix trees is the enormous reduction of space requirement and a much faster processing [16]. Besides, several matching algorithms developed for suffix trees can be adapted to enhanced suffix arrays [1]. With this feature, `GenAlyzer` gets at least the same biological significance as `REPuter`, as it incorporates all functionality of `REPuter` with the underlying `mkvtree`, `vmatchselect` and `vmatch` programs. In addition, the whole software is implemented to speed up computational time and reduce the space consumption. A comparison between suffix trees and virtual suffix trees regarding the running time for finding repeats is shown in table 4.1.

Genome	size (Mb)	search length (bp)	suffix tree (sec)	virtual tree (sec)
E. coli	4.42	150	5.4	1.7
S. cerevisae	11.50	180	14.8	4.7
D. melanogaster	114.44	700	310.7	44.4

Table 4.1: Computational time for finding maximal repeats in seconds (the computation was run on a Sun UltraSparc II 400MHz).

The preprocessing of the *database* sequence is done by `mkvtree`, where the virtual suffix tree of sequence S_1 is constructed. This step generates the corresponding index structure and stores it on files. For the matching task, `vmatch` implements all `REPuter` algorithms on virtual suffix trees. It reads the stored files as a database, matching the query sequence S_2 against the database sequences to find local similarities [16]. The default output of `vmatch` is a list of repeats similar to that of `REPuter`. This time, both database and query sequences have specifications regarding the start positions and lengths of their matches. In addition, the error rate between the matches is indicated, besides their E-value and identity score.

4.2 The New Repeat Graph

As already mentioned in Section 4.1, `GenAlyzer` provides the visualization of `vmatch`, the new version of *REPfind*. The graphical interface is designed to facilitate also the handling of the index construction and the matching task steps. Nevertheless, the application of `mkvtree` and `vmatch` can still be accomplished via the command line. Even though, the main window of `GenAlyzer` (Figure 4.1) offers an easy-to-use interface also for the

processing steps of repeat search, rather than only the repeat graph visualization. This feature increased the acceptance of the software in the biological community.

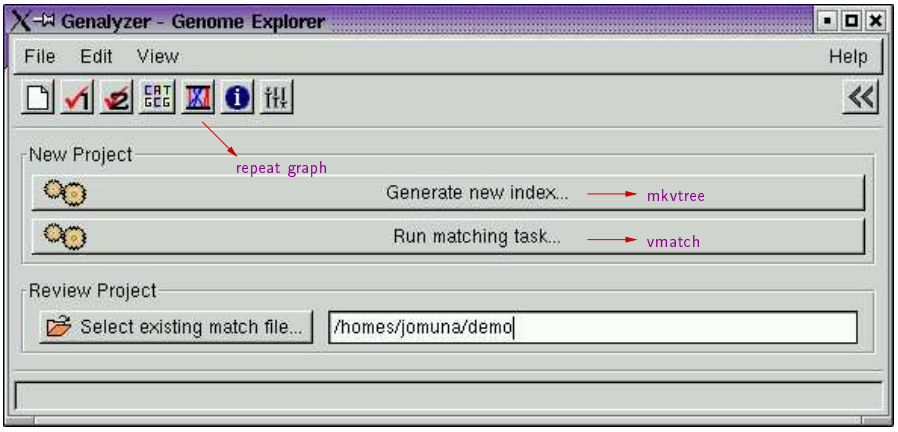

Figure 4.1: Main window of `GenAlyzer`. The arrows indicate the buttons for the respective computational steps.

In Figure 4.1, one can have a clear view of the buttons for the index construction by calling `mkvtree`, the processing of the matching task via `vmatch` and the repeat graph visualization (indicated by red arrows). The first two buttons open further windows, where the user specifies the files to be indexed and matched, respectively. The third button launches the inspector window (Figure 4.2). It illustrates the `vmatch` output, maintaining the basic architecture of the repeat graph as it was shown in *REPvis*.

The main difference consists in plotting the database sequence in the top line of the graph, and the query sequences in the bottom. The line corresponding to the larger sequence is scaled to fit the entire window. Zooming in and out the graph happens in the same window (Figure 4.3).

An additional feature is the *overview graph*. It consists of a duplication of the repeat graph, in a smaller scale, with the advantage that the whole overview of the repetitive structure remains while zooming in or out the actual repeat graph below. The entire overview graph is enclosed by a rectangle with red borders (Figure 4.2). As soon as the user zooms into a specific region in the repeat graph, the red rectangle shrinks, bordering exactly the zoomed region, as it can be observed in Figures 4.3 and 4.4.

According to the zooming scale used, and the distance between the matches, the lines joining both instances of the repeated substrings are restricted to short arrows (Figure 4.3). For this reason, the overview graph facilitates the spatial localization within the matched sequences. Moreover, the user can scroll the red rectangle onto other regions, without loosing the zoomed scale and without having to zoom out first, unlike in `REPuter`. Besides the overview graph, there is also another embedded browser, called the *project info*. It depicts the name of the running project as well as the parameters

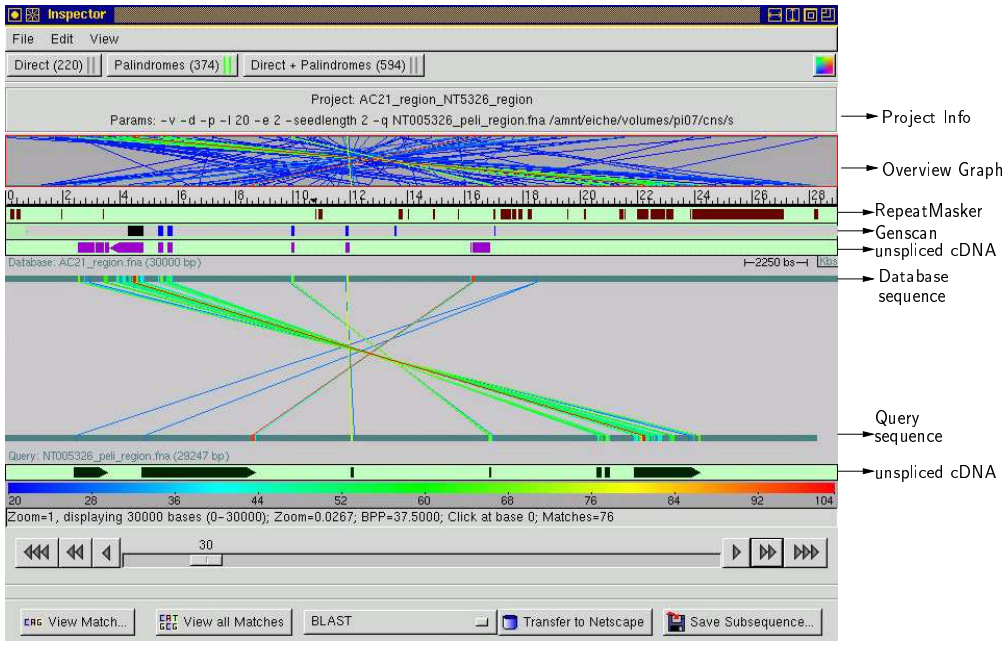

Figure 4.2: The new repeat graph. The top line of the repeat graph represents the *database* sequence. The query sequence is shown in the bottom line. The match graph displays palindromic matches of at least length 30 bp, with an edit distance of 2. The annotation is specific for each sequence. The *overview graph* displays all matches found according to the searching parameters, which can be seen in the *project info* bar.

used for the matching task. If not needed, both the overview and the project info bars can be collapsed, reducing the whole figure size.

The fact that the repeat graph window is not static permits the user to interpret the repetitive structure with more accuracy, because information like the color code is not lost. This was one of the inconveniences in `REPuter`, when opening the extra inspector window for zooming. Another button in the new inspector window refers to the visualization of the alignment of all matches, instead of only a specifically chosen one. To accompany the new repeat graph, as top and bottom lines represent different sequences, they can also be annotated differently. Upper and lower sequence line annotations are specified by the user.

4.3 More Matching Task and Output Options

In this Section, we account for an alternative output option as well as more matching tasks supported by `vmatch` in comparison to `REPuter`. We emphasize on parameters that are important for the further understanding of the utilization of the software in this work. `Vmatch` presents much more application domains than it is described here

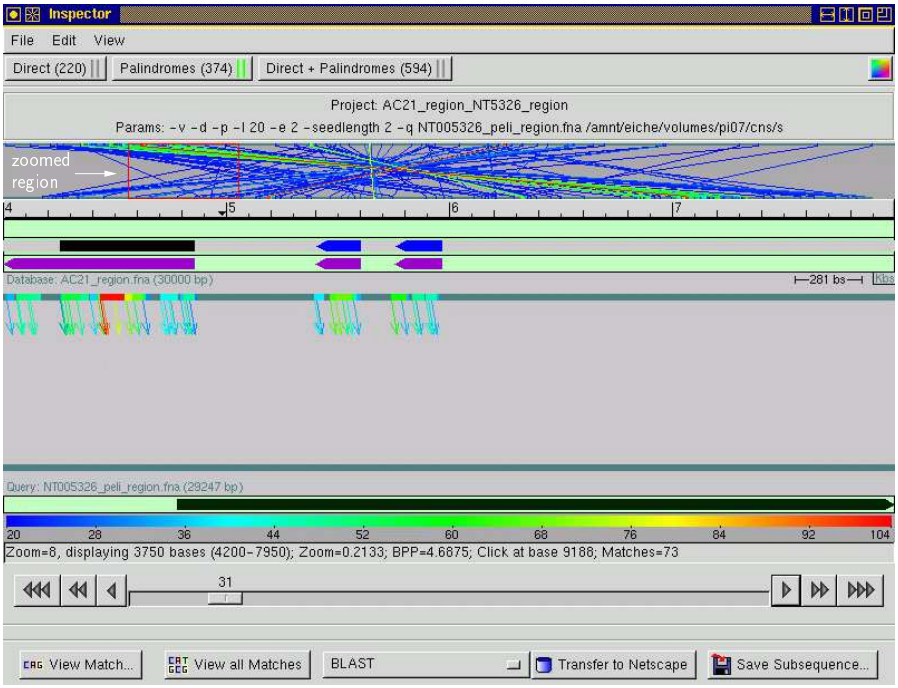

Figure 4.3: Zooming into the repeat graph in Figure 4.2, the matches are restricted to arrows, but the zoomed region can still be identified by observing the overview graph.

(see S. Kurtz [66] for a more detailed overview).

The original purpose of `REPuter` was to identify all repeated substrings in a given sequence under specific parameters. To continue supporting the capability of finding repeats within a single sequence S_1, `vmatch` provides the *self comparison task*. With this option, the sequence S_1 used to create the index is database and query sequence at the same time. The most common application of `REPuter` is covered by this option, giving also `vmatch` the ability to identify low copy repeats or do simple assembly checks. As mentioned before, `vmatch` handles the comparison between two different sequences, unlike `REPuter`. This task refers to the *matching database against query sequence* option. The index is constructed on sequence S_1, which is handled as database sequence. According to the users defined criteria, `vmatch` computes all similarities between S_1 and the query sequence S_2 without calculating the repeats within each sequence. The delivered data are much more restricted to what the user is looking for, avoiding superfluous information and, therefore, reducing time and space consumption. The third kind of matching task is used for motif searching, the *complete match* option. In this case, the entire query, also called *pattern*, must match a substring of the indexed sequences.

For each matching task, the matches can be direct or palindromic. As `REPuter`, `vmatch` is non-heuristic, guaranteeing to find all solutions for exact and degenerate matches. Here, the error parameters include hamming and edit distances, but insertions,

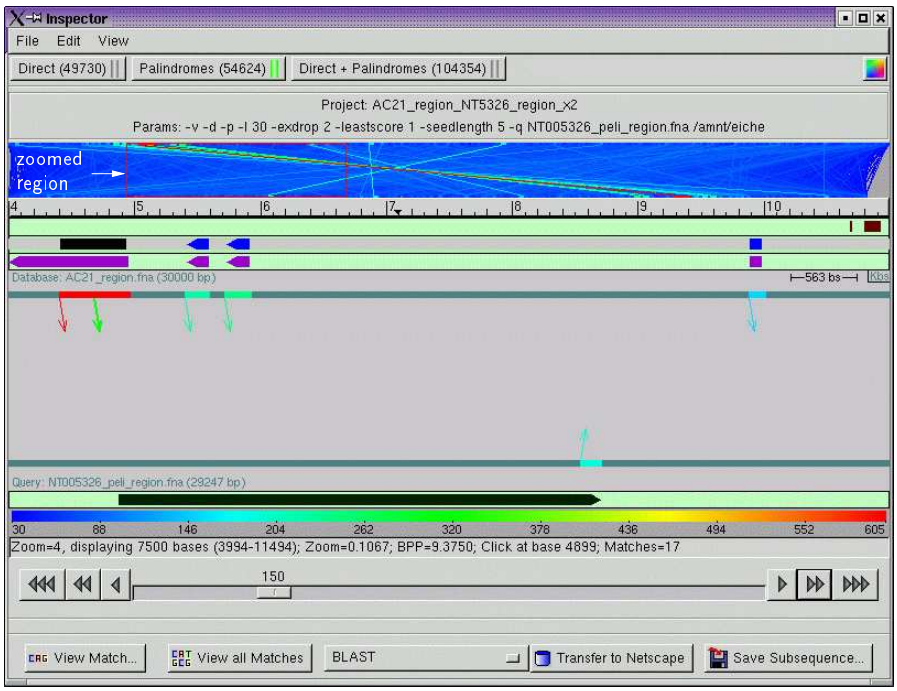

Figure 4.4: Zoomed view of the same region shown previously in Figure 4.3. As it can be observed in the project info browser, the X-drop approach was used for the matching task instead of the edit distance. The short bundle of repeats in Figure 4.3 are merged into larger matches, though containing more errors. The different length span resulted from this approach can also be visualized in the color code bar, which ranges from 30 bp to 605 bp, instead of 20 bp to 104 with the edit distance strategy (as in Figure 4.3).

deletions and mismatches between matches can also be computed applying the X-Drop approach [119]. This parameter represents an alternative strategy for seed extension. It scores the alignments such that each match scores 2, a mismatch gets a score of -1 and an indel -2. The purpose is to find the highest-scoring alignment. The argument X of the option $-exdrop$ is defined by the user and determines how much differences are tolerated in the accumulating extension steps.

This alternative was implemented into `vmatch` to cover a weakness of `REPuter`. Trying to find repeats with very low level of similarities using `REPuter` often led to unrealistic running times. Working within the limits of differences between matches resulted in several short repeats, arranged in larger blocks (Figure 4.3). Besides the fact that those agglomerated repeats sometimes confuses the visualization and thus the interpretation of the repeat structure (see Figure 2.5.3 for an example), some larger segmental duplications with lower similarities would not be identified by `REPuter`. Considering that these regions still have an evolutionary meaning, it is important for biologist to be able to identify and visualize them. The idea would be to merge that bundle of short repeats

into fewer and larger ones, containing more errors. This is exactly what results from the `vmatch` X-drop approach, being another important improvement of `REPuter`. A direct comparison between edit distance and X-drop results can be clearly seen in Figures 4.3 and 4.4.

Another adjustment done in the implementation of `vmatch` concerns the *unique sequence finding problem*. The developers of `REPuter` noticed that biologists were not only using `REPuter` for finding repeats, but also taking advantage of the non-heuristic properties of the software to search for unique sequences (Section 2.5.4). Hybridization experiments in general are accomplished with specific probes, which should hybridize only in the region of interest, avoiding false-positives. This probe specificity infers that it is a repeat-free region, not containing any sequence redundancies. Such techniques involve, for instance, primer design for DNA amplification, probes for Southern blots or FISH, and oligonucleotide design for chip technology [25, 72, 14]. This observation led to an option in `vmatch` which directly finds all substrings of the input database sequence over a specified threshold length that have no match in the query sequence. In a recent study, a group from the Lawrence Livermore National Laboratory profited by this new feature of `vmatch` to develop pathogen DNA diagnostics [37]. By comparing two different bacterial strains, the substrings in the pathogenic strain which do not occur in the other one can be good candidates to explain its pathogenicity.

4.4 Comparison of Gene Structure - Part II

It has been described in Section 2.5.6 how `REPuter` is used to compare gene structures, in spite of the limitations of the software. For analogy purposes, the example used in Figure 2.19 is reconsidered, testing the improvements which overcome those limitations of `REPuter`, in order to adapt it to comparative genome analysis. During the development of `vmatch`, the mouse contig containing the *Peli1* gene increased in sequence content, so that we were able to analyze the genomic region corresponding the entire gene[1]. This sequence is used as database and the human contig NT_005326, which contains the human *Peli1* gene is utilized as query sequence. The result of the `vmatch` output is visualized in Figure 4.5. The top line of the graph displays the mouse sequence and the bottom line the human one. The three annotation lines above the top sequence line represent the unspliced *Peli1* cDNA, the `GENSCAN` prediction and the repeated element positions. Regarding the human contig in the bottom of the graph, only the unsplicing of the *Peli1* cDNA was annotated. For the sake of resemblance, the same search parameters were used, looking for matches of length at least 20 bp and allowing at most 2 errors (as it can be observed in the *project info* browser). The graph in Figure 4.5 depicts all palindromic matches of length 30 bp or larger. This adjustment avoids the disturbance of the light background noise promoted by the relative low computational thresholds used, as it can be seen in the *overview* graph.

[1] GenBank accession number: AC091421

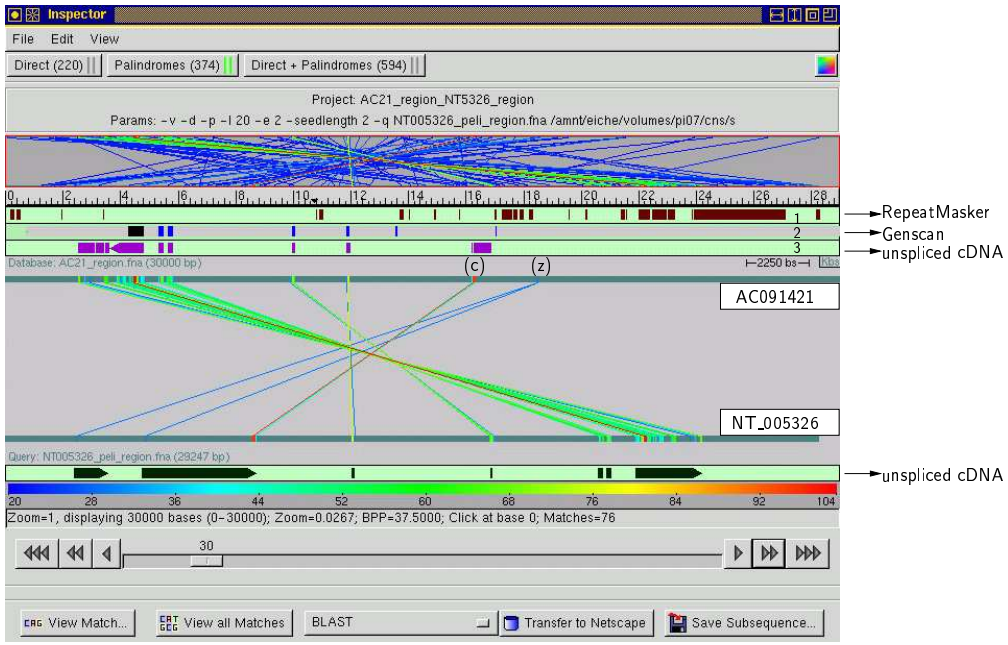

Figure 4.5: Comparison of gene structure - Part II. The genomic sequence of mouse contig AC091421 is represented in the top line or the repeat graph as the *database* sequence. The query sequence used was the homologous genomic sequence in human, in contig NT_005326. Both sequences were searched for conserved sequences of minimum length 20 bp with at most 2 errors. Comparing the GENSCAN predictions and the unspliced cDNA in the top annotation graph with the match (c), it is clear that this exon has been missed by the prediction tool. Besides, the match (z) represents a potential conserved noncoding sequence, as it does not match any known coding exon neither in mouse nor in human.

The main differences between REPuter and GenAlyzer are demonstrated by means of the visualization of the example above. First of all, it is clearly recognizable that the separation of the sequences to be compared in database and query, and, therefore, their display in top and bottom lines is a rewarding improvement for the match graph visualization. The intrasequence matches, i.e., repeats, are not computed in this matching task, and thus not shown in the graph. This leads to a more concise demonstration of the matches found only between the two species (Figure 4.5).

The coding regions of *Peli1* gene show a high level of conservation between mouse and human, observed by many matches localized in the exons region (see unspliced cDNA annotation line). Remembering the discussion on Figure 2.19 about the accuracy of GENSCAN predictions, the visualization of the same region using GenAlyzer points to identical results. It came out that GENSCAN has missed the prediction of an exon, labeled again with the letter (c) in Figure 4.5. Simple observations like these lead us to reaffirm

that comparative genomics is a powerful approach to identify coding regions. This kind of approach in which similarities are found by comparing two different sequences classifies `REPuter`, and, therefore, `vmatch` under the *extrinsic* methods of gene prediction.

The visualization adjustments shown in Figure 4.5 depict another match that attracts the users' attention, labeled with the letter (z). It refers to a conserved region of 30 bp with 2 errors, but still significant, with an E-value of $2.5e^{-5}$. This sequence appears only once in the mouse analyzed region, upstream the *Peli1* gene. It matches twice in human and it is localized in the 5'UTR region of the human *Peli1* gene, according to the sequence information obtained from GenBank. In these data, the 5'UTR of *Peli1* is about 4 kb long in the human, what disagrees with the general estimation of 2,8 kb maximal length for human 5'UTRs [80]. Moreover, it differs a lot from the length of the mouse 5'UTR, of about 640 bp. Either the mouse UTR sequence is incomplete, or the human one still contains vector sequences, for instance, or both. Fact is, that the sequence in question does not overlap with any predicted coding region or a repeated element (which can be clearly observed analyzing the annotation graph in Figure 4.5). This observation indicates that this match attends to a *conserved noncoding region*, thus having potential regulatory functions. Depending on the exact localization within an untranslated region or not, it could be either a DNA or an RNA binding sequence. The topic of conserved noncoding sequences is handled in the next Chapters.

4.5 Summary

The features of `vmatch` which where developed to adapt `REPuter` to the genomic era were shown to serve for a more comfortable employment of the software by biologists. Not only for the easy-to-use interface, the much faster computation and less space requirement, but also for the capability of using `vmatch` for at least the same biological application span as `REPuter`. Avoiding the loss of application fields is an important feature for genome analysis, once `REPuter` already showed to be very suitable for many sequence analysis tasks (Chapter 2). It has been shown that `GenAlyzer`, as the improved version of *REPvis*, is an appropriate interactive visualization for both genomic and post-genomic studies. The ability to compare two distinct sequences without concatenation generates a graphical representation suited for the inspection by humans. Besides, joining the `mkvtree` and `vmatch` applications in `GenAlyzer`'s interface has been shown to be a decisive feature for the software acceptance in the biological community.

5 The Neurologic Mutation Wobbler in Mice

The *wobbler* mutation was first described in 1956 by Falconer [35]. The affected mouse suffers an autosomal recessive mutation producing severe motoneuron degeneration and astrogliosis in the spinal cord [28]. The phenotype can be observed since the 3rd to 4th postnatal week. From then on, a rapid progression of the disease has been described – the majority of the animals die within the first year. The main phenotypic features consist of weakness of the forelimbs, tremor and gait disturbance ("wobbly gait"), followed by muscle atrophy, which is triggered by the motoneuron degeneration. The homozygous (wr/wr) animals are smaller and lighter than the heterozygous (wr/+), clinically normal mice. Histological modifications in neurons of the brain stem, cerebellum and also of the spinal cord have been verified 13 days after birth. In the beginning, only isolated cells presented the alterations, but from the 16th day on, large groups of cells were found vacuolized [91]. However, the neurons did not appear to undergo an apoptosis, leading researchers to eliminate this hypothesis. Besides the death of motoneurons, a proliferation of the astrocytes (astrogliosis) has been detected in the spinal cord [69]. Although some authors consider the astrogliosis a primary event, others argue that it is rather a response to the neurodegeneration [23]. In addition to this neurologic phenotype, the wobbler mouse presents a defect in the spermiogenesis, explaining why male animals are sterile. In this case, the sperms show a distinguished morphology: the mitochondria are displaced, and the formed acrosome is not functional.

Researchers have been spending several years in the investigation of the *wobbler* mutation. However, the histological and biochemical findings alone do not explain the causes of the disease. In the last years, the initial sequencing of the mouse genome brought new hopes to identify the molecular reason that supports the *wobbler* mutation. In this Chapter, we present the main results of these molecular investigations in affected mice (Section 5.1). In Section 5.2, we analyze the genomic region enclosing the mutation at the sequence level using `vmatch` and `GenAlyzer`, delineating the sequencing progression of the contigs containing the main *wobbler* candidate genes. This sequence analysis is followed by Section 5.3, where new hypotheses for the causes of this spinal atrophy disease are exposed, based on evolutionary conservations in noncoding sequences. Finally, this Chapter is summarized in Section 5.4.

5.1 The Wobbler Genomic Region

In the last 10 years, positional cloning has been demonstrated to be a successful and reliable method for the chromosomal localization of genes in one species, based on the

position of homologs in conserved blocks in another species. This method is often used for the identification of disease genes [99], and was also the main procedure to restrict the chromosomal localization of the *wobbler* "gene". In this section, we shortly describe how mouse and human comparisons contributed to the localization of the *wobbler* genomic region (Subsection 5.1.1). Although the conserved synteny facilitates this kind of approach, there are rather short conserved regions between both organisms' genomes than long ones, due to interruptions by insertions or inversions. Once the chromosomal localization has been identified, the genes contained in this region that are candidates for carrying the mutation are presented in Subsection 5.1.2.

5.1.1 Mouse and Human Comparisons

Animal models can be chosen as monitors for pathogenesis and therapy for human genetic diseases based on explicit gene orthology between the two organisms. The challenge is to analyze potential effective treatments resulted from such comparative approach. The laboratory mouse *Mus musculus* is used as main model organism for biomedical research in order to understand many aspects of human diseases. In the following, some reasons of this approach are summarized [23, 116]:

- The fast reproduction of mice and their short life spans facilitate their handling in the laboratory.
- The mouse genetics is very well known, allowing a precise manipulability of every known gene to determine its functions. The ability to manipulate the mouse genome aids in the identification of disease candidate genes.
- Mice present physiological, anatomical and biochemical parallels with humans. Ninety-nine percent of the approximately 30000 mouse genes have direct counterparts in humans.
- The expression analysis of mouse homologs to human genes can be carried out in many different tissues and several different developmental stages.
- Gene-poor regions of both mice and humans are very similar, suggesting that even noncoding regions can be explored to explain similarities and differences between the organisms.

Besides these general features, the availability of the mouse genome offers also access to transgenic and knock-out technologies, targeting human disease genes. These strategies allow the analysis of the gene function *in vivo* by transforming or mutating the gene of interest. In this way, the disease processes in the animal can accurately reflect those in humans.

The *wobbler* mouse is commonly used as model organism for human spinal muscle-atrophy (SMA) and amyotrophic lateral sclerosis (ALS). In case of the SMA, a defect

in the *survival of motoneurons gene* has been detected [71]. ALS is also characterized by motoneuron degeneration. The acquired ALS predominates in the population, but there are also hereditary cases, called *familiarly amyotrophic lateral sclerosis* (FALS). In FALS, the ubiquitous expressed gene for Cu/Zn superoxid dismutase (SOD1) seems to be affected [97]. However, the mechanism of motoneuron damage is still unknown. The minor effectiveness of available treatments and the fatal outcome of these diseases led researchers to use animal models like the *wobbler* mouse as tools for the development of treatments of such diseases [23].

With the sequence availability of the human and mouse genomes, syntenic regions have been identified, containing several homologous genes. Comparative genomics turned out to be a promising approach employed in the detection of candidate disease genes. Human/mouse homology maps have been constructed delineating all syntenic regions between these organisms, facilitating the search and analysis of homologous genes. These maps are accessible through the National Center for Biotechnology and Information (NCBI) web sites (see Appendix B).

5.1.2 Chromosomal Map and Candidate Genes

The experiments on positional cloning and molecular analysis of the *wobbler* genomic region have been carried out during the PhD theses of S. Fuchs and K. Resch [44, 93], at the Institute for Developmental Biology at the Bielefeld University, Germany. The *wobbler* mutation has been mapped on mouse chromosome 11; genes in this region have human orthologs on the syntenic segment 2p13-14 [95]. Candidate genes for the mutation have been identified as a result from the positional cloning, based on the physical mapping of the restricted chromosomal localization of the *wobbler* mutation (see Figure 5.1). In the beginning of this work, the candidate genes have been limited to the following (the GenBank accession numbers are indicated in parenthesis):

- *Pellino1* (*Peli1*): homologous to the *Drosophila melanogaster* Pelle adaptor protein Pellino [94] (AF302505).
- *Hepatocellular carcinoma antigen 8* (*Hcc8*): a tumor antigen, expressed in a variety of tissues (AF102177).
- *Uridil-diphosphoglucosepyrophosphorylase (Ugp2)*: involved in the glucose metabolism, catalyzing the transformation of glucose-1-phosphate and uridil-triphosphate into UDP-glucose [17] (NM_006759).
- *Malate dehydrogenase (Mdh1)*: involved in the citric acid cycle; highly expressed in heart and skeletal muscle (XM_002358).
- *LOC51057*: codes for a hypothetical protein, with unknown function (NM_015910).

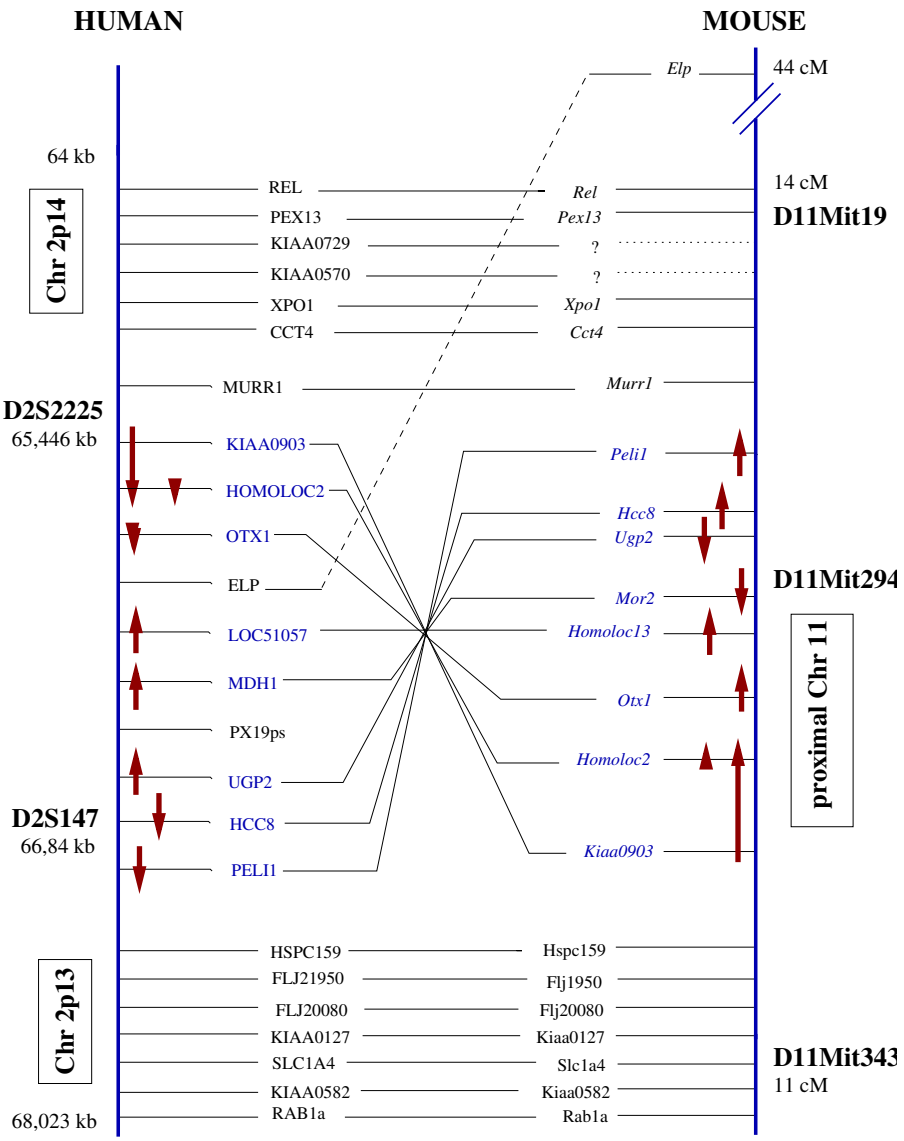

Figure 5.1: Chromosomal map of candidate genes in the *wobbler* region. The map represents the segment on human chromosome 2p13-14 homologous to the mouse chromosome 11. The *wobbler* critical region corresponds to the inverted segment. Arrows indicate the orientation of gene expression. This figure demonstrates the state of research in the beginning of this work. (Taken from [44])

- *Orthodenticle homolog 1* (*Otx1*): codes for a transcription factor, playing an important role in the brain development (XM_049268).
- *Homoloc2*: similar to human EST from Colon carcinoma (Caco-2) cell line II (AA305160).

- Anonymous cDNA *KIAA0903*: resulting from a project for long cDNAs [84, 85], its function is unknown, but presents a CAAX box and weak homologies to a calponin-domain (AX030068).

Analyzing the map in Figure 5.1, we can observe that the gene content in the region of interest is highly conserved, but wet-lab experiments clearly indicated that the gene order is disrupted by an inversion [45]. Although the human pseudogenes *endozepine like peptide* (*ELP*) [114] and *px19*-like protein (*PX19*) [53] were also localized in the chr2p13-14 interval, their corresponding murine orthologs were not detected in the *wobbler* region. Fuchs *et al.* [45] have identified the mouse *Elp* far distally on chromosome 11. The mouse counterparts of the preceding list of human genes are: *Peli1*, *Hcc8*, *Ugp2*, *Mor2*, *Homoloc13*, *Otx1*, *Homoloc2*, and *KIAA0903*, respectively.

5.2 Sequence Analysis of the Wobbler Region

Similar to the Human Genome Project, several institutes and universities constitute the consortium responsible for the Mouse Genome Project (see Chapter 3). The Genome Sequencing Center at the Institute for Molecular Biology in Jena, Germany, started in 1998 a cooperation with the Institute for Developmental Biology in Bielefeld, Germany, in order to generate genomic sequences corresponding to the *wobbler* region in the mouse. In this Section, we present the progress of this sequencing project, beginning with the analysis of the initial sequences, still in the draft stage (Subsection 5.2.1). Considering the joint efforts of the Mouse Genome Project consortium in publishing the mouse genome sequence in December 2002 [116], we also check the finished contigs in the *wobbler* region (Subsection 5.2.2). Making use of the advantages and wide range of applications of the `vmatch` program, it has been chosen as main tool to carry out the sequence investigations. Consequently, we show the outcome of our analysis through the graphical visualization of `GenAlyzer`, facilitating the interpretation of the results.

5.2.1 The Initial Sequencing of the *Wobbler* Genomic Region

In the year 2000, the Genome Sequencing Center in Jena produced 9 sequence segments located in the *wobbler* genomic region. These were named mK2-135B4, mK3-123J24, mK4-139O9, mK6-180K15, mK7-219P9, mK11-48H20, mK12-65I11, mK13-165L14, and mK14-124L2. Wet-lab experiments have demonstrated that the segments mK6-180K15, mK7-219P9, and mK12-65I11 showed overlapping regions, producing one merged segment, called mK5-185K22. In April 2001, these genomic segments were submitted to GenBank. Table 5.1 shows the correspondence of the Jena sequences denominations with the GenBank accession numbers.

Together, the available contigs covering the *wobbler* region account for almost 1 Mb sequence. These sequences are constituted of several smaller, unordered subfragments. The mouse genes identified earlier to be candidates for the *wobbler* mutation

Jena Segments	GB Acession Numbers	Length (bp)
mK2-135B4	AC091422	214440
mK3-123J24	AC091423	104570
mK4-139O9	AC091424	74735
mK5-185K22	AC091428	286564
mK11-48H20	AC091419	70085
mK13-165L14	AC091420	125030
mK14-124L2	AC091421	84497

Table 5.1: Correspondence of the mouse sequenced segments in the *wobbler* region with their GenBank accession numbers. The length of each contig is given in base pairs.

have been localized in the genomic contigs using `vmatch`. All 7 contigs were concatenated and used as database sequence, in the following order: AC091419, AC091420, AC091421, AC091422, AC091423, AC091424, and AC091428. The gene sequences were concatenated as query sequence (*Otx1*, *KIAA0903*, *Ugp2*, *Hcc8*, *Homoloc-2*, *Peli1*, *Mor2*, *Homoloc-13*, respectively), and matched to the corresponding contigs. Figure 5.2 shows the `GenAlyzer`'s visualization of this matching task, where vertical white lines separate the concatenated sequences. Observing the graph, we note that all genes have been recovered in the available genomic sequences, except the *Homoloc-13* gene. This leads us to hypothesize that either the used thresholds are not equally optimal for each gene, or that there are still significant gaps between the contigs that may hold the missing gene. The genes and contigs correspondences are also depicted in Table 5.2. Contigs AC091422 and AC091419 do not contain any known gene so far. However, an interesting observation refers to the gene *Mor2*; in a zoomed view of the match graph (data not shown), it turns out that this gene is contained in both AC091424 and AC091420 contigs. Further analysis have demonstrated that these contigs present overlapping regions that were not merged in the assembly, duplicating the sequence information and therefore, the gene content. This issue is be discussed later.

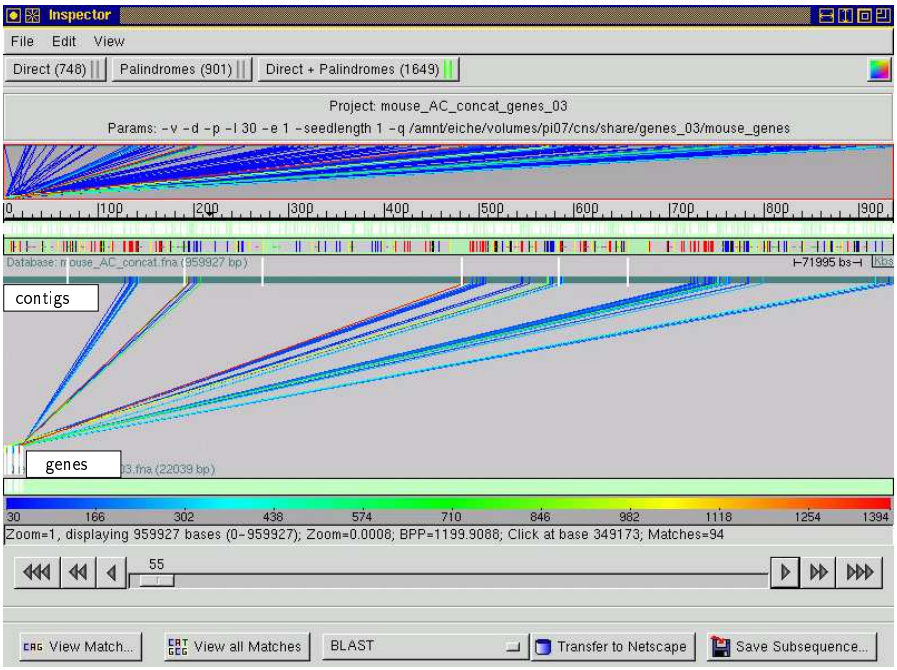

Figure 5.2: Localization of the *wobbler* candidate genes (bottom) in the corresponding mouse contigs (top). The concatenated contigs are represented as database sequence, whereas the genes succession is used as query sequence.

Contigs (GB ac. number)	*wobbler* candidate genes
AC091422	-
AC091423	*Otx1*, *KIAA0903*, *Homoloc-2*
AC091424	*Mor2*
AC091428	*Hcc8*, *Ugp2*
AC091419	-
AC091420	*Mor2*
AC091421	*Peli1*
-	*Homoloc-13*

Table 5.2: Correspondence of the mouse contigs in the *wobbler* region with the candidate genes for the mutation.

Checking the Assembly of Sequenced Contigs

Different from the *whole genome shotgun* strategy, the publicly available mouse genomic sequences had been sequenced using the *hierarchical shotgun sequencing* approach. This second method consists of the generation and organization of large clones covering the genome, like BAC vectors, for instance. Individual BACs are then selected and sequenced by the random shotgun strategy. Afterwards, the sequences are assembled to reconstruct the original sequence of the genome [70]. In order to avoid misassemblies, the small fragments are reorganized by looking for overlapping regions between them.

We already mentioned that 3 of 9 fragments in the *wobbler* region had been merged into one, non-redundant segment (mK5-185K22) beforehand. We demonstrate in Figure 5.3 how the merging of two of those fragments can be recognized using `vmatch` and `GenAlyzer`'s visualization. The database sequence is represented by mK5-185K22, and the query sequence is mK7-219P9. The matching task parameters were set on 50 bp minimal length, searching for exact, direct and palindromic matches. It turned out that the fragment mK7-219P9 was totally contained in the larger mK5-185K22 fragment.

This computational approach demonstrates that the simple, but precise in silico detection of overlapping segments to be assembled is, at least, as reliable as laborious wet-lab experiments, offering a clear visualization of the sequence analysis. In the following, we

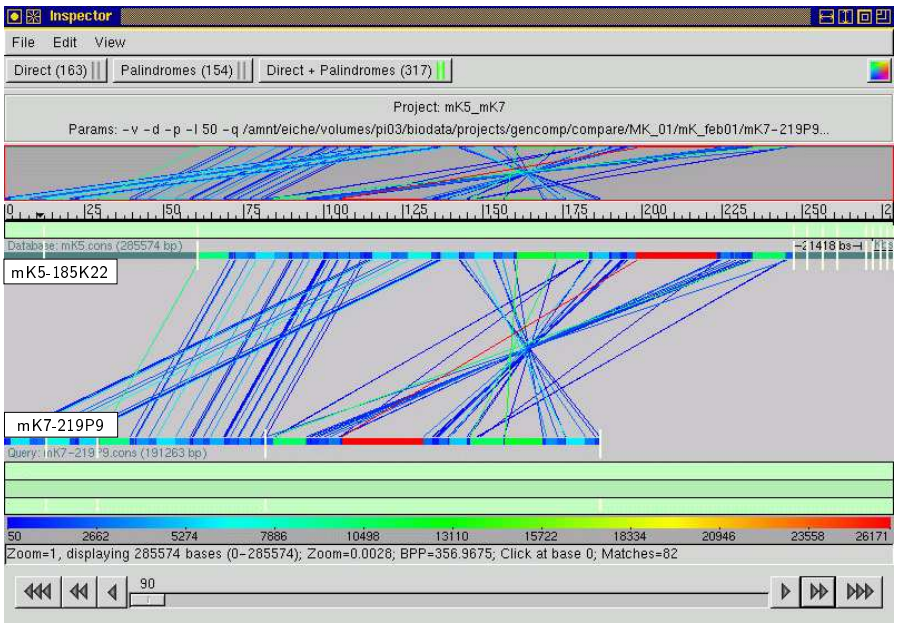

Figure 5.3: Assembly check between mK5-185K22 (top) and mK7-219P9 (bottom) mouse fragments in the *wobbler* region. The graph shows exact direct and palindromic matches of 90 bp minimal length. The whole query sequence is recovered in the database sequence.

present two more examples of such misassemblies, referring to the published sequence fragments in the *wobbler* region, listed previously in table 5.1. With regard to mK5-185K22, it did not only overlap with the mK7-219P9 fragment. In this case, we have found an additional sequence redundancy with the segment mK11-48H20. This time, the corresponding published contigs AC091428 and AC091419 were used as database and query sequences, respectively. It can be observed in Figure 5.4, that the query sequence also consisted of a subsequence of the segment AC091428. The same computation parameters were used as in the previous matching task. These contigs are still incomplete, but the different pieces are separated by Ns, so that the delimiters were not considered anymore in the annotation graph. However, analyzing the positions of the overlap, we can conclude that we are dealing with a double-overlap, between mK7-219P9, mK11-48H20, and mK5-185K22. Although these results have been drawn after the submission of the mouse contigs to the database, we advert that the redundant contig AC091419 should not be considered in the final mouse genome assembly.

A slightly different example of misassembly in the *wobbler* region refers to the contigs AC091424 and AC091420. In contrast to a total sequence recover as shown in the two previous figures, Figure 5.5 demonstrates a partial overlap between AC091424 and AC091420. The `GenAlyzer`'s visualization in Figure 5.5 depicts all exact direct matches of length at least 70 bp. Note that only the end of contig AC091420, used as query

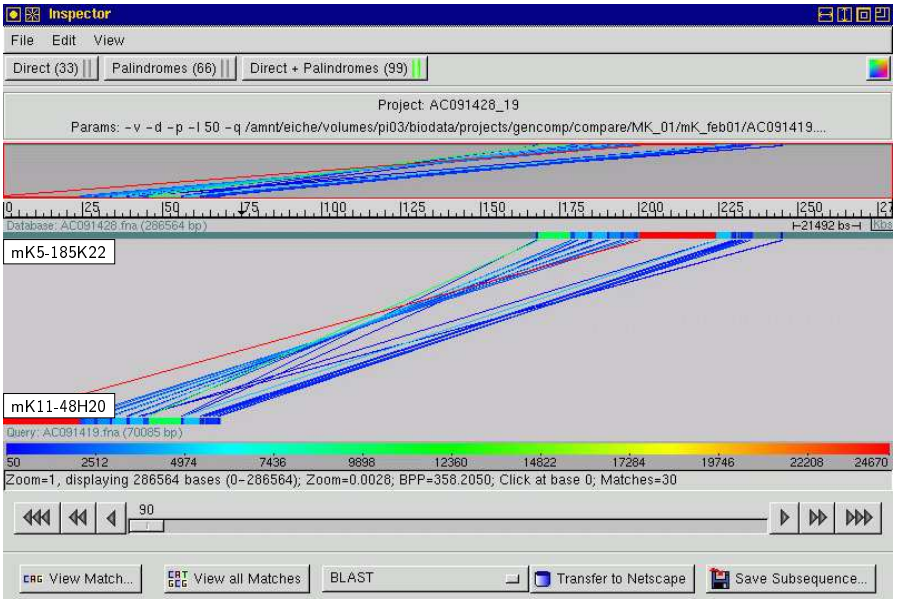

Figure 5.4: Assembly check between mK5-185K22 (top) and mK11-48H20 (bottom) (AC091428 and AC091419) mouse fragments in the *wobbler* region. The graph shows exact, direct and palindromic matches of 90 bp minimal length. The whole query sequence is recovered in the database sequence.

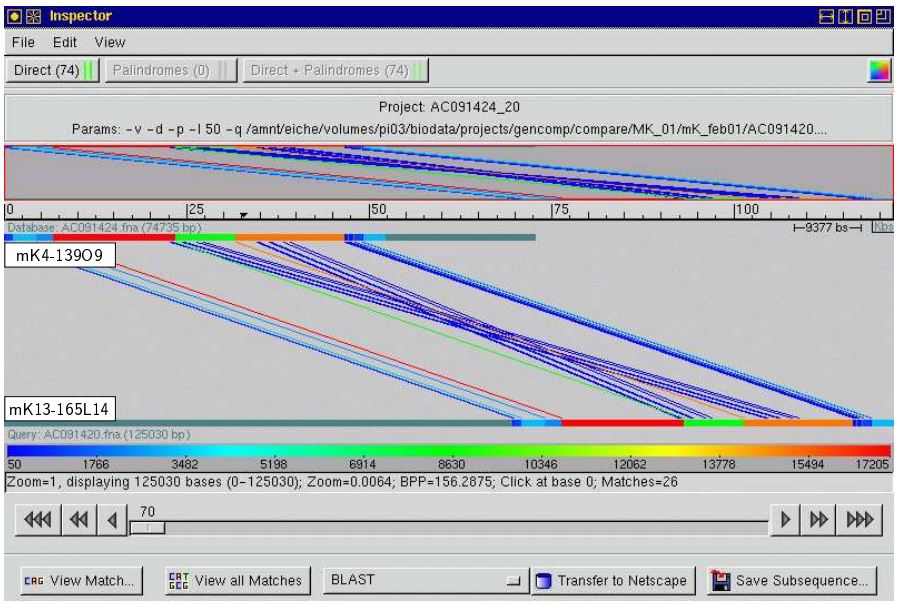

Figure 5.5: Assembly check between mK4-139O9 (top) and mK13-165L14 (bottom) (AC091424 and AC091420) mouse fragments in the *wobbler* region. The graph shows exact direct matches of 70 bp minimal length. A clear inverted segment in the matched region can be recognized. The end of the query sequence overlaps with the beginning of the database sequence. After determining which orientation of the observed inversion is the correct one, both contigs can be merged into one non-redundant fragment.

sequence in the matching task, overlaps with the beginning of the database sequence. The end of contig AC091424 remains without any matches. These findings show that both contigs can definitely be merged into one larger fragment, avoiding superfluous sequence information. Nevertheless, a part of the matched segment indicates a clear inversion between both contigs. Before the assembly, further experiments have to define which orientation of the sequence in question is the correct one.

Another kind of assembly analysis using `vmatch` and `GenAlyzer`'s visualization is the ordering of subfragments within a clone fragment. We take as example contig AC091428 that is constituted of 11 pieces in the following arbitrary order: mK5-7219, mK5-1535, mK5-3431, mK5-0221, mK5-0269, mK5-0299, mK5-0342, mK5-0105, mK5-0140, mK5.0154, and mK5-0202. We have already seen in Figure 5.2 and in Table 5.2, that the contig in question contains the genes *Hcc8* and *Ugp2*. Our approach was based on the application 'cDNA matching onto genomic sequences', described earlier in Chapter 2. By concatenating the cDNA sequences of *Hcc8* and *Ugp2*, we have generated the query sequence to be matched to the contig AC091428. Figure 5.6 shows the result of this matching task, where the concatenated sequences are separated by vertical white lines. Observing the graph, we note that matches were found spread all over the contig,

directly or reverse-complemented. Considering that genes are units with a determined succession of DNA bases, we reorganized 6 of the 11 pieces of the contig in question, resulting in a sequential order of the matches. The obtained match graph is visualized in Figure 5.7. The new organization did not allow crosses between the matches, and some of the genomic pieces had to be reverse-complemented to get a reasonable sequence unit. This approach split the contig AC091428 into two parts, one containing 6 ordered pieces (mK5.0342, mK5.0154, mK5.0140, mK5.0202, mK5.0299, mK5.3431, respectively), while the rest remained unordered. Even though these several pieces may have gaps inbetween, we were able to arrange over 50% of the fragments into the correct order.

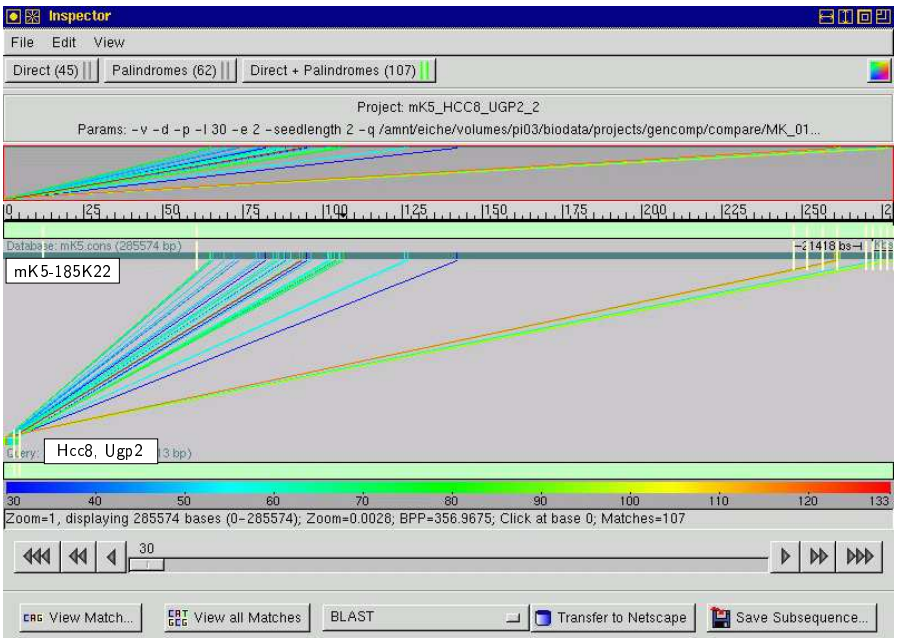

Figure 5.6: Contig organization in the mouse *wobbler* region. The database sequence is represented by the AC091428 (mK5-185K22) mouse contig, where vertical white lines indicate the borders of each subfragment. The concatenation of the cDNAs of *Hcc8* and *Ugp2* has been used as query sequence. The `GenAlyzer` visualization shows the distribution of the *Hcc8* and *Ugp2* exons spread over several contig pieces. This is the unordered view of the contig.

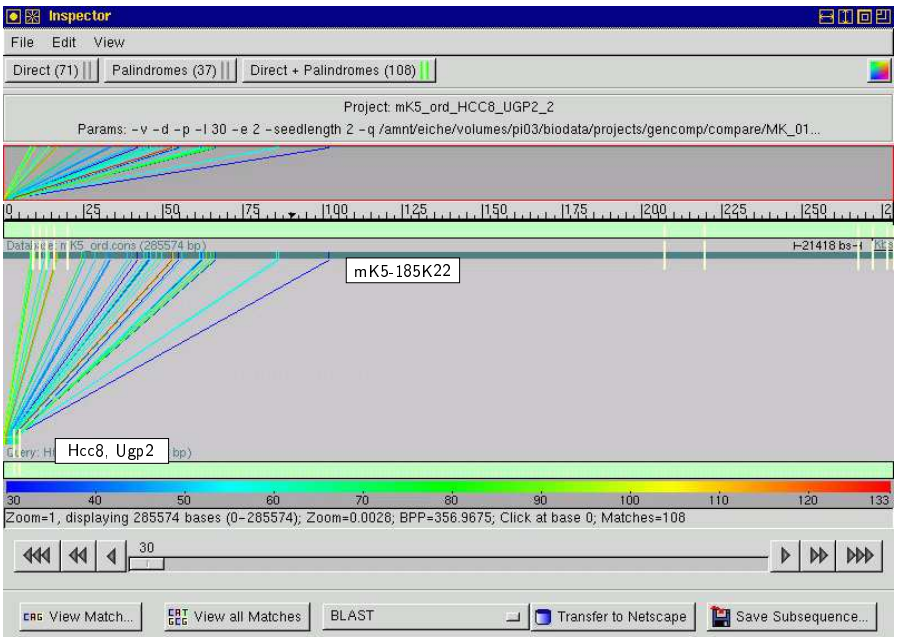

Figure 5.7: Contig organization in the mouse *wobbler* region (ctd.). The graphical visualization shows the distribution of the *Hcc8* and *Ugp2* exons over 6 ordered contig subfragments of AC091428. They have been organized to get a successive cDNA sequence, taking in consideration the orientation of gene expression. The 5 genomic pieces to the right are still unordered.

The Human Genomic Counterpart

By the time the mouse contigs were submitted to the public database, the Human Genome Project was close to its completion. The 7 mouse contigs accounting for almost 1 Mb genomic sequence show homologous counterparts in only 2 human contigs, covering about 1.74 Mb of the *Homo sapiens* genome. These contigs are denominated NT_005056 and NT_005326, localized in the human chromosome 2p13. Supported by the versatility of `vmatch`, we can establish the exact correspondence of 6 of the 7 mouse contigs to their correct counterparts in both human contigs (Figure 5.8) (the mouse contig AC091422 was not considered in this computation, because wet-lab experiments demonstrated that it presents chimeric features, loosing its significance for further analysis). NT_005056 and NT_005326 have been concatenated and used as database sequence, while the query sequence comprises the succession of the mouse contigs, in the following order: AC091423, AC091424, AC091428, AC091419, AC091420, AC091421. Considering that this matching task refers to comparisons of genomic sequences of two distinct species, we searched for matches of minimal length 20 bp, but without any error allowance, as the goal was to roughly identify the contigs correspondences.

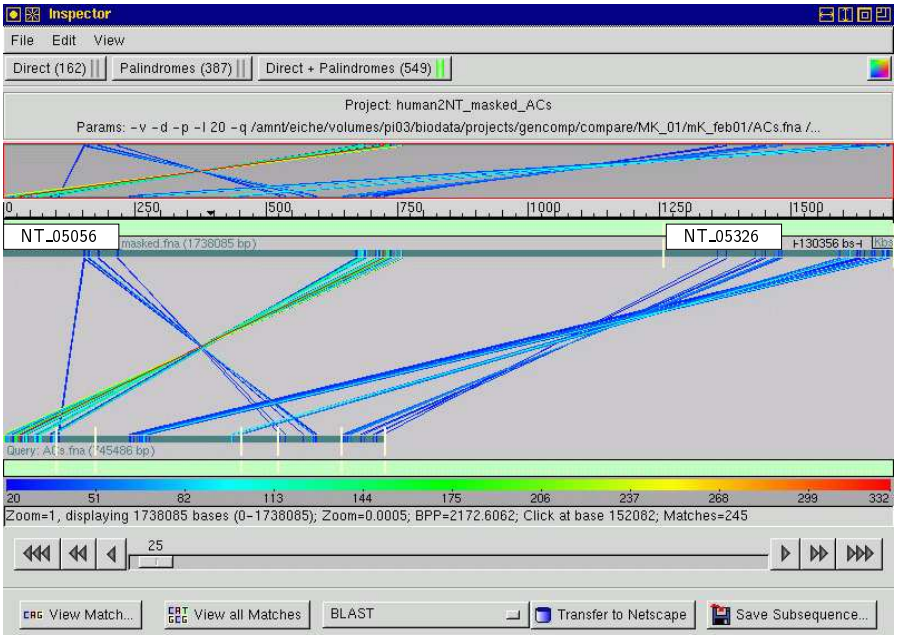

Figure 5.8: Human (top) and mouse (bottom) contig correspondence in the *wobbler* region. Human contigs NT_005056 and NT_005326 are concatenated and used as database sequence. The query sequence comprises the succession of the mouse contigs. The task was carried out searching for matches of minimal length 20 bp.

This approach allowed us to establish the human counterparts to the mouse contigs as follows: NT_005056 corresponds to the region of contigs AC091423, AC091424 and AC091420, while mouse AC091421 and AC091428 contigs match into the human genomic region denominated NT_005326. Note that the mouse contig AC091419 did not show any matches in the human genomic sequences. As it has been observed earlier in Figure 5.4, AC091419 was totally enclosed in contig AC091428, exactly in the interval where the latter one also did not present matches with the human contigs. This indicates that this region may have less similarities between mouse and human, possibly due to the absence of coding sequences in this segment.

Gene Structure Comparison of Some Wobbler Candidate Genes

Once the chromosomal localization of the *wobbler* mutation has been determined, all genes contained in the restricted region become candidate genes for carrying the specified mutation. During her PhD thesis, S. Fuchs [44] carried out several wet-lab experiments searching for differences between the *wobbler* and the wild-type mice. The establishment of full-length cDNAs for all candidate genes and a comparative sequencing procedure aimed to detect the mutation, but no mutation and no difference of transcript levels have been found so far. The approach of comparing the gene structures utilizing

`vmatch` has already been described in detail in Chapters 2 and 4. The example supporting the description was based on the mouse and human *Peli1* gene comparisons. The corresponding visualization of the results have already been shown in Figure 4.5.

Although the exon-intron structures of all candidate genes have been extensively studied by Fuchs *et al.* [45], we present the `GenAlyzer`'s visualization of the genomic comparisons between mouse and human, concentrating only on the *KIAA0903* and *Otx1* genes. Both genes are localized in the human contig NT_005056, homologous to the mouse contig AC091423. *Otx1* is a 2138 bp long cDNA, but it does not extend in the genomic sequence as much as *KIAA0903* does. The *KIAA0903* cDNA is 4911 bp long, coding for a large protein. Figure 5.9 shows the visualization of the matching task between the mouse and human contigs, computed by allowing 40 bp least length and at most 3 errors.

The annotation graph in Figure 5.9 represents the localization of repeated elements

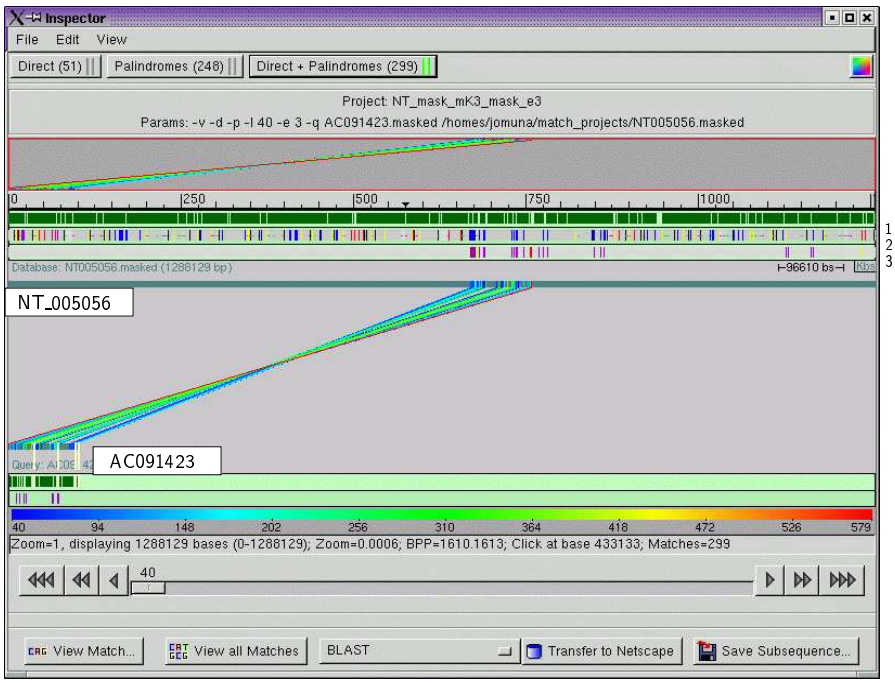

Figure 5.9: Mouse (bottom) and human (top) comparison of *KIAA0903* and *Otx1* genomic regions. The visualization of `GenAlyzer` shows the matches of at least 40 bp with 3 errors. The 3 annotation lines from the database sequences correspond to the masked repeated elements (1), `GENSCAN` prediction (2) and the unspliced regions of the cDNAs (3). *KIAA0903* and *Otx1* are also represented in the mouse, extending their sequences along the whole contig AC091423. As can be clearly seen, the mouse contig is not large enough to cover the entire *KIAA0903* gene. This gene covers a region of about 590 kb on the human genomic sequence.

in the first line (denoted by 1), and the GENSCAN gene predictions in the second line (denoted by 2). The purple boxes in the third line (denoted by 3) correspond to the unspliced cDNA matches of *KIAA0903* and *Otx1* onto the genomic sequence. It is clear that *KIAA0903* is not only a large cDNA, but its exons are found spread out in the human genomic sequence covering a region of almost 590 kb. Unfortunately, the mouse contig AC091423 was only about 104 kb long, covering the entire *Otx1* gene, but only the last exons from the human *KIAA0903*. In order to analyze the matched region in more detail, we zoomed in the Figure 5.9, focusing on the conserved segments between both genomic sequences (see Figure 5.10).

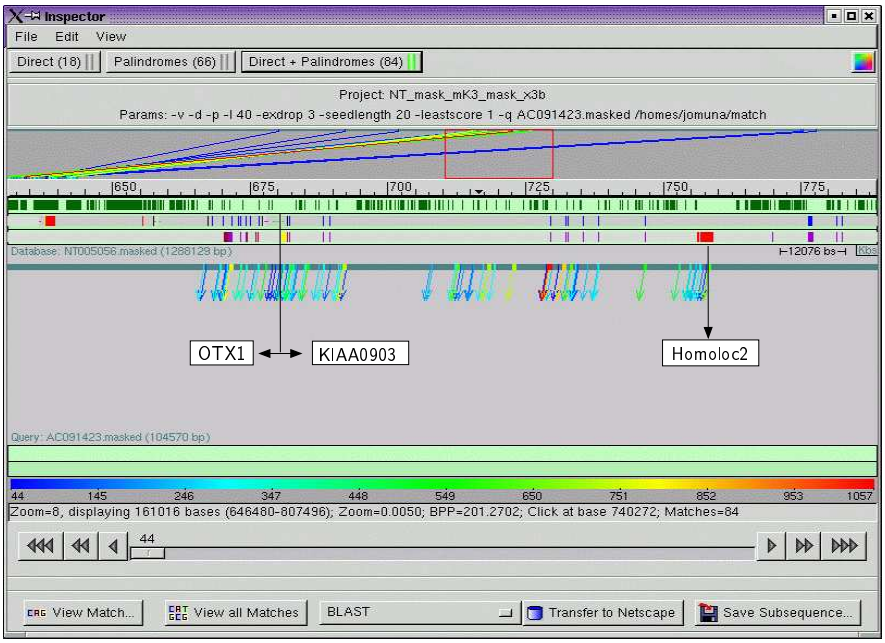

Figure 5.10: Mouse (bottom) and human (top) comparison of *KIAA0903* and *Otx1* genomic regions (ctd.). Enlarged view of the matched region, now computed using exdrop 3 parameter. All *Otx1* exons are conserved between the species, as well as the last *KIAA0903* exons. Besides the coding sequences, several matches can be seen inbetween the exons. The large red box in the third annotation line corresponds to an entry in an EST database, called *Homoloc2*.

We observe that all *Otx1* exons are represented by a similarity match, as well as the 3' end of *KIAA0903*. However, many matches can also be observed inbetween the known coding regions of *KIAA0903* and *Otx1*. This can be the first hint that leads us to evolutionary questions concerning regulatory functions in noncoding sequences. Sequences that are responsible for the regulation of gene expression, transcriptional control or even translation are expected to be actively conserved through evolution. Mutations or any kind of changes in these regions may lead to gene disfunction, generating diseases or even

be lethal. In 1997, Hardison *et al.* [55] wrote about the reason to sequence the mouse genome, under the observation that human-mouse sequence alignments contribute to discover and describe regulatory elements. Recently, researchers have discussed the distinction of actively conserved elements from elements that result from shared ancestry [41] with insufficient divergence time. The graph in Figure 5.10 is a suitable example for possible functionalities in noncoding sequences. Within a large intron of *KIAA0903*, a highly conserved region between mouse and human was detected. This segment is represented in the third annotation line by a large red box. Over 500 bp, with a percent identity of over 95%, it raised the hypothesis that the *KIAA0903* introns may be hiding unknown genes. This presumption was supported by EST database searches, since it was first identified as a human cDNA clone, *Homoloc2*. However, experimental results carried out by Fuchs *et al.* [45] demonstrated that this sequence neither has an open reading frame, nor a detectable transcript. These authors suggested a contamination of the EST database. Thus, the high conservation, together with other matches in intronic regions, suggests a regulatory function. However, the possibility that eighty million years between mouse and human divergence may not have been enough time to become free of evolutionary leftovers should not be ignored.

Our results led us to reconsider the initial idea to compare mouse and human genomic sequences for the *wobbler* mutation detection. The identification of conserved sequences with potential functional importance has been transfered to the regions that do not code for proteins. In Section 5.3, we introduce the issue about conservations in noncoding regions, which may have important biological functions. We give an overview about regulatory elements that influence gene expression. As all candidate genes for the *wobbler* mutation have been excluded by wet-lab experiments, maybe a conserved noncoding sequence with *cis*-acting effect on genes that might even be localized outside the *wobbler* critical region might be affected.

5.2.2 Analysis of the Finished Sequences

Since the publication of the human genomic sequence in February 2001 [70], progress has been made to finish the sequence assembly of the remaining draft contigs. The corresponding *wobbler* region in the human genome had been localized before in the contigs NT_005326 and NT_005056. Both contigs are now localized in the larger genomic sequence NT_005375, 15816879 bp long. Figure 5.11 shows the visualization of the match task, showing this new assembly, where the contig NT_005375 has been used as database sequence, and the draft contigs NT_005326 and NT_005056 as query sequences. The *wobbler* candidate genes had been matched to the new contig in a previous match task (data not shown), allowing to annotate their localization in the graph. This is represented by colored bars in the annotation graph of the upper sequence. Not surprisingly, it is the same region that matched to the draft contigs, concluding that the assembly of this region had not been subjected to large modifications, except the addition of new sequences.

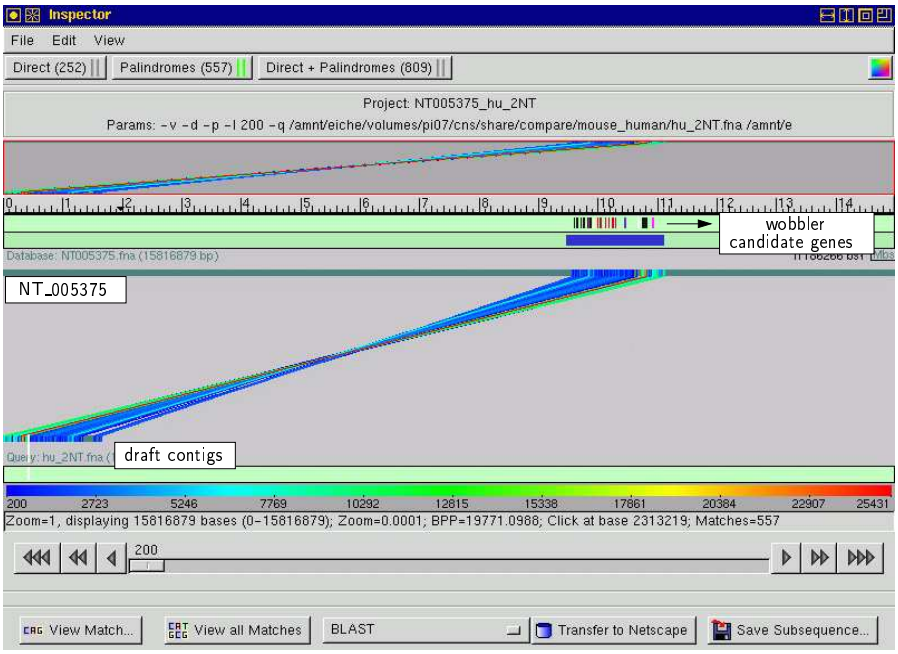

Figure 5.11: Localization of both human draft contigs (bottom) in NT_005375 (top). The draft sequences of the contigs NT_005326 and NT_005056 enclosing the *wobbler* genomic region were concatenated and used as query sequence. They were localized in the larger contig NT_005375, after the new release of the human genome assembly, in February 2003.

Similarly to the human draft contig sequences, we were able localize the mouse drafts in a larger contig, assembled in the release of February 2003. The same approach was carried out, by using as query sequence for the match task the concatenated mouse contigs (AC091419, AC091420, AC091421, AC091422, AC091423, AC091424, and AC091428, respectively). The new genomic contig where the drafts are localized is called NT_039515. The results of this match task can be seen in Figure 5.12. This time, a larger reorganization of the draft sequences has been carried out, so that the final assembled sequence did not contain the overlapping regions of the draft contigs we have discussed previously in this Chapter.

Finally, the general visualization of the finished mouse and human genomic sequences enclosing the *wobbler* region is depicted in Figure 5.13. The annotation graph shows the localization of the *wobbler* critical region (large blue box) as well as the candidate genes present in the segment (colored bars). In this match graph, all matches of least length 150 bp with at most 2 errors are represented. An interesting observation refers to the occurrence of segmental inversions between both genomic sequences. Three distinct blocks, labeled A, B and C, are found to be homologous in the mouse and the human sequences, but in a reversed orientation to each other. These results confirm the experi-

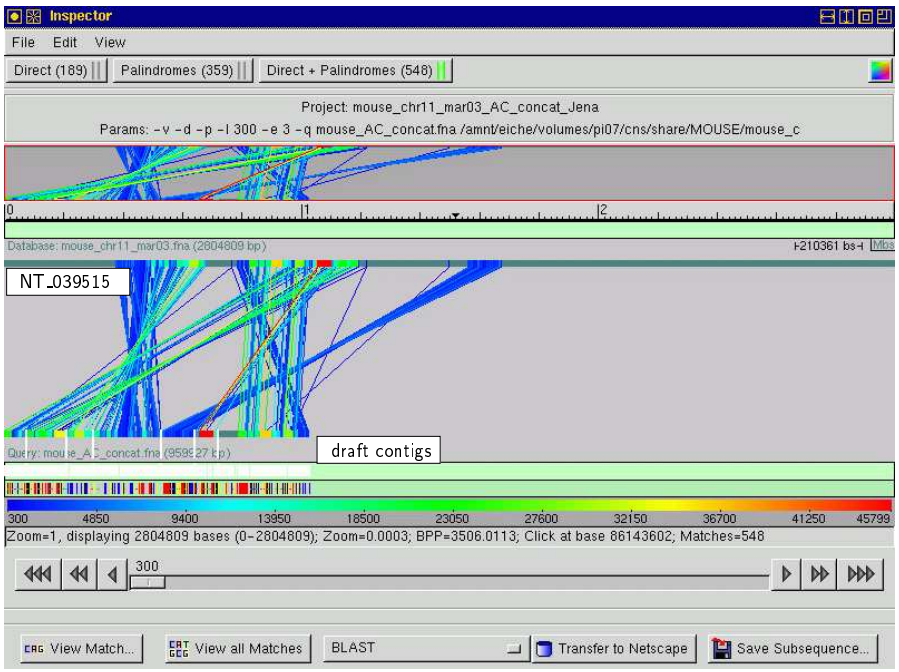

Figure 5.12: Correspondence of the new assembled mouse genomic sequence, NT_039515 (top) with the concatenation of the draft contigs (bottom) in the *wobbler* region. The overlapping regions between the draft contigs have been removed in the final assembly.

mental findings by Fuchs *et al.* [45], when concentrating on the *wobbler* genomic region as a whole. Figure 5.14 shows the order and orientation of the *wobbler* candidate genes in the human and the mouse genomic sequences as they were found in the match task results. Although all genes represent an inverted block, the ones colored red appear in a different orientation as shown before by Fuchs [44].

As `vmatch` permits only a sequence-based analysis, the correct orientation of the genes can only be determined by experimental work. Moreover, the sequences between blocks A, B and C that did not show significant matches should also be analyzed in the wet-lab. The results would indicate whether these blocks represent one large inverted segment or not, consistent with previous analysis in Figure 5.1

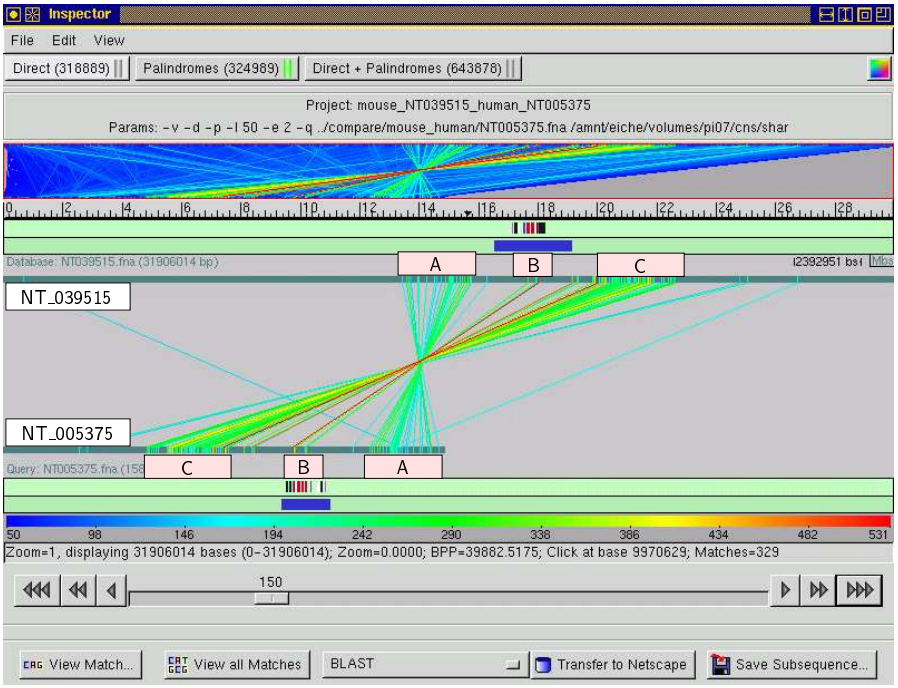

Figure 5.13: Correspondence of the new assembled mouse (top) and human (bottom) genomic sequences. The mouse contig NT_039515 presents a large homologous region with the human contig NT_005375. This segment includes the *wobbler* critical region (blue box in the annotation graph) and clearly confirms the experimentally observed inversion (block B).

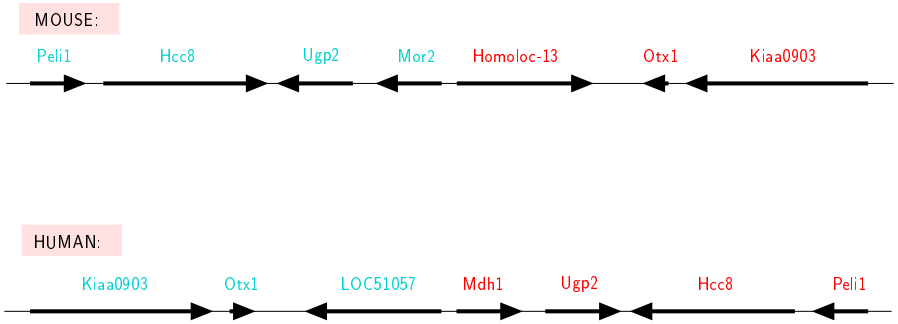

Figure 5.14: Graphical representation of the *wobbler* candidate genes in the finished mouse and human genomic sequences. Their order and orientation have been determined via match tasks of the corresponding cDNA sequences and the contigs NT_039515 (mouse) and NT_005375 (human). The orientation of the genes drawn in red is not consistent with previous experimental results [45].

5.3 Conservation in Noncoding Regions

Eukaryotes present a complex network of gene expression. Sequencing the whole genome of organisms contributed to the recognition that the regulatory context is a crucial part of gene function. As these regulatory sequences lie in noncoding regions of the genome, they are missed by the sequencing of only coding sequences. Such elements are under a positive selective pressure during evolution, due to their functionality as controllers. Consequently, when performing interspecific sequence comparisons excluding the highly conserved coding sequences, the analysis of noncoding regions also reveals conserved features [74, 42]. Many of these are likely to be *cis*-regulatory elements, the main elements in the transcription-control regions responsible for the regulation of eukaryotic gene expression. They are usually cell-specific, or even specific to the stage of development [13, 36]. The closeness of these elements relative to the transcription start site determines whether they are called *promoter-proximal elements*, or *enhancers*, which can be located far away from the start site.

In general, the term promoter refers to the sequences that determine the initiation site of a gene, like TATA-box or initiators, but sometimes the surrounding transcription-factor binding sites are also included in the *core promoter* [111]. With regard to the spatial localization of such control elements, promoters have shown to be considerably flexible, but separations of several tens of base pairs may decrease transcription. However, enhancers can stimulate transcription thousands of base pairs away from the start site. Moreover, they can be located up- or downstream a promoter, within an intron, or even downstream from the final exon of a gene. This wide spectrum of possible localizations in addition to the long-distance transcriptional control and the absence of a sequence consensus in these elements are the main reasons that explain the difficulties to predict and detect enhancers in silico. Today, the identification of enhancers is still done in the wet-lab. Recently, Muller *et al.* [82] discussed advantages and disadvantages of using the pufferfish and zebrafish for the detection and functional analysis of conserved *cis*-regulatory elements. Revolutionary breakthroughs in high-throughput gene expression monitoring technology and cheaper and faster transgenic bioassays are being developed in parallel to more bioinformatics tools. In the future, available sequence analysis software will be joined with more accurate and reliable promoter prediction tools, providing the background for large-scale regulatory elements detection spanning whole genomes.

Knowing that regulatory elements are sequences that do not code for proteins contributes to the restriction of the search space even in large-scale genomic approaches. A pre-selection of conserved noncoding sequences is the first step of such an expression analysis. In Chapter 6, we describe our approach to provide the necessary information to achieve the first level of regulatory regions detection based on sequence comparisons. We developed an integrated system of available bioinformatics tools, facilitating the analysis of conserved sequences between different species. Nevertheless, in the future, the computer scientists community will still face many difficulties in searching for regulatory

sequences, since sometimes the structure and sequence of *cis*-regulatory elements may change during evolution, even when the expression pattern is conserved. This implies that, in the end, the functional confirmation of identified conserved noncoding sequences requires the analysis in the context of the whole organism. The final assurance of the *cis*-regulatory role of such sequences will always be established by experimental work, after the in silico pre-selection has significantly reduced the number of candidate control sequences.

5.4 Summary

Human diseases as the SMA and ALS have been studied for years in order to identify the mutated genes involved in the phenotype, as well as the mutation itself. Model organisms like the *wobbler* mouse have been utilized for molecular and biomedical experiments, due to its easy handling and known genetics. The *wobbler* mutation in the mouse has been restricted by positional cloning to a 1 Mb region on chromosome 11, which is homologous to the human chromosome 2p13. Almost all genes within this mouse region have counterparts in human. We have demonstrated that the `vmatch` program is an appropriate tool also for the sequence analysis of genomic regions involved in diseases. With `GenAlyzer`'s match graph visualization, the genes have been located and compared to the human homologous sequences. Experimental work comparing the genes of wild-type and affected mice did not show any differences at the nucleotide sequence level, suggesting that the mutation may lie within some regulatory element. The utilization of comparative genomics is an effective approach to identify potential control regions, based on their conservation along evolution. The versatility of `vmatch` allowed us to vary the parameters searching for conservations in noncoding regions between mouse and human. In general, sequence-specific conservation of noncoding DNA implies functional constraints on these sequences and slower rates of molecular evolution. Our human and mouse genomic sequence comparisons revealed more conserved blocks in noncoding regions than it has been expected before, suggesting that eighty million years was not enough to become free of evolutionary leftovers. In the near future, time consuming and expensive analysis experiments will be substituted by sophisticated and reliable computational tools, as well as more appropriate laboratory approaches.

6 Identification and Analysis of Conserved Noncoding Sequences

Comparative genomics, the new approach that compares the genomic sequences of different organisms, has been shown to facilitate the understanding of gene expression through functional analysis of conserved regions that do not code for proteins [87, 3, 58, 90]. There can be many different functions within these conserved noncoding sequences (CNSs), such as transcription factor binding sites, enhancers, silencers, matrix attachment regions, etc. They can be better recognized under an evolutionary point of view, as vital information tends to be spared of changes during species development.

In the last years, high-throughput genomic sequencing induced the continuous expansion of sequence databases. There is still an increasing amount of raw genomic sequence data being generated today. This ongoing large-scale sequencing must be accompanied by improvements also in annotation and analysis techniques. New advances in software tools for post-sequencing functional analysis are being demanded and delivered. These programs are required to be flexible enough in order to be adapted to the information evolution due to the continuous flow of new sequence data. It was shown in Chapter 3, for instance, how `vmatch` successfully escorts such development into the comparative genomics era.

Our goal is to develop a system which provides bioinformatics support for generating and screening a set of *potential conserved noncoding sequences* (pCNSs) between genomic sequences of two distinct species. The exact number and localization of all genes (i.e., coding sequences) in the genomes that are being sequenced is still unknown. Consequently, the constant update of draft sequences, or the improvement of gene prediction programs, for instance, may identify genes that are unknown today. This is the reason for the terminology *potential* conserved noncoding sequences as output of our developed system. The automatic cascade of bioinformatics tools works in 3 steps: it first annotates the input sequences, then it computes conserved elements between two selected input sequences. Finally, the list of pCNSs is generated by filtering conserved elements that are known to code for proteins or to represent repeated elements. The computational cascade is called *Conserved Noncoding Sequences Repository Generator*, or *Connosseur* for short. The resulting Repository of pCNSs is based on a RDBMS, to supply the data management. This generated Repository of pCNSs contains all intermediate and final information concerning the annotation and comparison of entered genomic sequences. Depending on the background level of similarity between the organisms being compared, the amount of pCNSs can be very large and almost impracticable to handle in the wet-lab. In the Repository generated by the system, the user can

search for pCNSs that satisfy specific criteria (for instance, their position relative to coding sequences). The underlying DBMS allows to select those subsets in a convenient and efficient way. Moreover, the system is built up in a very flexible and modular way, allowing the addition of other tools for analysis improvements. With more sophisticated and accurate prediction programs, further restrictions on the pCNSs can be imposed in the future. *Connosseur* is the common step of subsequent functional analyses of pCNSs that depend on individual interests and investigation purposes.

This Chapter describes how the comparison of two sequences is automated in order to identify pCNSs with potential regulatory functions. The underlying database system used for the storage of all computed information is introduced in Section 6.1. In Section 6.2, the three steps of the *Connosseur* design are presented. For the understanding of the tools cascade, the underlying bioinformatics programs are shortly introduced in Section 6.3. The pipeline for the pCNSs identification and the implementation of *Connosseur* is detailed in Section 6.4 based on graphical representations. To illustrate each step of the cascade, Section 6.5 describes an example application of *Connosseur*. The mouse and human genomic sequences corresponding to the *wobbler* region are analyzed for pCNSs in Section 6.6. The performance of the system is presented in Section 6.7, based on the two application examples of *Connosseur*. Finally, this Chapter is summarized in Section 6.8.

6.1 Storage of Sequence Data

Biological databases, such as GenBank [7], Embl [109], Ensembl [19], etc., are built up in order to organize and distribute DNA and protein sequences data that are publicly available. These systems are large repositories containing information which can be simultaneously retrieved by many users. In this context, they fit to the definition of a traditional database given by Connoly and Begg [22]:

> *"A shared collection of logically related data, and a description of this data, designed to meet the information needs of an organization."*

The generation of a CNS database in the traditional way would require a large amount of conserved noncoding sequences which have been tested for regulatory functions beforehand. This kind of data is under way with the ease of sequencing the genomes of different organisms and the application of genome comparison approaches. But experimental assays in the wet-lab are still very laborious and take a long time to achieve concrete results. Furthermore, a traditional database would provide the user a list of previously computed CNSs, allowing only the retrieval of sequences. In order to automatically compute CNS from raw sequence information, we have developed *Connosseur*, which identifies conserved noncoding sequences starting from users specific genomic sequence data. To make use of the advantages of biological databases, *Connosseur* utilizes a database system to support the storage of the computed data, generating as product a

Repository of pCNSs. Chronologically, the data are generated by the annotation, comparison and filtering pipelines (bioinformatics tools cascade), and subsequently stored for further investigations (underlying database system). In this context, we give the following general definition of *Connosseur*:

> *"A collection of bioinformatics tools, which generates a repository of logically related information based on raw sequencing data."*

Connosseur is built up individually for each user. A single user processes the data, creating the Repository according to the individual investigation purposes. *Connosseur* does not support sharing of data. Thus we are not concerned on locking the system to avoid data conflicts during their computation. Similar to traditional databases, the created Repository of pCNSs can be accessed by multiple users by querying the data.

In the following, we introduce the underlying relational database system used by *Connosseur* for data storage.

The Underlying Database System

Databases are essential elements to share information within the scientific community. More and more people are creating their own databases, allowing colleagues to access their data directly. It is a flexible way to store information with easy retrieval procedures, i.e., just by applying query statements. In general, there are two types of database management systems (DBMS):

- Object-Oriented Database Management System (OODBMS): this type of system is consistent with object-oriented programming principles. It copes with complex objects, beyond tables of character data. This means, this database system handles data as objects rather than tables, providing access from text-format data over images to video files.
- Relational Database Management System (RDBMS): in this model, the information is stored in a collection of tables. *Relations* are used to describe information about the data contained in the database. *Relations* are physically represented by two-dimensional tables, where the rows (also called *tuples*) correspond to individual records, and the columns correspond to *attributes*.

In our work, we use a RDBMS, as the final Repository of pCNSs contains information that need to be related to each other. This kind of system is suited for creating repositories which need a great degree of flexibility to design future extensions. These characteristics match exactly our purposes. To establish the necessary relationships, the tables are labeled with unique identifiers, allowing the user to make connections between the data stored in different tables. This is a very important property when dealing with biological data, because common features of an element can be rapidly found through those established relationships.

6.2 Designing Connosseur in Three Phases

We design *Connosseur* in three main steps, constituting a conceptual, a logical, and a physical phase. They take into account the necessary bioinformatics tools within *Connosseur*, as well as the underlying relational database system of the generated Repository of pCNSs. The conceptual and logical design depends on the biological target data model. The first phase concerns the biological background of the system, and deals with the questions:

1. *what are the biological requirements?*
2. *what are the steps to fulfill the requirements?*
3. *what data should the generated repository contain?*

Based on the conceptual model, questions 1, 2 and 3 lead to the construction of the logical phase at a more technical level of the system, under a computer sciences point of view. In this second phase of the *Connosseur* design, we deal with the following questions:

4. *what is the appropriate architecture to fulfill the requirements?*
5. *what relations have to be established?*

The output of these phases is a global logical data model, consisting of a graphical scheme of the necessary biological data (the concept), an Entity Relationship Diagram (ERD) [22], a relational scheme or a flow diagram of the whole system (the logic), and a documentation describing the computation and storage of data. These features constitute the source of information for the third design step – the physical phase. This phase deals with the determination of the appropriate bioinformatics tools for satisfying the project's requirements, the datatype definitions of the entries in the Repository of pCNSs, and *how* the system will be implemented [47, 22, 81].

The biological requirements of *Connosseur* (conceptual phase) are delineated in Subsection 6.2.1, mapping this conceptual structure onto the logical model is presented in Subsection 6.2.2, and, finally, Subsection 6.2.3 gives an overview of the physical phase of *Connosseur*.

6.2.1 Conceptual Phase

The creation of a conceptual data model is the first step of the project's design. This modeling is done according to the users' requirements. To construct *Connosseur*, we gathered information from literature and discussions with molecular biologists, in order to reach the necessary background information for the development of this conceptual phase. The interest in searching for function in genomic regions inbetween genes and

coding exons increased with the publication of the human genome, in February 2001 [70]. The scientific community was surprised with the overestimation of gene content in the human genome before the publishing of the finished sequences. To justify the large amount of proteins in the higher vertebrates in general, several researchers appealed to different theories involving regulation of gene expression and translational control. It has been shown before that these noncoding regions can be successfully identified with comparative genomics approaches (see Chapter 3). Our goal is to automatically generate a list of pCNSs in the same way biologists usually do it manually. We separate the collected biological background information and requirements in three main levels, representing them schematically in Figure 6.1. These three levels are described in the following (satisfying questions concerning this phase):

a) Sequence Annotation: The insertion of genomic sequences into the system triggers a cascade of annotation techniques, dependent on the investigation purposes. Gene prediction and repeat finding constitute the basic components of the sequence annotation. Additional annotation refers to the exon/intron structure of known genes via the unsplicing of corresponding cDNA sequences onto the genomic sequences. Another optional kind of annotation is the utilization of a more detailed gene annotation file delivered by the user (see Figure 6.1a).

b) Sequence Comparison: Comparative genomics approaches enable searching for similarities between the given sequences. Conserved regions between both input sequences are found in different densities and rates in distinct genomic regions (see Figure 6.1b).

c) pCNSs Analysis: Sequences that code for proteins are highly conserved among organisms. From a list of conservations, scientists intuitively subtract the conserved elements which overlap known coding exons to get the noncoding regions. This method is called *normalization*, generating a list containing only potential conserved noncoding sequences. As CNSs can serve as signals (like promoters or enhancers), regulating gene expression, or influencing the splicing apparatus, they are found in different positions relative to genes or their exons. Classifying the pCNSs in defined locations, such as *upstream* or *downstream* to a gene, or even *intronic*, facilitates their further functional analysis (see Figure 6.1c).

During the process of developing a conceptual data model, each step of the model has been tested manually, in order to validate the approach against the user requirements. The positive result led us to model the next step, the logical phase.

6.2.2 Logical Phase

The conceptual data model created in the previous phase is refined and mapped onto a logical data model. While the first phase gives a general concept of the information content and the necessary steps to fulfill the requirements, the second phase considers how

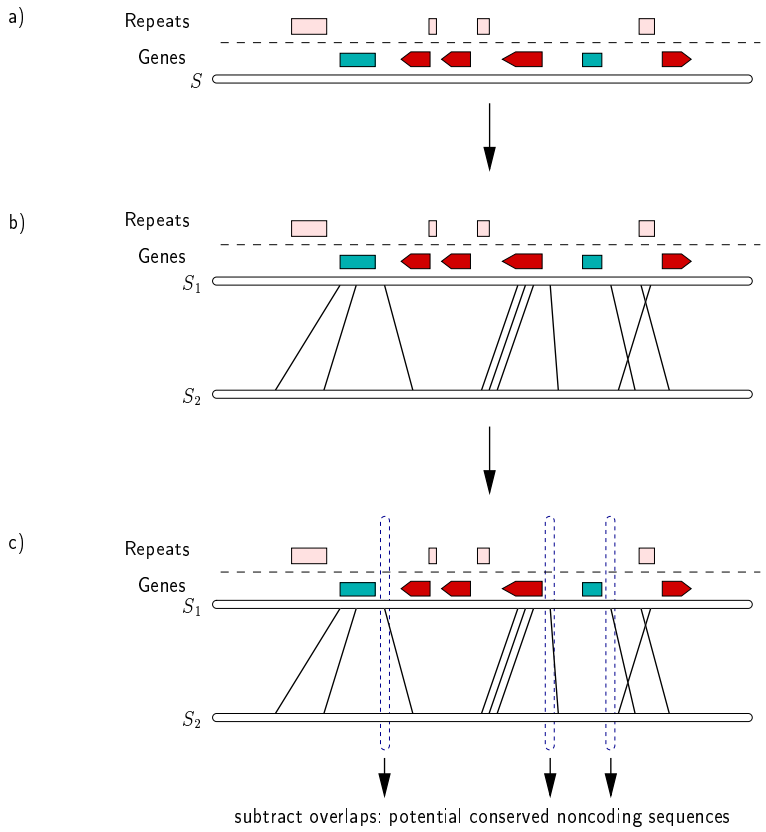

Figure 6.1: Conceptual Phase: a) sequence annotation; b) sequences comparison; c) pCNSs analysis. S_1: first genomic sequence; S_2: second genomic sequence. The graph represents the annotation and the pCNSs detection for S_1, but both sequences can be annotated. The user defines from which sequence (S_1 or S_2) the pCNSs are extracted.

to build up the general architecture of *Connosseur*, in order to give a logical structure to the system (according question 4). In addition, the logical phase is structured according to the data retrieval from the system, and *how* to establish the relationships between these data in order to fulfill the initial biological requirements (satisfying question 5).

The scheme of Figure 6.1 is refined into a data flow diagram, constituting the logical structure of *Connosseur*. This diagram (see Figure 6.2) shows how *Connosseur* generates the Repository of pCNSs via annotation techniques and comparative genomics. This Figure depicts the overall system architecture, based on the user requirements modeled in the conceptual phase. The squared boxes represent *information*, either provided by the user or generated by the triggered bioinformatics tools cascade. The oval boxes are *states*, in which the data is transformed or analyzed. The dashed arrows point towards the *pCNSrep* box, showing that all entered or computed information is stored in the Repository of pCNSs.

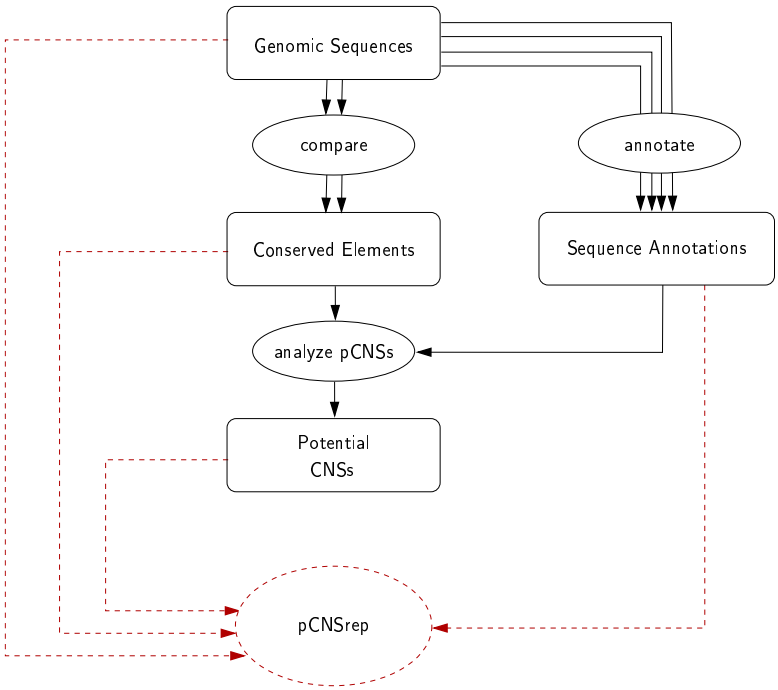

Figure 6.2: Flow diagram of *Connosseur*. Squared boxes represent inserted or produced data; oval boxes are states, in which the information is computed or transformed. The number of arrows stands for the relative number of data sets passing through each state. Dashed lines indicate the storage of computed results in the pCNSs Repository.

The states in oval boxes hold the corresponding tools that perform the transformations and analyses of the data. Some of them are triggered just by the presence of the previous information, others are initialized on the users command. The *Sequence Annotations* data, for instance, is generated through the *annotate* state as soon as the user enters a genomic sequence into the system. This step is done automatically for each new sequence entered, independent of other processing steps in the system. It concerns mainly the localization of repeated elements, as well as the identification of genes contained in the corresponding genomic sequence. This constant autonomous annotation of sequences is represented by four arrows indicating the processing in the *annotate* state. Yet the two arrows pointing to the *compare* state indicate the computation of *Conserved Elements* derived from only two entries of the *Genomic Sequences* data. This restriction is based on the underlying software for genomic comparisons, `vmatch`, which computes pairwise alignments. At this level, the user interacts with the system, determining which sequences from the *Genomic Sequences* data set are compared to each other. Finally, in the *analyze pCNSs* step, the single list of *Conserved Elements* produced in the previous state (represented by only one arrow in the flow diagram) is analyzed for the conditions 'coding' or 'potentially noncoding'. To achieve this classification, another arrow points

at the *analyze pCNSs* state, out of the *Sequence Annotations* box. This arrow represents the annotation corresponding to the genomic sequences involved in the comparison.

Both the *Conserved Elements* and the *Sequence Annotations* boxes are used to achieve the intended data restriction, because repeated elements are quite conserved among higher vertebrates. Biologists prefer to eliminate them from the conserved elements list, avoiding a large background noise of conservation. Nevertheless, the repeated elements information is still present in the Repository of pCNSs for further investigations regarding the correlation between divergence in repeated elements and in other noncoding sequences. Furthermore, since gene predictions and cDNA localizations refer to coding regions, they are also eliminated in the *analyze pCNSs* state. This state generates the *Potential CNSs* data. This data mining step refers to the *normalization* procedure, as already mentioned in the conceptual data model description. Also included in this state is the analysis of the resulting pCNSs with regard to their position relative to genes or exons. The conserved sequences in noncoding regions will be called *intronic*, *upstream*, or *downstream* to a gene, according to its relative arrangement to the neighboring coding sequences. More details on the individual states are given in Section 6.4.

The logical phase also determines the architecture of the generated Repository of pCNSs. As it contains biological data that need to be connected to each other, the utilized underlying database system should provide a structure which supports the necessary query statements for data retrieval. This requires the establishment of relationships between different sequence features contained in the Repository of pCNSs, in order to extract biologically meaningful information from the initial raw sequences. Furthermore, several bioinformatics programs provide their outputs in a tabulated format. The attributes contained in those tables are text or numeric datatypes. These reasons led us to chose a RDBMS to support the data storage, covering all needs for information retrieval from the Repository of pCNSs.

The tables are filled with data as a result of the cascade of bioinformatics tools. Genomic sequences to be compared and searched for conserved segments constitute the major data handled by *Connosseur*. Sequences of known cDNAs comprised in the input genomic sequences are also important. Beginning with this minimal initial data, the information to be stored in the Repository of pCNSs is produced as the steps of sequence annotation and comparison are being completed in *Connosseur*.

In addition to the general scheme of the flow diagram, another result of this logical phase is the Entity-Relationship Diagram (ERD), shown in Figure 6.3. It answers the question 5 (see page 82), asking which relationships are established between the information computed in *Connosseur*. The relations are drawn as squared boxes, consistent with the flow diagram. Diamonds labeled with ‘is a’ define fixed features of the *Genomic Sequences*, being connected to this relation via solid lines. Diamonds labeled ‘is calculated from’ represent the calculation of data that depend on the user defined pre-computed information, from tables that are connected with dashed lines.

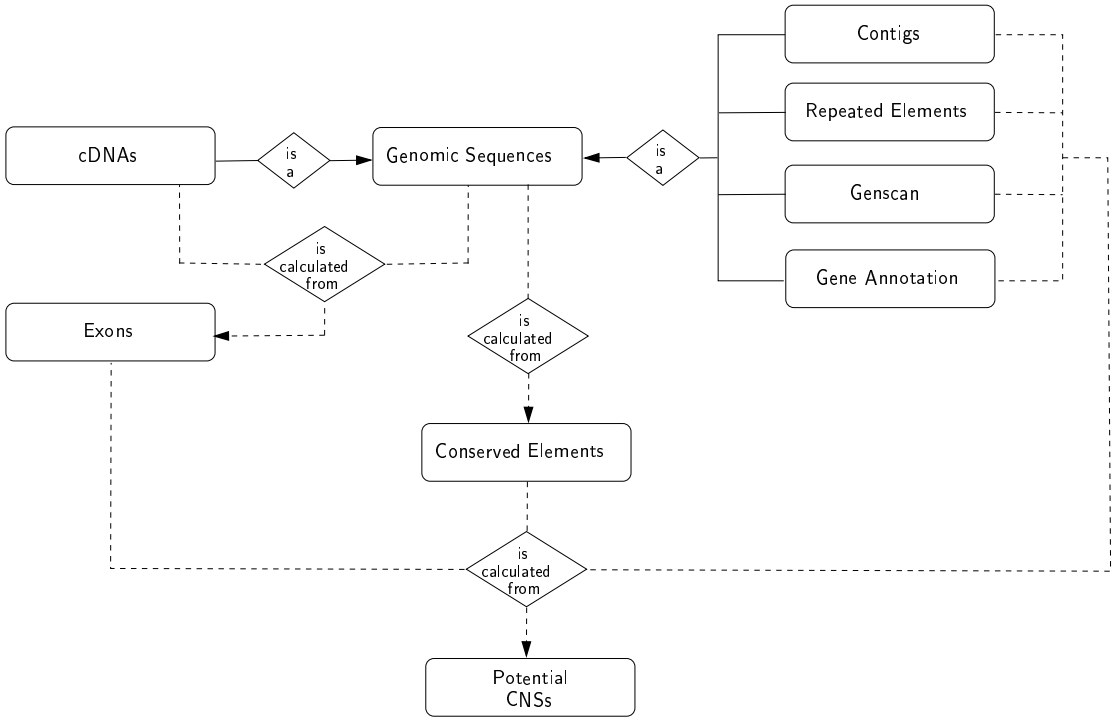

Figure 6.3: Logical Phase: the Entity-Relationship Diagram.

6.2.3 Physical Phase

The physical phase is the last phase of the *Connosseur* design. It considers the lowest level of abstraction and determines *which* computational tools will constitute *Connosseur*, *how* they are integrated, and *which* datatypes should be defined for the entries in the generated Repository of pCNSs.

In order to satisfy the biological requirements of sequence annotations, `RepeatMasker` is used to localize all repeated elements in the input genomic sequences. For the identification of coding sequences, `GENSCAN` has been chosen as gene prediction software. To unsplice known cDNAs onto the corresponding genomic sequences, and to carry out the comparative genomics step, the `vmatch` program is utilized as an appropriate tool. All three programs are presented in Section 6.3.

As RDBMS we chose PostgreSQL [81] to support the data storage delivered by the computational cascade. The advantage of this system is its simple and logical structure. PostgreSQL accepts user-defined datatypes and functions, as well as a broad set of Structured Query Language (SQL) functions and types. The Repository of pCNSs is initialized in the beginning of *Connosseur*, by setting up the tables which constitute the underlying database. These tables are constructed via a shell script. Their entities,

attributes, as well as the data type definitions are outlined in Appendix C. Afterwards, shell and Perl scripts automate the subsequent computational steps in order to call the underlying programs, parse the output data, and fill the tables with the corresponding information. The connection of *Connosseur* to the Repository of pCNSs is performed through the Perl Database Interface (DBI) [24].

6.3 Cascade of Bioinformatics Tools

Several programs, such as `RepeatMasker` (Smit, A. & Green, P., unpublished) and `GENSCAN` [15], are today very popular and widespread in the biological community. Without an automatic system, they have to be used separately from each other. There is no program that automatically joins these programs with computational tools for sequence comparison. In order to fill this gap, *Connosseur* builds up a cascade in which sequences are automatically annotated for the localization of repeated elements and the prediction of genes, using `RepeatMasker` and `GENSCAN`, respectively. Followed by this basic level of the annotation process, the generated *sequence annotation pipeline* is joined with a *comparative genomics pipeline*. This step, in turn, uses the versatility of `vmatch`, employing it in the comparison of genomic sequences, besides other steps during the sequence analysis [68, 67, 66]. In the following, we shortly describe these three main underlying bioinformatics tools in *Connosseur*.

`RepeatMasker` All eukaryotic genomes contain repeated elements, such as LINEs, SINEs, LTRs, etc. These mobile elements vary between genomes of different species in the proportion and activity of the classes of elements [5]. Those repetitive elements comprise up to 50% of the human genome, with few quantitative differences relative to the mouse genome. Usually, protein coding exons do not contain such elements, and untranslated regions (UTRs) may contain about 10% of repeated elements [80]. This leads to the speculation that conserved elements with potential regulatory functions are located in noncoding and nonrepetitive genomic segments [18].

`RepeatMasker` compares the input DNA sequence against libraries of repeated elements. The alignments are performed by the program `cross-match`, an implementation of the Smith-Waterman-Gotoh algorithm [108, 50] developed by P. Green (unpublished). The libraries provided with `RepeatMasker` are extracted from the interspersed repeat database of RepBase [61].

Masking those repetitive elements in genomic sequences before comparison contributes to the acceleration of the comparative matching task, as the fraction of repetitive elements is ignored in the alignment. The exclusion of these segments also provides a much better sensitivity for identifying CNSs with potential functional roles. `RepeatMasker` screens DNA sequences for interspersed repeats and low complexity regions. One of the outputs is a modified version of the input sequence, with repetitive regions being marked so that they are ignored in the subsequent alignment. The program delivers also a table

with detailed annotation of the repeated elements contained in the input DNA sequence, specifying the classes of repeats and their absolute positions. This information is stored in the Repository of pCNSs.

`GENSCAN` At the moment, `GENSCAN` is the most widely used gene prediction program. It has a higher accuracy than other predictive methods with 75% to 80% exact identified exons from a set containing human and other vertebrate genes. Another feature that overcomes the weaknesses of other tools is the ability to analyze potential genes in both DNA strands. In combination to this double-stranded model, `GENSCAN` can handle sequences that may contain partial or complete genes, multiple genes, or even no genes at all. This turned out to be useful for analyzing short pieces of sequenced DNA, as well as long human genomic contigs. They still represent draft sequences, and, therefore, may contain incomplete sequences of genes.

`GENSCAN` is based on a probabilistic model, which captures general and specific properties of sequence composition of the distinct functional units of eukaryotic genes, such as exons, introns, promoters, etc. The authors advert that the identification of promoters could still be improved with new models, increasing the prediction accuracy. They focused on constructing a tool which is flexible enough so that potential genes are not missed just because they lack a sequence similar to our preconceived notion of how a promoter looks like. Another feature of `GENSCAN` is the utilization of methods that model functional signals in DNA. These are used to determine splice site signals, as the vast majority of exons are internal exons and therefore begin with an acceptor site and end with a donor site. The initiation exon is defined from the translational start up to the donor splice site and the termination exon is determined by the acceptor splice site until the stop codon [15].

`Vmatch` The `vmatch` program has already been introduced in Section 3.2.3. Its main differences to `REPuter` and technical improvements have been described in detail in Chapter 4. The wide range of applications of this program provides the necessary flexibility to employ it also in *Connosseur*. Besides the annotation of known transcribed exons, `vmatch` is also used for the comparison between two genomic sequences, which is the main step towards the identification of pCNSs.

The genome from the organism to be compared to other sequences is considered the database sequence. Its index needs to be constructed only once, being pre-processed for any subsequent matching task and stored as a collection of files. `Vmatch` generates an output in a table-like format, as `RepeatMasker` and `GENSCAN` do, simplifying the storage of the computed data into the tables of the Repository of pCNSs. Another decisive characteristic of `vmatch` is the interactive visualization of the match file by `GenAlyzer`. Combining the views of the match graph with the sequence annotation features in the annotation graph, biologists can get a better idea of the analyzed sequence structure. The ability to zoom into specified regions in the graph, as well as to select the displayed

matches by length allows the user to get detailed insights into the genomic organization in both compared sequences. The visual localization of conserved regions is more comprehensive for biologists, rather than just a corresponding list of numeric positions. In general, the user decides the adequate threshold for its investigation purposes by means of the visualization of processed matches.

6.4 Developing Connosseur

The cascade of bioinformatics tools, i.e., the sequence analysis and the feature computation in *Connosseur*, is not totally independent from the information storage into the Repository of pCNSs. Genomic sequences used in the system can be very large, so they are preferably maintained as flat files rather than in the underlying database tables. This implies that *Connosseur* launches all computational tools and establishes the interface for storing the results in the Repository of pCNSs. However, the computed output usually does not fit the tabulated database structure, needing to be parsed first into an adequate format. These parsers are also provided in *Connosseur*. Sometimes, the parsing requires temporary connection of *Connosseur* to the Repository of pCNSs, in order to extract the unique identifiers, called *primary keys*, of a specific table, the *parent* table. These attributes are used as *foreign keys* in another table, establishing the relation between the *parent* and the *child* tables. This kind of information is crucial for future query processing, in order to allow for feature connections of different tables.

To launch the initialization of the database containing all table arrangements, we provide the necessary information, like server, database and user names, as well as the path indication and the table specifications file. This section deals with the detailed delineation of the three parts of the development of *Connosseur*. Maintaining the consistency with the design description in Section 6.2, the sequence annotation pipeline is explained in Subsection 6.4.1, followed by Subsection 6.4.2, where the calculation of conserved elements is described. Finally, the third part handles the analysis of pCNSs, in Subsection 6.4.3.

In Figure 6.4, a modified version of the flow diagram introduced in Section 6.2.2 is shown. The three distinct states (*annotate*, *compare* and *analyze pCNSs*) are numbered and highlighted in a colored background, facilitating the localization in the whole system, when discussing the details of each computational step in the following subsections.

6.4.1 Part One: Sequence Annotation Pipeline

The sequence annotation pipeline involves two boxes from the information data as depicted in the general *Connosseur* scheme in Figure 6.5. The data and the state included in this step of the computational cascade are highlighted in black. The boxes and states which are not considered here are drawn in gray. There are three possible kinds of sequence annotations: a) positions of repeated elements are defined, gene prediction is

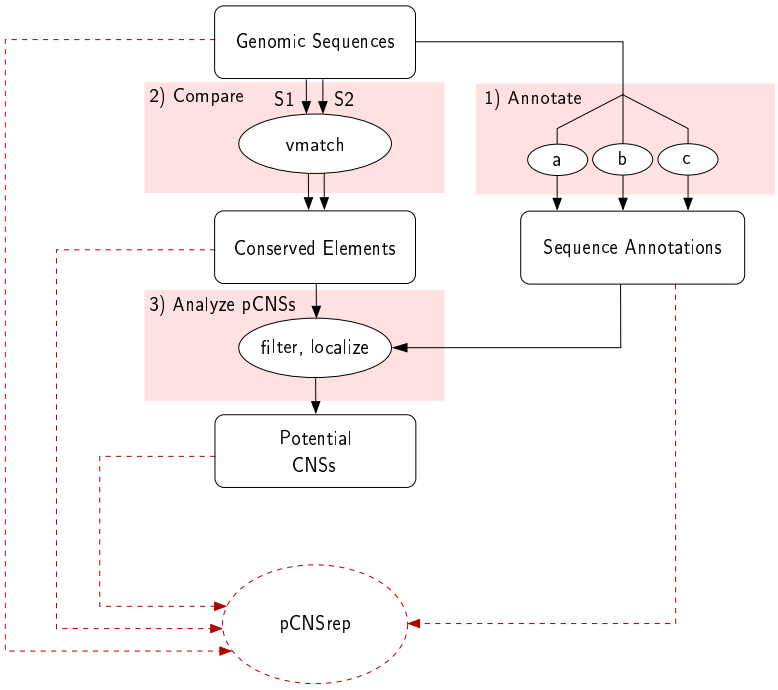

Figure 6.4: General flow diagram, highlighting the three computational steps with a colored background. Squared boxes represent inserted or produced data; oval boxes indicate the underlying programs in each state, which compute or transform the data. In the *annotate* state, the letters indicate three different kinds of annotation that can be carried out: a) `RepeatMasker` and `GENSCAN` annotations; b) cDNA unsplicing and c) gene annotation file provided by the user. S_1 and S_2 are database and query sequences, respectively, used for the `vmatch` matching task.

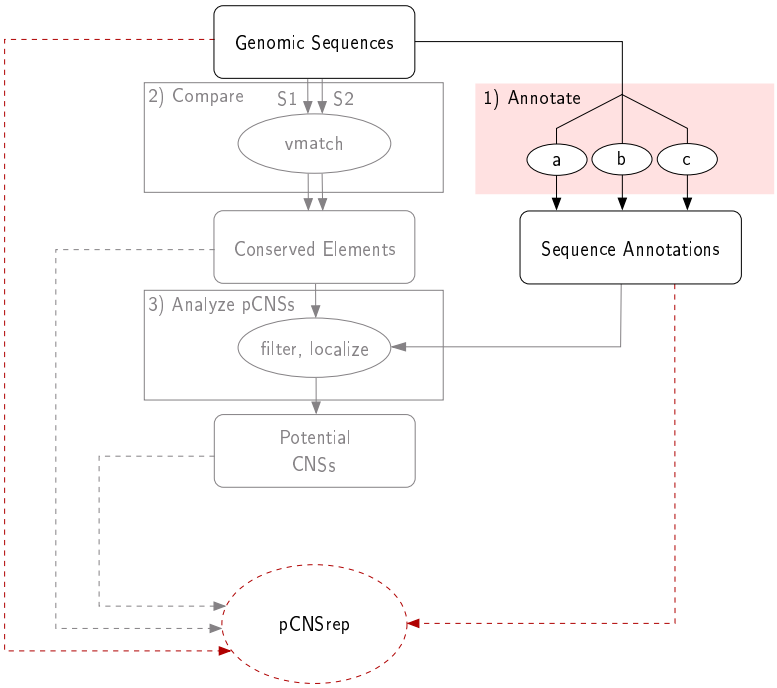

Figure 6.5: Flow diagram highlighting the *annotate* state. The gray fields are not considered in this step. Three kinds of annotation can be carried out: a) `RepeatMasker` and `GENSCAN` annotations; b) the unsplicing of cDNAs onto the genomic sequence; c) the insertion of a gene annotation file provided by the user.

determined; b) known cDNAs are unspliced onto the genomic sequence; c) user's specific gene annotation is entered in a separate file. The first annotation type is automatically initialized with the insertion of a genomic sequence into the system. The second and third types are automatically carried out if the user enters the necessary information, like the sequence of known cDNAs or the separate gene annotation file. In this way, the input genomic sequence has always a basic annotation available, in form of masked repeats and predicted genes. This is very useful for regions in the genome which do not contain known genes, or the gene sequences are not completely available. Consequently, the latter two types of annotations can be used independently from each other, but always together with the first one.

Before any kind of annotation is carried out, the first step is to insert the main features of the genomic sequences into the system. This is the basic initial information needed to trigger the bioinformatics tools cascade. *Connosseur* supports only input files in single or multiple fasta format. As mentioned before, the sequence itself remains as flat file in the *project path*, representing the directory containing sequence files. The computational tools access the correct sequence file via its *filename*. In general, the filenames can get very cryptic in their designations, so the user must chose a devised *name* that is easy to

remember and represents an unique identification for the file under consideration. This specific name works as a *reference* to the corresponding file throughout *Connosseur*. An additional feature for entering a genomic sequence into the system is its accession number, allowing the access of further information in publicly available biological databases. Finally, the species the input sequence belongs to is also stored, facilitating further tasks of sequence comparisons, for instance. All these entries describe specific features of the given sequence, and are used as attributes of the relation *Genomic Sequences*. Another important characteristic of the input sequence is its length in base pairs. This is automatically calculated with the sequence insertion. The number of contigs of each input sequence is determined, providing the information whether the sequence is a multiple fasta file or not. Entering the genomic sequence into the system triggers the bioinformatics tools cascade, beginning with the first, default annotation type. All three types of annotation are described in the following.

First Annotation The first kind of annotation begins with the contig verification of the input sequence. A tabulated file containing the corresponding contig information is generated. Part of the data needed as attributes (number of segments and length) is a subordinate product delivered by the program `mkvtree`, which calculates the index structure of a sequence in form of an enhanced suffix arrays (see Section 4.1). Computing the index of the input sequence in this step gives us two advantages: first, the necessary contig information is delivered as a consequence, and second, the index files are already computed for this sequence and stored on file for further matching tasks. The contig file in a table-like format is then parsed to get the appropriate configuration for storage into the Repository of pCNSs, and is finally inserted into the corresponding relation of the system. A graphical representation of this annotation step is shown in Figure 6.6. Still in this first kind of annotation, the input genomic sequence is analyzed by the programs `RepeatMasker` and `GENSCAN`. Both are installed locally, avoiding Internet interactions and permitting the computation of large data sets. `GENSCAN` is run with default options, and `RepeatMasker` adapts its parameters to the species the genomic input sequence belongs to. The default setting of `RepeatMasker` that substitutes masked nucleotides by Ns is changed into *small caps*. This program also delivers as output a table containing the repeated elements' positions, so that substituting these regions by *small caps* allows the user to visualize what is behind the masked segments at the sequence level. Again, the output files of both programs are parsed and then inserted into the database. Finishing this first annotation step, the start and stop positions of contig delimitations, predicted genes and repeated elements are transformed in the *visualization annotation format* (VA file) of `GenAlyzer`. This file can be uploaded in `GenAlyzer`'s visualization graph. This preprocessing will be useful for visualizing the subsequent results of matching tasks accomplished with the corresponding input sequence.

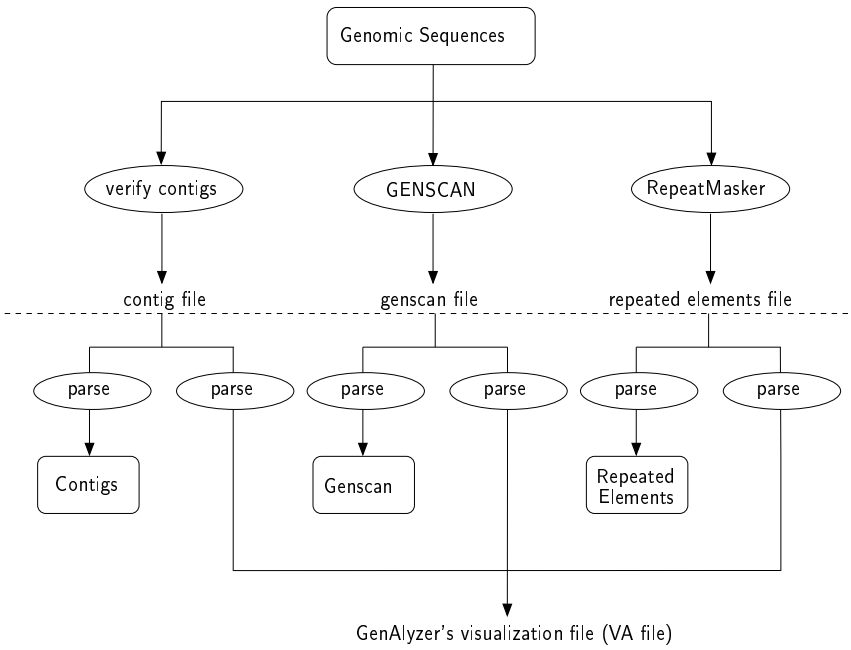

Figure 6.6: First sequence annotation pipeline. The genomic sequences are automatically annotated for contigs presence, gene predictions, and repeated elements classification. The output files are parsed both into the corresponding Repository tables format and the VA file, which provides the annotation graph visualization in `GenAlyzer`.

Second Annotation This kind of annotation refers to the unsplicing of cDNAs onto the appropriate genomic sequence. These are DNA sequences complementary to the corresponding mRNAs. As mRNAs originate from the genomic sequences, representing the exons of the gene in question, mapping their sequences onto the genome leads us to an overview of the corresponding gene structure. This approach has already been demonstrated in detail in Chapter 2. The procedure takes place by providing the sequences of cDNAs corresponding to genes which are known to be localized in the given genomic sequence. The features of those cDNAs are entered into the *cDNA* table in a similar way as the genomic sequences. Their filename is delivered, as well as their accession number. In order to combine different relations in the database, the user also enters the same specific name as entered before in the input sequence insertion. This permits the system to conect the cDNAs to the genomic sequences, computing automatically the matching task via `vmatch`, using successively each cDNA sequence belonging to the input sequence in question. The input sequence is used as *database sequence*, as its index has already been computed in the first annotation step, and the corresponding cDNA sequences are used as *query sequences*. A general overview of the matching task is shown in Figure 6.7a.

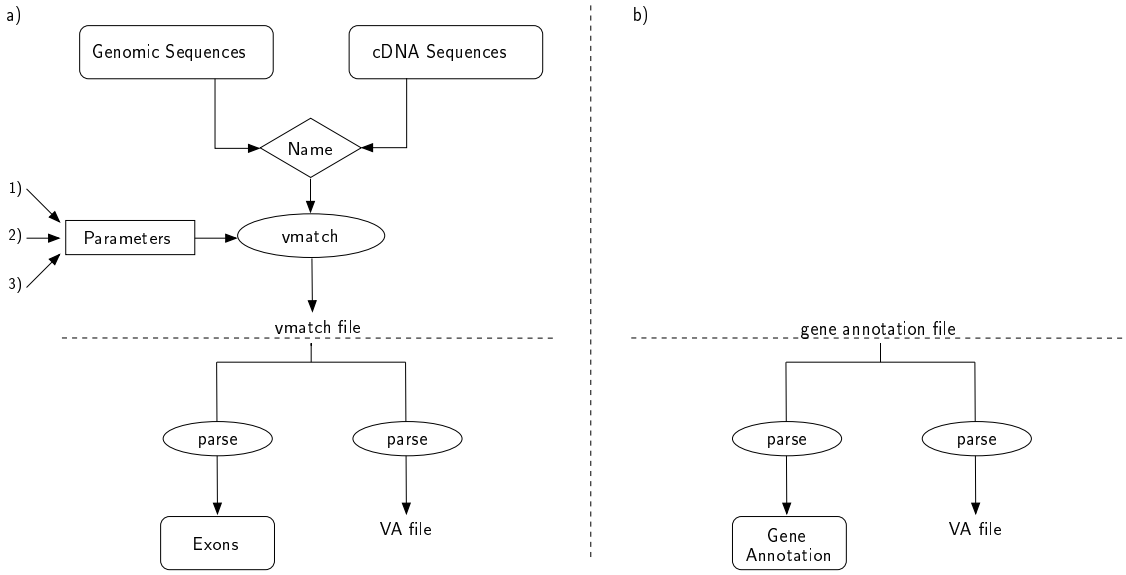

Figure 6.7: Second (a) and third (b) sequence annotation pipelines. a) cDNA sequences are unspliced onto the corresponding genomic sequence. The diamond labeled with 'Name' works as a *reference*, selecting all cDNAs that correspond to the given input genomic sequence. The user enters the parameters for the matching task either via the command line (1), or the default settings (2), or the query environment (3). b) The user enters a specific gene annotation file, which is parsed into the corresponding database table format and the appropriate VA file format for the `GenAlyzer`'s visualization.

The matching task is adapted to three different kinds of users. First, if the user is well acquainted with `vmatch`, the program parameters can be entered directly via a configuration file containing all the necessary `vmatch` parameters. If this is not the case, the user can either run the program with default settings, or use the third option, a query environment. Here, the system asks questions about the chosen thresholds, and the user answers are used for the parameter settings. This option takes into account all necessary argument dependencies as established by Kurtz [66]. The values of the default option have been chosen based on our own experiences with this kind of matching task. Still, in each case described above, the user should launch the `GenAlyzer` visualization of the match file, like depicted in Figure 6.8. This ensures that the chosen parameters cover the entire cDNA sequence in a non-redundant way, i.e., without generating too many short overlapping matches. If a noisy background is generated, the matching task should be recomputed with different thresholds. After the optimization of the unsplicing procedure, the exon/intron structure of the corresponding gene is roughly determined. The resulting `vmatch` output is parsed and stored in the *Exons* table in the system. The exons' start and stop positions are important for the determination of potential CNSs.

The thresholds of the individual matching tasks are also stored in the Repository of

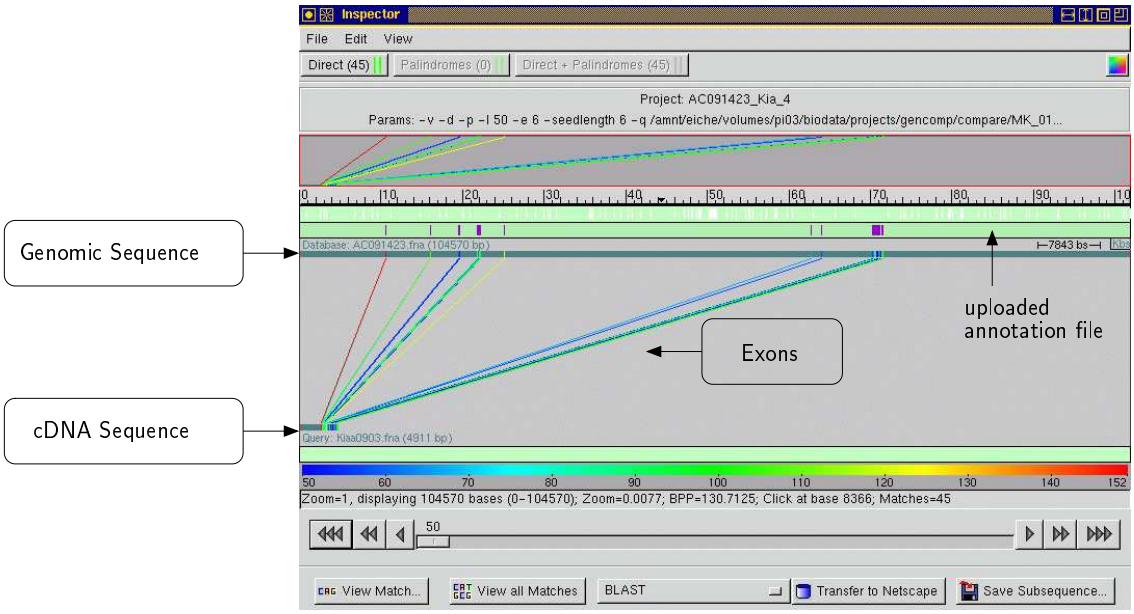

Figure 6.8: `GenAlyzer`'s visualization of the second sequence annotation pipeline. The unsplicing of the cDNA sequences onto the corresponding genomic sequence is represented by the matches, constituting the exons information.

pCNS, including the parameter values, database and query sequences, and the associated match file name. In this way, a history of the computed tasks is maintained, facilitating the understanding of the resulting matching pattern. Moreover, an analysis of the parameter settings for a specific computation can be advantageous if the matching procedure needs to be recomputed under different stringencies. Unique identifiers also relate the matching history to the appropriate rows in the *Exons* table, being here used as foreign keys. For every matching task entry, there can be *many* corresponding entries in the *Exons* table. In this way, a relationship of *one-to-many* (1:N) is established (see Figure 6.9). This feature is only an example, and contributes to the *normalization* of the database, where data redundancy is avoided. Finally, as in the first kind of annotation, the unspliced matches' positions are parsed into a VA file for the `GenAlyzer` visualization (see Figure 6.8). In most of the cases, the correct transcriptional start is not known in such cDNA sequences. UTR regions are often incomplete and can not be explicitly separated from coding exons. If the determination of conservation between UTRs is indispensable for the specific investigation purposes, the third kind of annotation would be more appropriate.

Third Annotation An alternative annotation permits the user to enter a specific gene annotation file containing, for instance, start and stop positions of 5' and 3' UTRs, as well as coding exons (see Figure 6.7). This file is inserted into the database system in

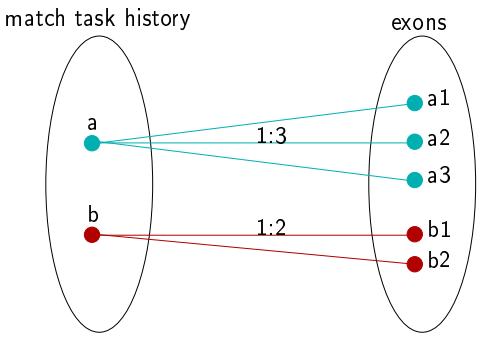

Figure 6.9: One-to-many relationship (1:N) between the matching tasks parameters and the *Exons* table.

order to provide more specific information about the structure of genes localized in the genomic sequence of interest. If such a structure is not available or well defined, the second annotation type is more appropriate, guaranteeing at least the generic exon/intron information. The same approach for generating the `GenAlyzer`'s VA file is used as in the first two annotation steps.

Parsing Outputs There are three steps in the parsing process of the output files from `RepeatMasker`, `GENSCAN` and `vmatch`, in order to fit them to the Repository tables' format. First, the column separators from the tables in the flat files are transformed into single tabulators. These are used as delimiters between adjacent columns, for inserting the data in the Repository of pCNSs. Second, the *System* table is used for storing the last used unique identifiers of *Connosseur*'s Repository relations. As the identifiers have to be unique, the *System* table is updated each time new features are computed. The third step in the parsing procedure is the establishment of the relations between parent and child tables. Child tables get the corresponding primary keys as foreign key attributes. This correspondence is set up based on the specific 'name' entered as common, unique attribute in the *Genomic Sequences* table.

6.4.2 Part Two: Sequence Comparison – Conserved Elements Calculation

The conservation between species, which has already been introduced in Chapter 3, can be computed utilizing different bioinformatics tools. Although all of them are based on alignments, our decision to use `vmatch` for comparing different genomic sequences is supported by its fast computation and user-friendly interface. Two points emphasize the importance of the output as a graphical visualization: first, the output data sets can get very large when handling genomic sequences; second, a low threshold is often needed for detecting conserved segments in noncoding regions, which normally increases

even more the output data sets. In conclusion, the utilization of `vmatch` and the output visualization supported by `GenAlyzer` is appropriate for the identification of pCNSs. The general flow diagram of *Connosseur* is represented again in Figure 6.10, focusing on the data and state boxes important in this comparison step.

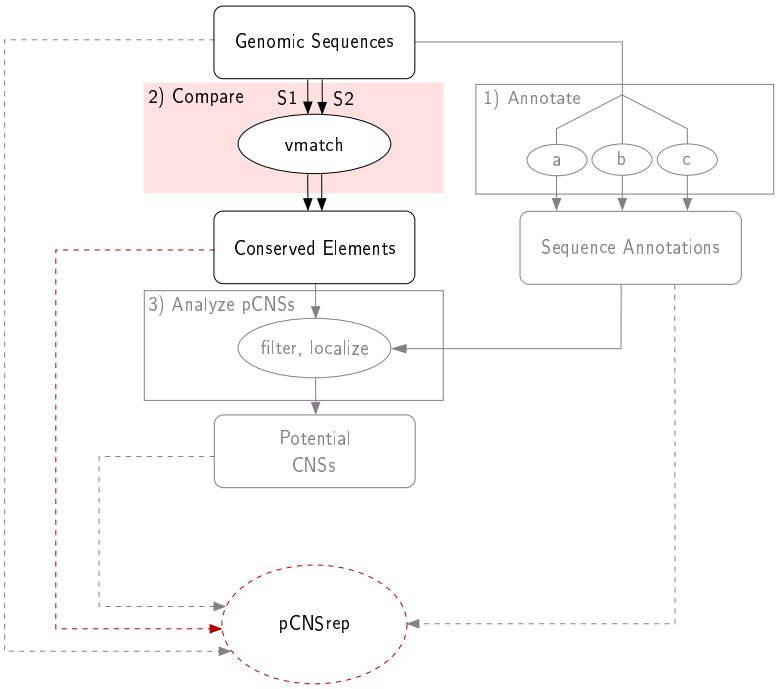

Figure 6.10: Flow diagram highlighting the *compare* state. The gray fields are not considered in this step. S_1 and S_2 represent the database and query genomic sequences, respectively, used for the comparison.

A more detailed view of the procedures in the *comparison* state involving the *Genomic Sequences* as well as the *Conserved Elements* information data boxes is depicted in Figure 6.11 (as already mentioned, the underlying tool in this step is `vmatch`). Both arrows coming out of the *Genomic Sequences* box and pointing at the *Conserved Elements* data indicate that two sequences from the set, S_1 and S_2, are being compared in each run. The choice of the sequences defines which species will be compared. This decision is made based on the background level of similarity between the species, and, of course, on the investigation purpose. Both genomic sequences had their index structure automatically computed before (in the first kind of sequence annotation). This pre-computation gives the user the flexibility to chose which sequence is going to be the *database* sequence S_1 for the matching task, and which one the *query* sequence S_2.

The comparison procedure in this step is analogous to the second kind of sequence annotation described in Subsection 6.4.1. Similar to the cDNA unsplicing onto the genomic sequences, `vmatch` is now employed to match sequence S_2 to sequence S_1. This

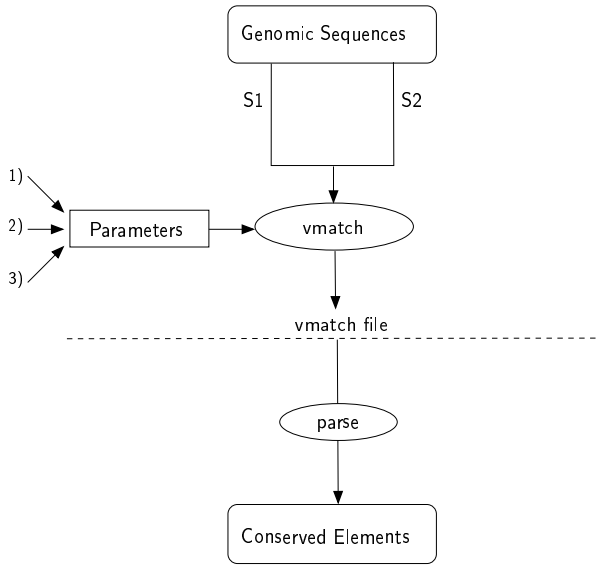

Figure 6.11: Sequence Comparison step: Conserved elements calculation. Two sequences, S_1 and S_2, are chosen as database and query sequences, respectively, for the matching task using `vmatch`. The parameters can be set either via a `vmatch` configuration file (1), or the default settings (2), or the query environment (3). The output file of `vmatch` is parsed into the appropriate database format and inserted into the corresponding table.

part of the *Connosseur* cascade is shown in Figure 6.11. Again, the user has three ways to set the parameters of `vmatch` for the matching task: either via the `vmatch` parameters file, or by starting the query environment, or by default values.

After the generation of the list of conserved elements, the output is parsed and inserted into the corresponding database table. In this case, the VA file is not generated, as the computed matches represent the conservation between distinct sequences, rather than features for their annotations. However, the resulting repeat graph can be visualized by launching `GenAlyzer` (see Figure 6.12) and uploading the corresponding VA files for sequences S_1 and S_2, which where generated before in the sequence annotation pipeline. The VA files are always constructed for the database sequence, i.e., for the annotation graph of the top line of the match graph. To visualize the annotation of the query sequence (bottom), the user has to adapt the VA file to the graphical layout.

Analyzing the match graph together with the annotation graph, the user can decide if the parameter settings satisfy the research aims. If only a few conserved segments are found with the used threshold, or they all match coding sequences, the matching task can be recomputed, with a lower stringency. With the recomputation, the output of the former task is automatically deleted from the *Conserved Elements* relation, avoiding obsolete information in the Repository of pCNSs. The recalculation of the match task

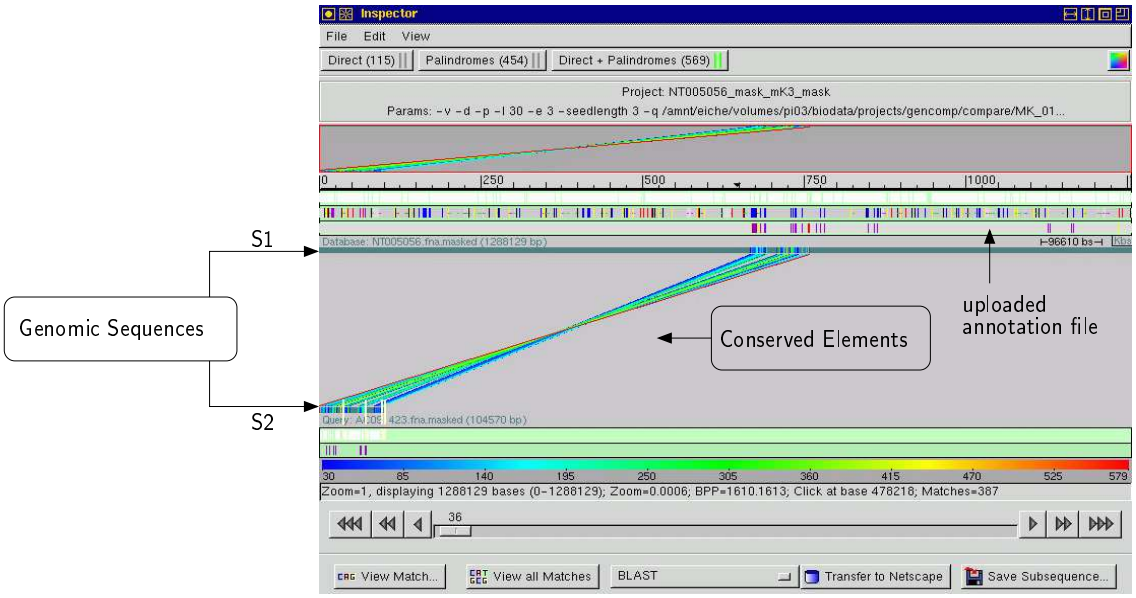

Figure 6.12: Sequence Comparison step: Conserved elements calculation (ctd.). `GenAlyzer`'s visualization of the conserved matches between the database (S_1) and the query (S_2) sequences.

occurs in a similar way as described previously for cDNA match files (see Figure 6.9).

This part of the system and the underlying programs are flexible, permitting the observation and visualization of intermediate results. By analyzing pre-computed results, the user can adapt the investigation at any level, even by trying to compare different combinations of organisms. This approach can deliver interesting clues by relating species with distinct evolutionary distances to each other.

6.4.3 Part Three: Conserved Noncoding Sequences Analysis

The *Connosseur* cascade proceeds with the analysis of the potential conserved noncoding sequences. The data and state boxes necessary for this step are highlighted in the general scheme in Figure 6.13. There is only one arrow at the *analyze pCNSs* state, as each cycle of the previous comparison state generates only one list of *Conserved Elements*, which is then analyzed for the condition 'noncoding'. In this third part of the system, two distinct programs are responsible for the generation of the *Potential CNSs* table: `filter` and `localize`. Although they are related to each other, we describe them separately for a better understanding.

`filter` The list of conserved elements generated in the previous step contains all regions that represent a positive evolutionary pressure according to the chosen parameters.

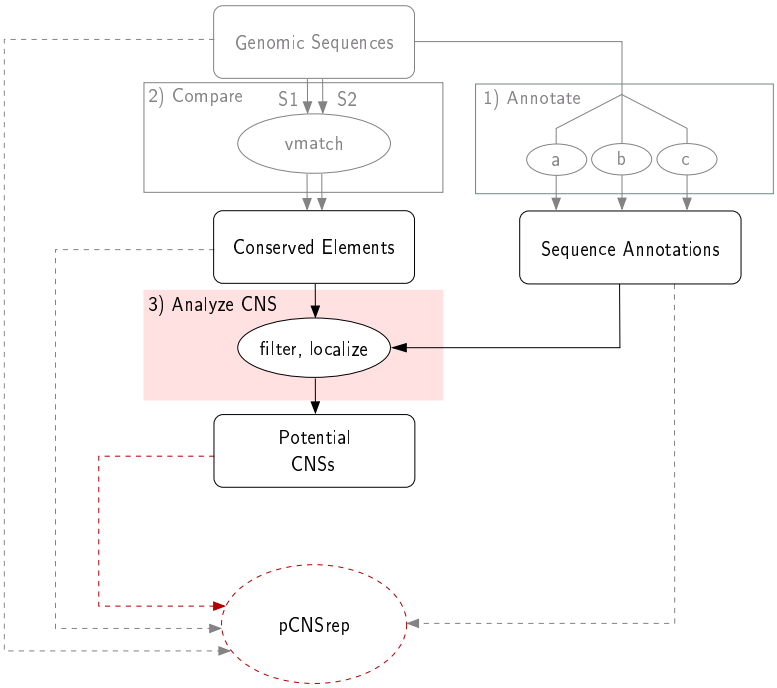

Figure 6.13: Flow diagram highlighting the *analyze pCNSs* state. The gray fields are not considered in this step. In the *analyze state*, the underlying programs are responsible for extracting the pCNSs from the *Conserved Elements* list, and localize them relative to annotated genes.

Repeated elements, as well as protein coding regions, have high conservation rates among species, so that the segments in the *Conserved Elements* data set that overlap such regions are filtered out. The remaining subsequences lie neither in repeated elements nor in known protein coding regions, according to the available annotation. In order to maintain a default line throughout *Connosseur*, the information from the *Repeated Elements* and the `GENSCAN` relations are considered in the filtering process as default data for this normalization step. The filtering process can be extended beyond the default, depending on whether or not the genomic sequences have also been annotated with the second or third kind of annotation procedures described earlier in this Chapter.

For an overview of the procedure, we concentrate on the default method. First, each entry in the *Conserved Elements* list is checked against the repeated elements positions, whether it overlaps or even is totally contained in one region of the compared data element. If an overlap is found, the corresponding fragment is cut out, and the remaining, non-overlapping region (*remaining conservation*) is delivered as output in a new list, the table of *Potential CNSs* data. The conserved elements that do not overlap at all are transfered into the repository without any changes. This pre-filtered list constitutes the new input list of conserved elements for the next filtering step, using

the GENSCAN predicted exon positions. In this case, only predicted initiation, internal, and termination exons are considered, as promoters and polyadenylation signals do not represent coding regions. The normalization procedure is repeated, but with a different interpretation of the overlaps. As already discussed in Chapter 5, pCNSs that present regulatory functions can be localized in several regions relative to the genes they influence. Investigations about conservations in the splicing machinery involve the analysis of the direct surroundings of exons. Many conserved elements that lie in coding regions are extended over these exons boundaries. For this reason, the filtering step allows for the determination of a minimal *remaining conservation* that is admitted in the output. This procedure normally results in many small fragments of pCNSs, after the overlap was cut-out. However, it concedes a detailed investigation about splice sites conservation between different organisms, giving interesting insights into the evolution of gene structure. If this does not fit the individual research purposes, the user will set a high *remaining conservation* threshold, getting only pCNSs which lie totally in introns or in intergenic regions.

The graph in Figure 6.14 depicts the steps of the entire default filtering procedure. The horizontal bars represent the DNA sequence (S), where colored blocks show the positioning of repeated elements (REs), conserved elements from the match file (CEs), or predicted exons by GENSCAN (G). Filtering out the repeated elements, vertical dashed lines are drawn to indicate the overlap boundaries (Figure 6.14a). These overlaps are cut-out (Figure 6.14b), transforming the match file CEs into CEs', used in the exons filtering step (Figure 6.14c). Again, vertical lines set the positions of exons' boundaries, and the filtering step is repeated. Note that this time, the *remaining conservation* is checked before delivering the segment to the output. The final result from the `filter` program in the default mode is represented now by CEs" in Figure 6.14d, constituting the data stored in the *Potential CNSs* table of the Repository of pCNSs.

In order to visualize these results in GenAlyzer's match graph, a modified match file is generated based on the filtered output. All matches that represent a total overlap to any kind of sequence annotation feature are removed in the GenAlyzer's visualization. In the new match file, only the matches that have been subjected to cut-out operations are maintained, in their original length. This approach eliminates the background noise of the initial match graph, usually generated by matches within repeated elements. The final pCNSs are annotated in the VA file, allowing a comparison to the original matches, and the remaining conserved regions (see Sections 6.5 and 6.6 for examples).

`localize` Continuing the default mode of *Connosseur*, the localization of the computed pCNSs is determined based on their positioning relative to the exons predicted by GENSCAN. In addition, if the `filter` program was also employed upon the *Exons* and/or the *Gene Annotation* data, the pCNSs will also be positioned according to this information. This step is a complement to the `filter` program, as this already checks the positions of the conserved elements and the compared data. Consequently, to de-

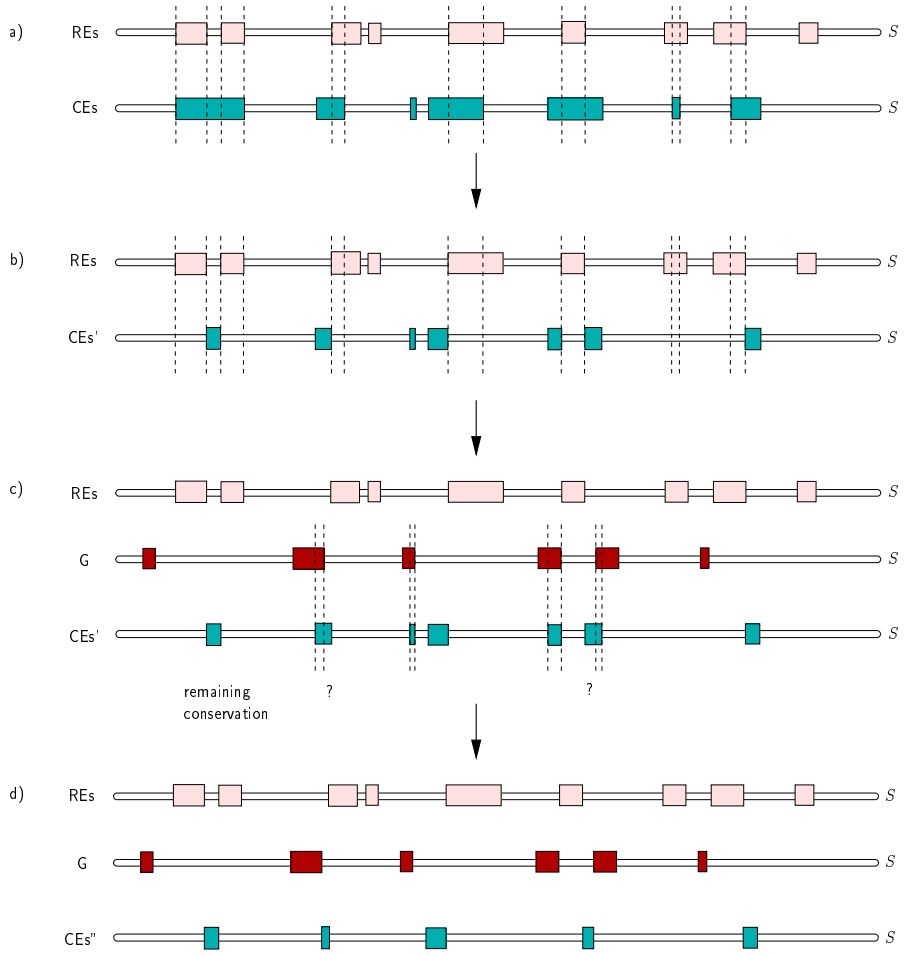

Figure 6.14: Graphical representation of the filtering procedure. Vertical dashed lines show the overlapping boundaries. a)-b) Conserved elements (CEs) that overlap with repeated elements (REs) are filtered out, generating a new list, CEs'. c)-d) Overlapping segments with `GENSCAN` predicted exons (G) are excluded from the CEs' list, considering the allowed *remaining conservation*. After filtering out all overlaps, CEs" are stored in the Repository of pCNSs.

termine the localization of the pCNSs in question, the `localize` program just checks the feature of the data component being compared. Placing pCNSs relative to `GENSCAN` predictions would result in a more detailed information than to the other two kinds of sequence annotations, as `GENSCAN` generates an output containing promoters, internal exons, and polyadenylation signals. However, the program identifies internal exons with a higher accuracy than promoters or even initial exons, as already mentioned in Section 6.3. Sometimes, it does not recognize promoter regions at all. In addition to this obstruction, the cDNA unsplicing and/or the gene annotation file may not present such

detailed information about the gene structure. For this reason, we decided to create a general pCNSs localization scheme, which can cope with all three kinds of sequence annotations. This means that the pCNSs are classified as follows:

1. *Intergenic upstream* gene G, when found upstream of the first exon of G;
2. *Intergenic downstream* gene G, when positioned downstream of the last exon of G;
3. *Intronic*, when localized somewhere between the first and last exon of G.
4. *Intergenic*, when localized more then 1 kb away from G.

Note that this classification takes into account the orientation of gene G. A graphical representation of this general scheme can be seen in Figure 6.15. At last, the results of the pCNSs arrangements are stored in the table *Localize*, where a foreign key points to the corresponding pCNS element in the *Potential CNSs* relation.

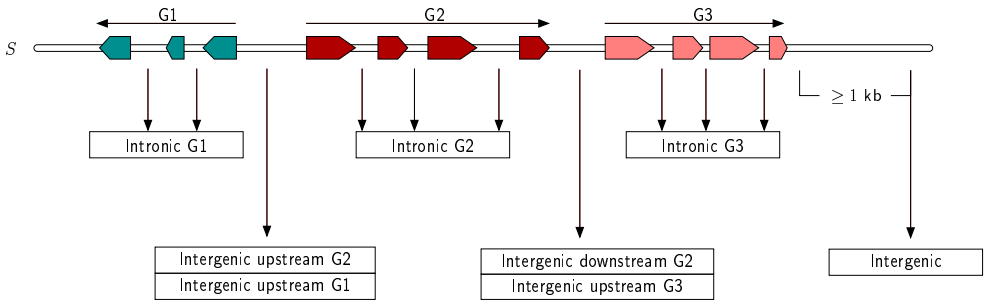

Figure 6.15: Graphical representation of the localization of pCNSs. The classification follows the position of pCNSs relative to exons, according to the gene orientation, being labeled as: *intronic*, *intergenic upstream*, *intergenic downstream*, or just *intergenic* to the corresponding gene.

The access to the Repository of pCNSs is done by SQL queries for the retrieval of any kind of stored data. 'SELECT' statements can be formulated with a high degree of flexibility. The user can get a subset of pCNSs which satisfies specific criteria, joining different tables if necessary. The sequences of selected pCNSs are extracted either from the database or the query sequences, and delivered as a multiple fasta file. These sequences can be submitted to further bioinformatics analysis tools, according to the individual investigation purposes.

6.5 Example Application

The identification and localization of pCNSs is the initial step to functional analysis of gene regulatory regions. The development of *Connosseur* gives biologists easy access to

such data, facilitating the pre-selection of genomic sequences that may contain functional activity. Wet-lab experiments carried out to demonstrate the functionality of CNSs are very expensive, laborious, and time consuming. For this reason, we decided to test *Connosseur* based on a literature example. At the moment, one of the most cited publications concerning CNSs deals with the identification of an interleukin regulator, by Loots *et al.* [73]. Their work was already mentioned in Chapter 3, as an example of cross-species genome comparison. We used their successful model to demonstrate the reliability of *Connosseur*, comparing the authors' approach with ours.

The human IL13, IL4, and IL5 genes are localized on chromosome 5q31. This region is known to be homologous to mouse chromosome 11 [43]. Loots *et al.* carried out a comparative approach between both genomic sequences using the sequence alignment tool `PipMaker` (see Chapter 3) [101]. At that time, sequences were still in a very early draft condition, leading the authors to use several contigs lying in the region of interest. Joining these contigs resulted in a segment of about 1 Mb to be compared between the two organisms. The comparative approach ended in many fragments of conserved elements, of several different sizes and percent identities. The authors have chosen an empirical threshold of minimal 100 bp with at least 70% percent identity to select a subset of conserved sequences. They observed 245 conserved elements fitting this criteria, from which 90 were defined as noncoding. Their classification segregated these CNSs into intronic (45) and non-genic (45), if the sequence was located farther than 1 kb from a known gene. The largest CNS found was about 400 bp (CNS1), presenting a percent identity of 84%, and was located between IL13 and IL4. In order to verify whether this large region has been conserved during evolution due to functional constraints rather than to insufficient divergence time, several wet-lab analyses have been accomplished, such as the determination of copy number in the human genome, the presence in other vertebrates, and transgenic mice experiments. These experiments confirmed the authors expectations, revealing that CNS1 has in fact regulatory functions, acting not only on IL13 and IL4, but also influencing many other genes in the chromosome 5q31 region.

Today, the human and mouse genomic sequences are close to completion, so that we only needed one contig of human chromosome 5 enclosing the gene region of interest[1]. Its counterpart, the mouse contig NT_031405, also included all the genes in question. These were the two genomic sequences entered in *Connosseur*, triggering the first step of the sequence annotation cascade. The cDNA sequences of IL13, IL4, and IL5 were entered in order to unsplice them onto the genome, getting the general gene structure of those genes (data not shown). The result of the second step of *Connosseur*, the comparison of both human and mouse contigs, can be observed in `GenAlyzer`'s visualization in Figure 6.16.

The information about the gene structures is visualized in the annotation graph, together with the repeated elements and predicted exons data. In the computed matching task, all direct and palindromic matches of minimal length 30 bp were searched, using

[1] GenBank accession number: NT_007072

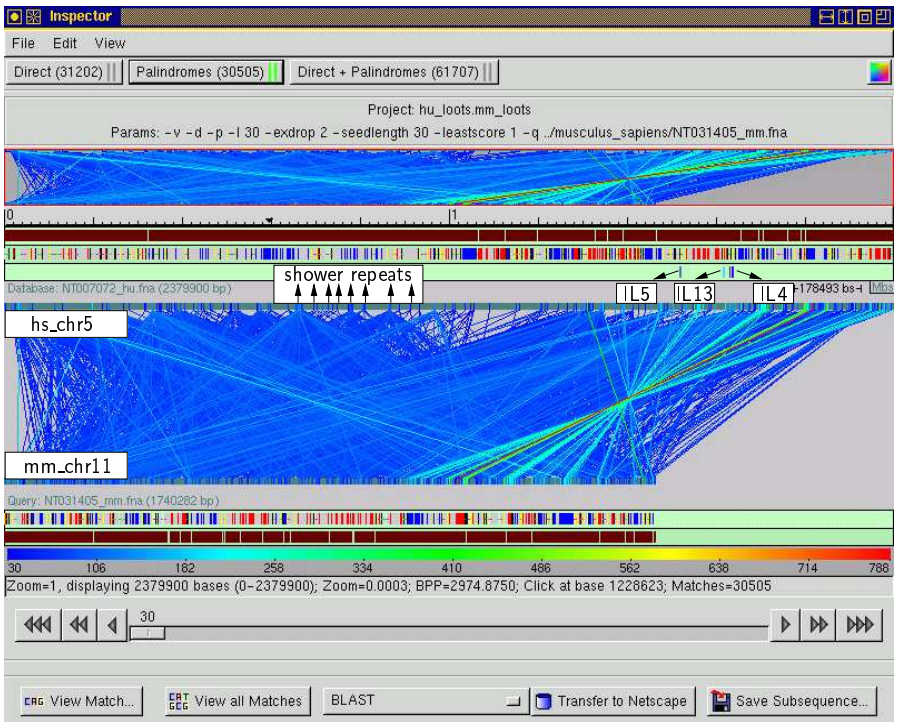

Figure 6.16: `GenAlyzer`'s visualization of the comparison between human chromosome 5q31 region (top) and mouse chromosome 11 (bottom). The blue background is a result from matches in conserved repeated elements, which generated a typical 'shower' shape.

an exdrop value of 3. The reason for such a low threshold is that conservations in noncoding regions are usually less similar than in coding regions. This approach should increase the number of long matches with less identity, as discussed before in Chapter 4. Even though, analyzing Figure 6.16, we observe that a large amount of short matches (of about 30 bp, represented in blue lines in the match graph) have been generated, almost reaching the background noise level. The shape of these matches led us to denominate them *shower repeats* (they also appear in self-comparisons of sequences) (see Figure 6.16). Zooming into this region (see Figure 6.17), we note that a single substring in the database sequence has many matches spread all over the query sequence. This is a typical sign for interspersed repeats, which are known to be scattered throughout the genome. This hypothesis is confirmed by the browser information of the `RepeatMasker` annotation graph. The single substring totally overlaps a repeated element.

These background matches do not deliver significant information when searching for pCNSs. Even so, the larger matches, ranging from 300 bp up to 800 bp, can be seen shining up in colors before the blue background (see Figure 6.16). This region of larger matches is restricted to the end of the contig, exactly where the genes of interest are arranged. We extracted this subsequence containing the most significant matches, sub-

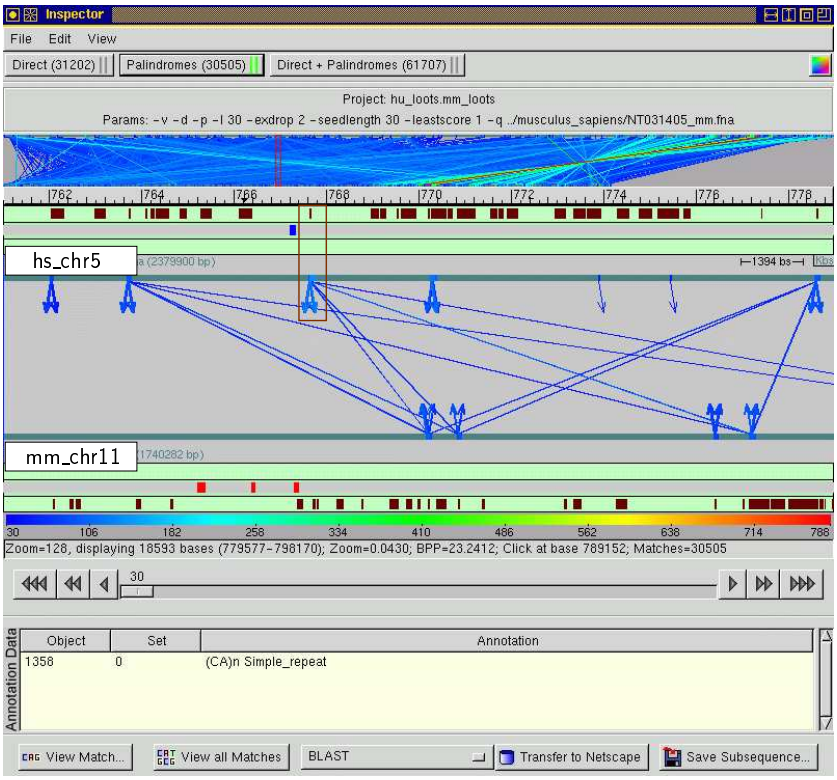

Figure 6.17: `GenAlyzer`'s visualization of the comparison between human chromosome 5q31 (top) region against mouse chromosome 11 (bottom) (ctd.). Zoomed view of the region of the *shower repeats*. A single substring matches several times in the compared sequence. This is usually a sign for interspersed repeats, or even simple repeat sequences, as shown in the annotation browser, referring to the classification of the match enclosed in a brown framed box.

mitting it to *Connosseur*. Figure 6.18 depicts the result of the matching task of this extracted subsequence. Although the computed pCNSs are already shown in the annotation graph (line 4), the blue background still confuses the visualization. We show this intermediate graphical output of *Connosseur* to compare with the final one, represented in Figure 6.19.

The graph in Figure 6.19, in turn, depicts the `GenAlyzer`'s graph of the modified match file, i.e., after the matches have been filtered for overlaps with repeated elements, `GENSCAN` predictions, and IL13 and IL4 unspliced cDNAs. This procedure allows a more precise investigation of the conserved regions that are left over, as the background noise has been eliminated. Figure 6.20 depicts an enlarged view of the IL13 and IL4 surroundings by zooming into the previous match graph. A highly conserved region is observed inbetween both genes, highlighted in a brown framed box. This region does not overlap with any information block in the annotation graph. Note that the fourth

annotation line represents the computed pCNSs, and the arrows in the match graph, their original length. Extracting the corresponding sequence, we were able to confirm that this segment refers to CNS1, the regulatory region Loots *et al.* had identified before (green block annotated in the graph). Another observation with regard to the conservation rates in the coding regions of IL13 and IL4 can also be made. It can be seen in the graph in Figure 6.20 that only a few matches were found to overlap exons. This is also totally consistent with the findings of Loots *et al.*, arguing that these genes are more conserved at the amino acid level rather than at the nucleotide sequence level.

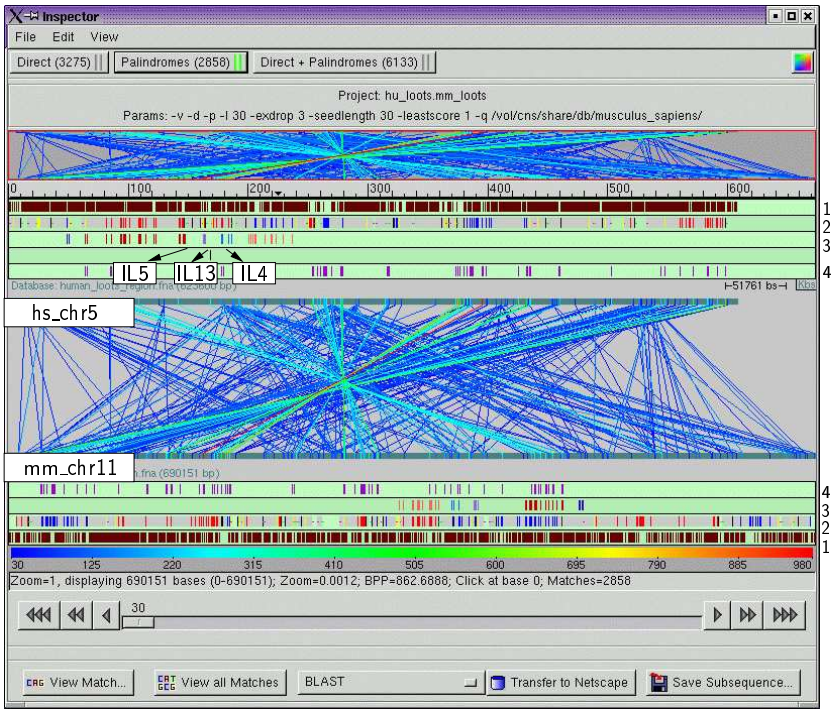

Figure 6.18: `GenAlyzer`'s visualization of the comparison between human (top) and mouse (bottom) genomic sequences in the interleukin genes region. Arrows indicate the position of IL13, IL4 and IL5 in the annotation graph. The blue background represents the large number of repeated elements conserved between both species. The annotation lines correspond to the `RepeatMasker` output (1), the `GENSCAN` prediction (2), the unspliced cDNAs (3) and to the computed pCNS (4).

With this example application from the literature, we were able to demonstrate the functionality of *Connosseur* in a rapid and reliable way, as CNS1 had already been tested for its regulatory activity. The performance of *Connosseur* referring to this application example is depicted in Table 6.1. The identification of new CNSs is accelerated by *Connosseur*, and the produced Repository of pCNSs both offers the selected sequences for wet-lab analysis, and the possibility to carry out further in silico investigations.

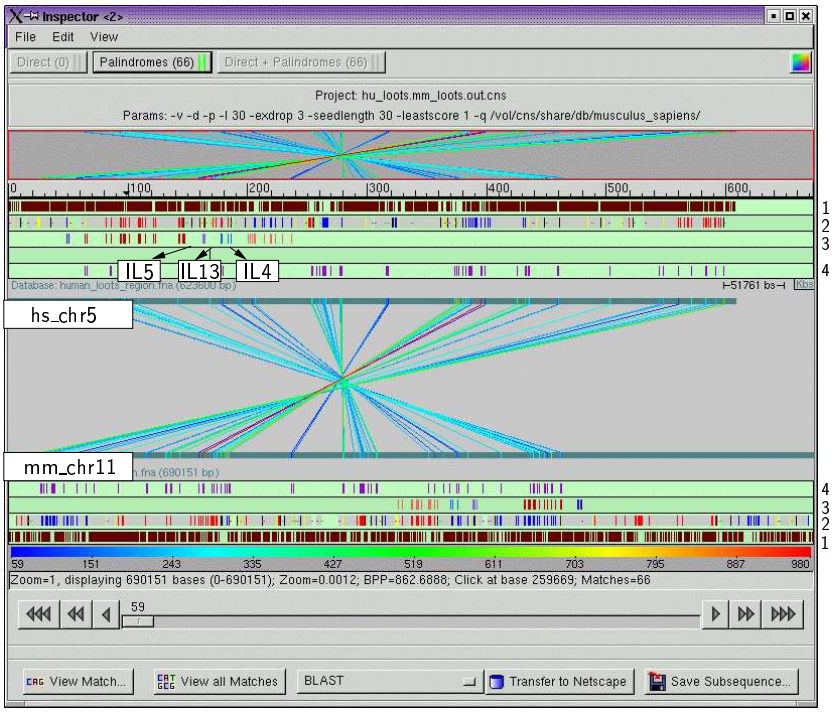

Figure 6.19: `GenAlyzer`'s visualization of the comparison between human (top) and mouse (bottom) genomic sequences in the interleukin genes region (ctd.). Arrows indicate the position of IL13, IL4, and IL5 in the annotation graph. The blue background noise has been eliminated by filtering the conserved elements list for repeated elements, `GENSCAN` predictions and IL13 and IL4 cDNAs. The annotation lines correspond to the `RepeatMasker` output (1), the `GENSCAN` prediction (2), the unspliced cDNAs (3) and to the computed pCNSs (4).

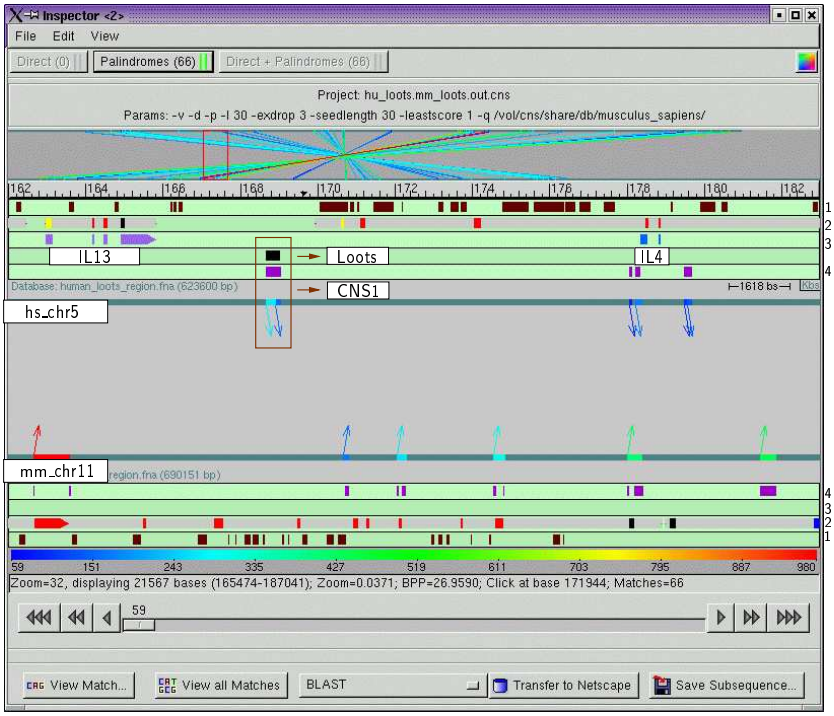

Figure 6.20: `GenAlyzer`'s visualization of the comparison between human (top) and mouse (bottom) genomic sequences in the interleukin genes region (ctd.). White boxes indicate the positions of IL13 and IL4 exons in the annotation graph. Clearly, the conserved segment of about 450 bp between IL13 and IL4 (enclosed by a brown framed box) does not overlap with any repeated element (1), or with exons of known genes (3) or with predicted ones (2), as it can be observed in the annotation graph. After analysis of this sequence, it has been confirmed to be the CNS1 element shown by Loots *et al.* to regulate not only IL13 and IL4, but also many other genes in the chromosome 5q31 region. Furthermore, the arrows in the match graph represent the original length of the matches utilized in the filtering. This procedure produced pCNSs without overlaps, as it can be clearly visualized in the annotation graph (4).

6.6 Analysis of the Wobbler Region

The development of *Connosseur* was originally motivated by the difficulties to detect the mutation responsible for the *wobbler* disease in coding regions of the candidate genes. In Chapter 5, we have already described in detail the sequence analysis of the *wobbler* critical region on mouse chromosome 11, delineating the genomic sequence comparisons with the syntenic region on human chromosome 2.

A substring of about 2.8 Mb containing the restricted *wobbler* region has been extracted from the mouse genomic contig NT_039515, in order to submit it to *Connosseur*. Similarly, the corresponding human region was extracted from contig NT_005375. Both genomic sequences have been entered into *Connosseur*, triggering the cascade of bioinformatics tools. After annotating the sequences for repeated elements, `GENSCAN` prediction, and exon localization of all candidate genes, NT_039515 was compared with NT_005375 using the default threshold settings (minimal length 30 bp, with an exdrop value of 3). The modified match graph, after filtering out sequence annotation features by *Connosseur*, is presented in Figure 6.21. This is a more detailed visualization of the critical region than shown before in Figure 5.13. The structure of all candidate genes has been annotated in the same color for both mouse and human sequences. The inverted segment can be clearly identified, although this general view suggests that the reversed region extends to the left and to the right beyond the *Peli1* and *KIAA0903* candidate genes that delimit the *wobbler* critical region.

Two regions of the restricted segment from Figure 6.21 were selected to analyze the relative content and localization of conserved elements by zooming into the match graph. The zooming factor is identical in both analyses. Concentrating on the mouse genomic sequence around the *Hcc8* gene, almost all annotated exons show corresponding matches in the human sequence (see Figure 6.22a). In contrast, the genomic region around the *Otx1* gene and part of the *KIAA0903* exons presents much more conserved elements in noncoding sequences, e.g., which do not overlap with annotated exons and also do not lie in repeated elements (as those had been filtered out beforehand) (see Figure 6.22b). The computed pCNSs are depicted in the annotation lines, as the arrows in the graph represent the matches in the original lengths of the conserved segments. In Section 6.7, we analyze the performance of *Connosseur* for the *wobbler* region.

However, it is important to take in consideration some observations about the association of divergence and interspersed repeats in noncoding genomic sequences. Recently, Chiaromonte *et al.* [18] have aligned a region of human chromosome 22 with the orthologous sequences on mouse chromosome 16, correlating the fraction of noncoding nonrepetitive nucleotides with the fraction of nucleotides belonging to repetitive elements (LINEs, SINEs, etc). Their results have shown that genomic segments with high density of repeated elements present only few matches in the noncoding nonrepetitive regions around the interspersed repeats. Consequently, regions on the human sequence with few repetitive elements presented much more conservations within noncoding nonrepetitive DNA. As repeated elements show a propension to cluster in specific regions

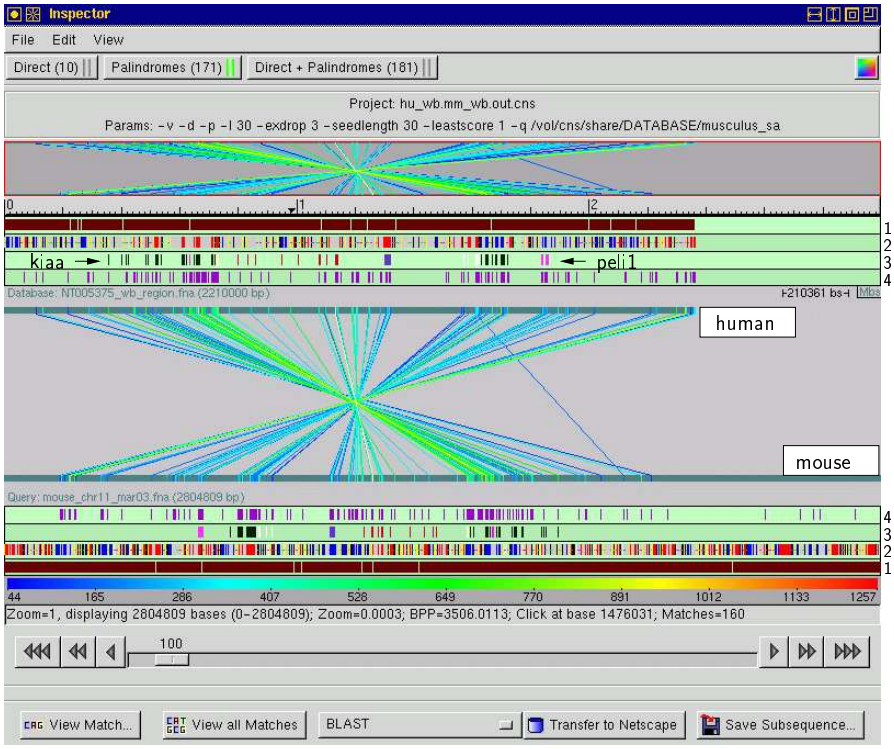

Figure 6.21: `GenAlyzer` visualization of the mouse (bottom) and human (top) genomic comparisons of the *wobbler* region. The graph displays all original, palindromic matches with a minimal length of 100 bp, computed with an exdrop value of 3, after the filtering procedure. Annotation lines 1 and 2 show the repeated elements and `GENSCAN` predictions, respectively. The *wobbler* candidate genes are represented in the annotation graph for both species (line 3). The computed pCNSs are depicted in the annotation line 4.

in the genome, the authors suggested that these segments are more prone to mutations than regions that are avoided by the insertion of mobile elements. The consequence would be that CNSs in regions of high density of interspersed repeats are more likely to be conserved due to functional constraints, resulting from their positive selection during evolution. Although these suppositions fit to our intuition, the described association was only recently mathematically confirmed [18]. In our example depicted in Figure 6.22, considering that the zooming factor has been the same for both analyses, the number of pCNSs visualized in (b) is higher when compared to (a). Observing the annotation graphs, the restricted region in (a) seems to cluster more repeated elements than in (b). Assuming the previously described hypothesis, not all pCNSs found in (b) will necessary have a regulatory function. However, further investigations are needed to confirm this hypothesis before generalizing it for the whole genome.

Another interesting observation of the analysis of the *wobbler* region concerns the

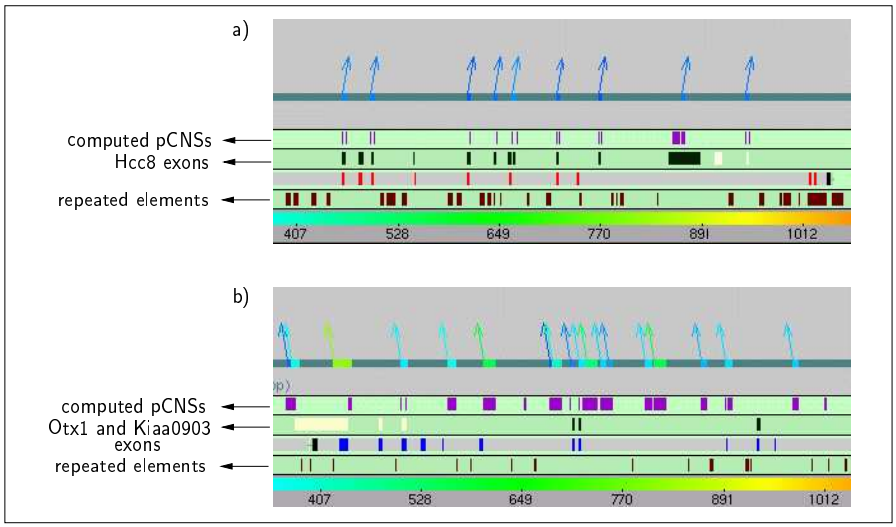

Figure 6.22: Correlation between conserved noncoding sequences in regions with density variation of repeated elements. (a) The segment around the mouse *Hcc8* gene presents very few pCNSs and a high frequency of interspersed repeats. (b) Many pCNSs have been found in the region of *Otx1* gene and *KIAA0903* exons, although the low density of repeated elements suggests a small tolerance of mutations in this genomic segment with subsequent selection.

identification of conserved splice sites. As already mentioned in the development of *Connosseur* in Section 6.4, the investigation of splice sites requires the analysis of the surroundings of exons. The user can determine how many base pairs are allowed as *remaining conservation* after the filtering procedure. Figure 6.23 depicts a section of the matching task of Figure 6.21, around the mouse *Ugp2* gene. The corresponding exons are represented as white blocks in the annotation graph. Comparing the original

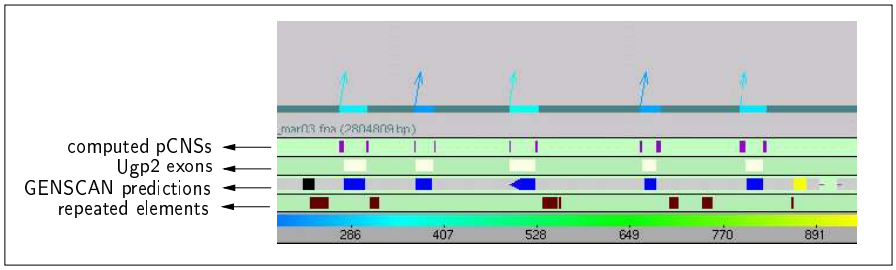

Figure 6.23: Conservation of splice sites between mouse and human. The region around the *Ugp2* gene is shown, depicting the pCNSs around the exons, with a *remaining conservations* value of 5 bp.

matches (arrows) with the computed pCNSs (annotation graph), we can clearly see the *remaining conservations* (set up to 5 bp) around the exons. These sequences can easily be selected by SQL query statements, in order to analyze them with more detail.

6.7 Performance

Optimizing the performance of *Connosseur* is a very complex task. It has been taken care of avoiding redundant data in the Repository of pCNSs, storing all items with a minimal amount of duplications. The performance of the overall system depends primarily on the efficiency of the underlying bioinformatics tools. All programs are running locally, allowing the submission of large genomic sequences. The complexity of `RepeatMasker` is $O(n^2)$, requiring much computational time when dealing with large sequence data sets. In contrast, the comparison of these sequences by `vmatch` is faster, due to its underlying virtual suffix trees structure. However, the complexity of the filtering procedure is also estimated to be $O(n^2)$, as all computed matches have to be compared with all entries of each annotation file. An overview of the overall computation time of *Connosseur* for both applications presented in Sections 6.5 (mouse: 690151 bp; human: 623600 bp) and 6.6 (mouse: 2804809; human: 2210000) is depicted in Table 6.1.

Processing		Section 6.5		Portion	Section 6.6		Portion
		human	mouse		human	mouse	
state	action	[min]	[min]	%	[min]	[min]	%
Annotate	RepeatMasker	115.2	129.4	96.2	418.8	330.5	79.2
	GenScan	1.6	1.8	1.3	5.6	3.7	1.0
	vmatch	0.0	0.0	0.0	0.2	0.3	0.1
Compare	vmatch	0.7		0.3	7.5		0.7
Analyze pCNSs	filter localize	2.5	3.1	2.2	75.9	104.0	19.00
	$\sum$	**254**		100	**946**		100

Table 6.1: Performance of *Connosseur* in the applications of Sections 6.5 and 6.6. The computation was carried out on a Sun UltraSparc II 450MHz.

Note that the computation of the *annotate* state is usually done only once for each genomic sequence. Yet *Connosseur* offers the flexibility to recompute the `vmatch` comparison tasks independently from the other processes in the system. Running *Connosseur* in modern computers allows the user to analyze different comparison outputs, computed under distinct threshold stringencies, in very few minutes.

6.8 Summary

In this Chapter, we presented *Connosseur*, a combination of bioinformatics tools, generating a Repository of pCNSs in an underlying database system. *Connosseur* has been developed in order to systematically identify conserved noncoding sequences between evolutionary related species via comparative genomics approaches. The computational cascade offers all basic information necessary for genomic comparisons and identification of pCNSs. After the sequence annotations, the comparison between the genomes of two different species is carried out with `vmatch`. This approach provides the user with all advantages of `vmatch`, already discussed in Chapter 4. Moreover, `GenAlyzer`'s interactive interface permits a rapid evaluation of the used matching task parameters, as demonstrated in the example application. The consistency of our results in comparison to the analysis of Loots *et al.* in the human chromosome 5q31 encourages biologists to use our system developed for identifying new pCNSs with potential functional role in the regulation of other genes.

For a biologist's intuition, after eighty million years of species divergence, the amount of conserved regions should be proportional to the selection of functional elements. However, recent investigations have shown that the number of CNSs definitively depends on the genomic region in study. By analyzing the *wobbler* critical region between mouse and human genomic sequences, it turned out that the clustering of potential CNSs may have a correlation with the localization of repeated elements. In regions with few repeated elements, the amount of pCNSs found is larger than regions with high density of interspersed repeats. In the future, more accurate regulatory elements prediction programs will allow further restrictions of the pCNSs. Finally, experimental methods have to be carried out for the detection of functionality of those pCNSs.

The integration of bioinformatics tools in *Connosseur* has shown to accelerate the extraction of biological meaningful information from raw sequence data. Furthermore, the flexible architecture of *Connosseur* enables the extension of bioinformatics computations in the future.

7 Discussion

The applications of REPuter have shown that the development of flexible software is crucial to support the progress in molecular biology research. The implementation of REPuter, in a combined effort of computer scientists and biologists, has shown to result in such an appropriate bioinformatics tool. We have demonstrated that the applicability of REPuter in biological problems extends far beyond its original purposes. The different interpretations of repeat analysis in DNA sequences, ranging from the traditional identification of low copy repeats to the investigation of gene structures, were successfully supported by the detection of repeated substrings using REPuter.

The computational and visual improvements of REPuter, implemented in vmatch and GenAlyzer, respectively, have shown to widen even more the range of biological applications based on the analysis of repetitive substrings. Besides EST clustering approaches (A. Sczyrba, personal communication), recent studies have shown the utilization of vmatch and GenAlyzer in the identification and analysis of ancestor centromeres, and their suggested correlation with human diseases (repeats-driven deletions, chromosomal rearrangements) [32]. However, the current major application of vmatch and GenAlyzer is found in the field of comparative genomics. With the sequencing of several organisms' genomes, computational biologists were challenged to develop fast tools that cope with large sequences. We have demonstrated that vmatch and GenAlyzer achieve those requirements through their efficient implementation of virtual suffix trees and easy-to-use interactive visualization.

We have shown that the comparative genomics approaches carried out with vmatch and GenAlyzer are also useful in biomedical research. Interspecies alignments are becoming a routine strategy for the identification of the sequences affected by mutations in disease genes. We have demonstrated that the utilization of comparative genomics could hint at the defect in the *wobbler* mouse that generates the motoneuron degeneration. The suggestion that the affected sequence could be a regulatory element led us to search for conserved noncoding sequences in the critical region. Comparing the syntenic segments containing the *wobbler* critical region in mouse and human, the analysis of the candidate gene structures revealed a large number of conserved regions in noncoding sequences. The complex networks of gene expression still make it difficult to detect the responsible mutation. The experimental work involved in the confirmation of biological function of all CNSs can often be significantly reduced if computational methods are able to give clear indications of CNSs localization beforehand. Improvements in such pre-screening processes of putative control elements are highly desirable in the biological community [110]. For this reason, we developed *Connosseur*, making use of existing

bioinformatics tools to automate a cascade for sequence annotation and genomic comparison. *Connosseur* provides a screening of conserved elements resulting from pairwise alignments, followed by accurate data mining and filtering procedures for the extraction of potential CNSs. All intermediate annotation and final comparison data are stored in the Repository of pCNSs, supported by a relational database system.

The properties of `vmatch` and `GenAlyzer` led us to choose them as main underlying alignment tools for the genomic sequence comparisons in *Connosseur*. The fast computation of large sequences in `vmatch` offers *Connosseur* a good performance. Within the wide range of applications of `vmatch`, the unsplicing of cDNAs onto the genomic sequence has been used in *Connosseur* as one option for the sequence annotation. The following strategy of gene structure comparison complements the gene prediction carried out by `GENSCAN`. The match and annotation graphs of `GenAlyzer` enable the user to clearly visualize the conserved elements within coding and noncoding sequences. In this context, we have shown that `vmatch` is an appropriate program for the identification of putative regulatory sequences in conserved noncoding regions. The reason is that functions of conserved noncoding sequences are unaffected by relatively small insertions or deletions of base pairs [27]. Consequently, standard local alignment algorithms that search for ungapped conserved regions (e.g., `PipMaker`, `VISTA`) are less suitable than `vmatch`, which allows gaps in addition to base substitutions in the search of degenerate matches.

The implementation of *Connosseur* allows the flexible choice of the pair of genomic sequences to be compared from the input set. Depending on the evolutionary distance between the species chosen for the comparison, it becomes difficult to definitely distinguish between yet undiscovered coding sequences and functionally noncoding sequences. For this reason, we have denominated the output of *Connosseur* as *potential conserved noncoding sequences*. To confirm the putative regulatory function, the ideal tests are gain-of-function assays, where the element is added to a reporter gene and transfected into appropriate cell lines, or the creation of transgenic animal models. Yet the possibility of being an exon can be tested in silico by searching for ESTs in current databases. A high-throughput automated system for such an approach has been recently developed by Beckstette (unpublished), called `G-enlight`. However, some authors discuss the limitations of EST databases, arguing that a large proportion of ESTs are contaminating sequences, including unspliced introns, genomic DNA and spurious transcriptions [64, 112]. For this reason, experimental analysis should be carried out for the confirmation of the element's expression in different tissues. The pCNS *Homoloc2* found to be highly conserved between mouse and human in the *wobbler* critical region is an example of such a possible contamination. Although it has been originally described as an EST from a human carcinoma cell line, its expression could not be detected in wet-lab experiments carried out by Fuchs [44].

The variability of conservation between two subsequences is dependent on the matching approach utilized in `vmatch` (edit- or hamming-distance, or exdrop approach). The capability of recomputing the comparison of selected genomic sequences in *Connosseur*

has shown to be worthwhile, as different thresholds for the matching task can be tested, making use of the versatility of `vmatch`. Although the general assumption is that sequences coding for proteins are usually more conserved between species than noncoding regions, there can be exceptions. For instance, the degeneration of the genetic code leads different triplets to code for the same amino acid. In this way, similarities at the amino acid level are sometimes higher than at the corresponding nucleotide sequence level. We have demonstrated an example of this divergence in conservation, consistent with previous experimental data analyses carried out by Loots *et al.* [73]. Using *Connosseur*, we have confirmed the localization of the previously described highly conserved noncoding sequence between the IL13 and IL4 genes. The chosen threshold did identify this CNS, but did not show any matches in the coding regions of IL13, and did not cover all exons of IL4 either. A recomputation of the matching task using a lower stringency may have included these exons, but also resulting in an overprediction of noncoding sequences as conserved (false positives) [27]. Nevertheless, the user has the choice to recompute the matching task, adapting the parameters to specific investigation purposes.

The problem of establishing the cutoff values to define noncoding elements as conserved is that they are usually based on biologist's intuition and practical experiences for what constitutes a biologically significant threshold. A possible improvement could be the statistical determination of the selection criteria in multiple genome comparisons, as described by Dubchak *et al.* [27]. Even though, some authors still affirm that "it is impossible to pick universal thresholds of conservation for the purpose of identifying sequences that are under selection" [42]. Concerning the distinct rates of evolution in different genomic regions, focusing on a smaller fragment around the genes of interest would be a more appropriate input for *Connosseur*. In this way, a broader range of stringency is provided, in which the user can test lower thresholds, retarding the background noise level in the matching task.

7.1 Future Work

The current version of *Connosseur* computes in fact only pairwise alignments, as it is based on `vmatch`. But the versatility of `vmatch` enables the comparison of two generated Repositories of pCNSs, in which the pCNSs sequences are extracted, concatenated and submitted to a matching task. This approach would deliver information about sequences and subsequences occurring in all four species used in both previously pairwise computed genomic comparisons. The specificity for the detection of regulatory regions increases significantly when more than two species are used in the comparative analysis. Noncoding sequences that are present in several species are more likely to be conserved due to functional constraints than to insufficient divergence time and shared ancestry. The choice of which species to compare is difficult, since sufficient similarity between the genomes must remain to identify homologous regions, but also a significant amount of mutations should have occurred, avoiding the detection of too many evolutionary leftovers. There

are current attempts to target the sequencing of further genomes on species that may deliver more significant results for comparative genomics. The National Institutes of Health Intramural Sequencing Center (NISC) 'Comparative Sequencing Program', for instance, established one of its goals to be the creation of a multispecies data set that might help to guide decisions about which genomes to sequence more completely in the future [42]. Recently, the `vmatch` program has also been the target of improvements towards multiple genome comparison, generating a tool called *Multiple Genome Alignment* (MGA) [56]. With the development of a user-friendly interface, and an interactive visualization for MGA, like `GenAlyzer`, *Connosseur* can be extended to use MGA to perform multiple genome comparisons.

The identification and localization of pCNSs is the common step for further investigations on conserved noncoding sequences. Ongoing work by Janina Scholz (diploma thesis) proposes the identification of regulatory elements among pre-computed pCNSs, which are involved in gene expression. Another research direction enabled by *Connosseur* is the detection of noncoding RNA genes, which have gained attention in the last years, for their importance in transcriptional regulation.

We believe that *Connosseur* may be the first step towards a public CNS database in the traditional way. With the facility to select pre-computed pCNSs for experimental assays, the functional annotation of these noncoding sequences is accelerated. As soon as enough data is present, including these detailed functional information, a public CNS database can be created. From then on, new identified pCNSs by *Connosseur* can be compared to CNSs in the database, whose functions have already been determined and well described. If new functional assignments are detected on further computed pCNSs by *Connosseur*, these sequences can, in turn, be submitted to the public CNS database.

Summarizing, *Connosseur* enables a reliable pre-screening of pCNSs resulting from pairwise genomic sequence comparison. The generated Repository of pCNSs allows the user to select specific sequences for further investigations, by querying the database. The amount of sequences resulting from the computation is dependent on three aspects: i) the evolutionary distance between both species being compared; ii) the genomic region in study; iii) the threshold used in the comparison task. *Connosseur* maintains a default line of computation throughout the tools cascade, keeping the necessary users' knowledge level of the underlying software relatively low. This feature will contribute to the acceptance of *Connosseur* in the biological community. In the future, improved tools for the in silico assignment of pCNSs biological functionality can be integrated to *Connosseur*, although subsequent confirmation of their function will still depend upon experimental assays [79].

Bibliography

[1] M.I. Abouelhoda, S. Kurtz, and E. Ohlebusch. The enhanced suffix array and its applications to genome analysis. In *Proceedings of the Second Workshop on Algorithms in Bioinformatics, LNCS 2452* , pages 449–463. Springer Verlag, 2002.

[2] M.D. Adams, J.M. Kelley, J.D. Gocayne, M. Dubnick, M.H. Polymeropoulos, H. Xiao, C.R. Merril, A. Wu, B. Olde, R.F. Moreno, A. Kerlavage, W. McCombie, and J. Venter. Complementary DNA sequencing: expressed sequence tags and human genome project. *Science*, 252:1651–1656, 1991.

[3] M.A. Ansari-Lari, J.C. Oeltjen, S. Schwartz, Z. Zhang, D.M. Muzny, J. Lu, J.H. Gorrell, A.C. Chinault, J.W. Belmont, W. Miller, and R.A. Gibbs. Comparative sequence analysis of a gene-rich cluster at human chromosome 12p13 and its syntenic region in mouse chromosome 6. *Genome Research*, 8:29–40, 1998.

[4] S. Bagheri-Fam, C: Ferraz, J. Demaille, G. Scherer, and D. Pfeifer. Comparative genomics of the SOX9 region in human and *Fugu rubripes*: conservation of short regulatory sequence elements within large intergenic regions. *Genomics*, 78(1-2):73–82, 2001.

[5] M.A. Batzer and P.L. Deininger. ALU repeats and human genomic diversity. *Nature Reviews Genetics*, 3:1–10, 2002.

[6] S. Batzoglou, L. Pachter, J.P. Mesirov, B. Berger, and E.S. Lander. Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Research*, 10:950–958, 2000.

[7] D.A. Benson, D.J. Karsch-Mizrachi, J.O. Lipman, and D.L. Wheeler. GenBank. *Nucleic Acids Research*, 31(1):23–27, 2003.

[8] J. Bowlers, G. Schepers, and P. Koopman. Phylogeny of the SOX family of developmental transcription factors based on sequence and structural indications. *Dev. Biol.*, 227:239–255, 2000.

[9] N. Bray, I. Dubchak, and L. Pachter. AVID: a global alignment program. *Genome Research*, 13(1):97–102, 2003.

[10] S. Brenner. The human genome: the nature of the enterprise. *Ciba Found Symp*, 149:6–12, 1990.

[11] C. Brewer, S. Holloway, P. Zawalnyski, A. Schinzel, and D. FitzPatrick. A Chromosomal deletion map of human malformations. *Am. J. Hum. Genet.*, 63:1153–1159, 1998.

[12] C. Brewer, S. Holloway, P. Zawalnyski, A. Schinzel, and D. FitzPatrick. A chromosomal duplication map of malformations: regions of suspected haplo- and triplolethality - and tolerance of segmental aneuploidy - in humans. *Am. J. Hum. Genet.*, 64:1702–1708, 1999.

[13] P. Bucher. Regulatory elements and expression profiles. *Current Opinion in Structural Biology*, 9:400–407, 1999.

[14] P.G. Buckley, K.K. Mantripragada, M. Benetkiewicz, I. Tapia-Paez, T.D. Stahl, M. Rosenquist, H. Ali, C. Jarbo, C. Bustos, C. Hirvela, B.S. Wilen, I. Fransson, C. Thyr, and cols. A full-coverage, high-resolution human chromosome 22 genomic microarray for clinical and research applications. *Human Molecular Genetics*, 11(25):3221–3229, 2002.

[15] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, 268:78–94, 1997.

[16] P. Chain, S. Kurtz, E. Ohlebusch, and T. Slezak. An applications-focused review of comparative genomics tools: capabilities, limitations, and future challenges. *Briefings in Bioinformatics*, 2003. (submitted).

[17] S.D. Cheng, H.L. Peng, and H.Y. Chang. Localization of the human UGP2 gene encoding the muscle isoform of UDP-glucose-pyrophosphorylase to 2p13-p14 by fluorescence in situ hybridization. *Genomics*, 39(3):414–416, 1997.

[18] F. Chiaromonte, S. Yang, L. Elnitski, V.B. Yap, W. Miller, and R.C. Hardison. Association between divergence and interspersed repeats in mammalian noncoding genomic DNA. *Proc. National Academy of Science*, 98(25):14503–14508, 2001.

[19] M. Clamp, D. Andrews, P. Barker, and cols. Ensembl 2002: accommodating comparative genomics. *Nucleic Acids Research*, 31(1):38–42, 2003.

[20] M.S. Clark. Comparative genomics: the key to understanding the human genome project. *BioEssays*, 21:121–130, 1999.

[21] J.E. Collins, A.J. Mungall, K.L. Badcock, J.M. Fay, and I. Dunham. The organization of the g-glutamil transferase genes and other low copy repeats in human chromosome 22q11. *Genome Research*, 7:522–531, 1997.

[22] T. Connolly and C. Begg. *Database systems: a practical approach to design, implementation and management*. Addison Wesley, third edition, 2002.

[23] M.C.G. Deniselle, S.L. Gonzalez, and A.F. Nicola. Cellular basis of steroid neuroprotection in the 'wobbler' mouse, a genetic model of motoneuron disease. *Cellular and Molecular Neurobiology*, 21(3):237–254, 2001.

[24] A. Descartes and T. Bunce. *Programming the Perl DBI*. O'Reilly, first edition, 2000.

[25] D.P. Dickinson, Y. Zhao, M. Thiesse, and M.J. Siciliano. Direct mapping of seven genes enconding human type 2 cystatins to a single site located at 20p11.2. *Genomics*, 24(1):172–175, 1994.

[26] C. Dietriech, B. Cusak, H. Wang, K. Rateitschak, A. Krause, and M. Vingron. Annotating regulatory DNA based on Man-Mouse genomic comparison. *Bioinformatics*, 18(2):S84–S90, 2002.

[27] I. Dubchak, M. Brudno, G.G. Loots, L. Pachter, C. Mayor, E.M. Rubin, and K.A. Frazer. Active conservation of noncoding sequences revelead by three-way species comparison. *Genome Research*, 10, 2000.

[28] L.W. Duchen and S.J. Strich. An hereditary motor neuron disease with progressive denervation of muscle in the mouse: the mutant 'wobbler'. *J. Neurol. Neurosurg. Psychiatry*, 31(6):535–542, 1968.

[29] I. Dunham, A.R. Hunt, J.E. Collins, and cols. The DNA sequence of human chromosome 22. *Nature*, 402:489–495, 1999.

[30] L. Edelman, R.K. Pandita, and B.E. Morrow. Low-copy-repeats mediate the common 3-Mb deletion in patients with velo-cardio-facial syndrome. *Am. J. Hum. Genet.*, 64:1076–1086, 1999.

[31] L. Edelman, R.K. Pandita, E. Spiteri, B. Funke, R. Goldberg, N. Palaiswamy, R.S.K. Chaganti, E. Magenis, R.J. Shprintzen, and B.E. Morrow. A common molecular basis for rearrengement disorders on chromosome 22q11. *Hum. Mol. Genet.*, 8:1157–1167, 1999.

[32] V. Eder, V. Mario, M. Teti, M. Rocchi, and N. Archidiacono. Chromosome 6 phylogeny in primates. 2003. (submitted).

[33] D. Ehling. *Etablierung von DNA-Sonden zur Detektion numerischer und struktureller Aberrationen der Chromosomen 13, 21 und 22 mittels Fluoreszent in situ Hybridisierung*. PhD thesis, Universität Bielefeld, 2003.

[34] E. Eichler. Masquerading repeats: paralogous pitfalls of the human genome. *Genome Research*, 8:758–762, 1998.

[35] D. Falconer. 'Wobbler' (wr). *Mouse News Letter*, 15:22, 1956.

[36] S. Fessele, H. Maier, C. Zischek, P.J. Nelson, and T. Werner. Regulatory context is a crucial part of gene function. *TRENDS in Genetics*, 18(2):60–63, 2002.

[37] J.P. Fitch, S.N. Gardner, T.A. Kuczmarski, S. Kurtz, R. Myers, L.L. Ott, T.R. Slezak, E.A. Vitalis, A.T. Zemla, and P.M. McCready. Rapid development of nucleic acid diagnostics. *Proceedings of the IEEE*, 90(11):1708–1721, 2002.

[38] W.M. Fitch. Distinguishing homologous from analogous proteins. *Syst. Zool.*, 19(2):99–113, 1970.

[39] R.D. Fleischmann, M.D. Adams, and White, O. *et al.* Whole-genome random sequencing and assembly of *Haemophilus influenzae Rd*. *Science*, 269:496–512, 1995.

[40] Foster, J.W. *et al.* Campomelic dysplasia and autosomal sex reversal caused by mutations in an SRY-related gene. *Nature*, 372:525–530, 1994.

[41] K. Frazer, J. Sheenan, R. Stokowski, X. Chen, R. Hosseini, J. Cheng, S. Fodor, D. Cox, and N. Patil. Evolutionary conserved Sequences on Human Chromosome 21. *Genome Research*, 11:1651–1659, 2001.

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

[43] K.A. Frazer, Y. Ueda, Y. Zhu, V.R. Gifford, M.R. Garofalo, N. Mohandas, C.H. Martin, M.J. Palazzolo, J.F. Cheng, and E.M. Rubin. Computational and biological analysis of 680 kb of DNA sequence from the human 5q31 cytokine gene cluster region. *Genome Research*, 7(5):495–512, 1997.

[44] S. Fuchs. *Mutationsanalyse von Kandidatengenen für die neurologische Mutation 'wobbler' der Maus*. PhD thesis, Universität Bielefeld, 2001.

[45] S. Fuchs, K. Resch, C. Thiel, M. Ulbrich, M. Platzer, H. Jockusch, and T. Schmitt-John. Comparative transcription map of the 'wobbler' critical region on mouse chromosome 11 and the homologous region on human chromosome 2p13-14. *BMC Genetics*, 3(14), 2002.

[46] M. Geel, E.E. Eichler, A.F. Beck, Z. Shan, T. Haaf, S.M. Maarel, R.R. Frants, and P.J. Jong. A cascade of complex subtelomeric duplications during evolution of the hominoid and old world monkey genomes. *Am. J. Hum. Genet.*, 70:269–278, 2002.

[47] C. Gibas and P. Jambeck. *Developing bioinformatics computer skills*. O'Reilly, first edition, 2001.

[48] B. Goettgens, L.M. Barton, M.A. Chapman, A.M. Sinclair, B. Knudsen, D. Grafham, J.G.R. Gilbert, J. Rogers, D.R. Bentley, and A.R. Green. Transcriptional regulation of the stem cell leukemia gene (SCL) - comparative analysis of five vertebrate SCL loci. *Genome Research*, 12:749–759, 2002.

[49] B. Goettgens, J.G.R. Gilbert, L.M. Barton, D. Grafham, J. Rogers, D.R. Bentley, and A.R. Green. Long-range comparison of human and mouse SCL loci: localized regions of sensitivity to restriction endonucleases correspond precisely with peaks of conserved noncoding sequences. *Genome Research*, 11:87–97, 2001.

[50] O. Gotoh. An improved algorithm for matching biological sequences. *JMB*, 162:705–708, 1982.

[51] S.G. Gregory, M. Sekhon, S. Schein, J. Zhao, K. Osoegawa, C.E. Scott, R.S. Evans, P.W. Burridge, T.V. Cox, C.A. Fox, and cols. A physical map of the mouse genome. *Nature*, 418:743–750, 2002.

[52] D. Gusfield. *Algorithms on strings, trees and sequences.* Cambridge University Press, NY, 1997.

[53] L. Guzman-Rojas, J.C. Sims, R. Rangel, C. Guret, Y. Sun, J.M. Alcocer, and H. Martinez-Valdez. PRELI, the human homologue of the avian px 19, is expressed by germinal center B lymphocytes. *Int. Immunol.*, 12:607–612, 2000.

[54] S. Halford, E. Lindsay, M. Nayudu, A.H. Carey, A. Baldini, and P.J. Scambler. Low-copy-number repeat sequences flank the DiGeorge/velo-cardio-facial syndrome loci at 22q11. *Human Molecular Genetics*, 2(2):191–196, 1993.

[55] R. Hardison, J. Oeltjen, and W. Miller. Long human-mouse sequence alignments reveal novel regulatory elements: a reason to sequence the mouse genome. *Genome Research*, 7:959–966, 1997.

[56] M. Höhl, S. Kurtz, and E. Ohlebusch. Efficient multiple genome alignment. *Bioinformatics*, 18(Suppl. 1):S312–S320, 2002.

[57] C. Huang, Y. Lin, Y. Yang, S. Huang, and C. Chen. The telomeres of *Streptomyces* chromosomes contain conserved palindromic sequences with potential to form complex secondary structures. *Mol. Microbiol.*, 28:905–916, 1998.

[58] N. Jareborg, E. Birney, and R. Durbin. Comparative analysis of noncoding regions of 77 orthologous mouse and human gene pairs. *Genome Research*, 9:815–824, 1999.

[59] Watson. J.D., M. Gilman, J. Witkowski, and M. Zoller. *Recombinant DNA*. Scientific American Books, second edition, 1992.

[60] Y. Ji, E.E. Eichler, S. Schwartz, and R.D. Nicholls. Structure of chromosomal duplicons and their role in mediating human genomic disorders. *Genome Research*, 10:597–610, 2000.

[61] J. Jurka. Repbase update: a database and an electronic journal of repetitive elements. *Trends Genet*, 16(9):418–20, 2000.

[62] S. Karlin and S.F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. USA*, 87:2264–2268, 1990.

[63] R. Kischner, D. Erturk, C. Zeitz, S. Sahin, J. Ramser, F.P.M. Cremers, H.H. Ropers, and W. Berger. DNA sequencing comparison of human and mouse retinitis pigmentosa GTPase regulator (RPGR) identifies tissue-specific exons and putative regulatory elements. *Hum. Genet.*, 109:271–278, 2001.

[64] S. Kondo, A. Shinagawa, T. Saito, H. Kiyosawa, I. Yamanaka, K. Aizawa, s. Fukuda, a. Hara, M. Itoh, J. Kawai, K. Shibata, and Y. Hayashizaki. Computational analysis of full-length mouse cDNAs compared with human genome sequences. *Mammalian Genome*, 12, 673-677 2001.

[65] B.F. Koop. Human and rodent DNA sequence comparisons: a mosaic model of genomic evolution. *Trends in Genetics*, 11(9):367–371, 1995.

[66] S. Kurtz. A Time and Space Efficient Algorithm for the Substring Matching Problem. Technical report, Zentrum für Bioinformatik, Universität Hamburg, 2002.

[67] S. Kurtz, J.V. Choudhuri, E. Ohlebusch, C. Schleiermacher, J. Stoye, and R. Giegerich. REPuter: the Manifold Applications of Repeat Analysis on a Genomic Scale. *Nucleic Acids Research*, 29(22):4633–4642, 2001.

[68] S. Kurtz, E. Ohlebusch, C. Schleiermacher, J. Stoye, and R. Giegerich. Computation and visualization of degenerate repeats in complete genomes. *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, 8:228–238, 2000.

[69] S. Laage, G. Zobel, and H. Jockusch. Astrocyte overgrowth in the brain stem and spinal cord of mice affected by spinal atrophy, 'wobbler'. *Developmental Neurscience*, 10(3):190–198, 1988.

[70] E.S. Lander, L.M. Linton, C. Birren, B. Nusbaum, M.C. Zody, J. Baldwin, K. Devon, M. Doyle, and Fitzhugh, W. *et al.* Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.

[71] S. Lefebvre, L. Burglen, S. Reboullet, O. Clermont, P. Burlet, L. Viollet, B. Benichou, C. Cruaud, P. Millasseau, and Zeviani, M. *et al.* Identification and characterization of a spinal muscle atrophy- determining gene. *Cell*, 80(1):155–165, 1995.

[72] D.F. Liu, P. Claxton, P. Marlton, A. Hajra, J. Siciliano, M. Freedman, S.C. Chandrasekharappa, K. Yanagisawa, R.L. Stallings, F.S. Collins, and cols. Identification of yeast artificial chromosomes containing the inversion 16 p-arm breakpoint associated with acute myelomonocytic leukemia. *Blood*, 82(3):716–721, 1993.

[73] G. Loots, R. Locksley, C. Blankespoor, Z. Wang, W. Miller, E. Rubin, and K. Frazer. Identification of a coordinate regulator of interleukins 4, 13, and 5 by cross-species sequence comparison. *Science*, 288:136–140, 2000.

[74] M.Z. Ludwig. Functional evolution of noncoding DNA. *Current Opinion in Genetics and Development*, 12:634–639, 2002.

[75] J. Lund, F. Chen, A. Hua, B. Roe, M. Budarf, B.S. Emanuel, and R. Reeves. Comparative sequence analysis of 634 kb of the mouse chromosome 16 region of conserved synteny with the human velocardiofacial syndrome region on chromosome 22q11.2. *Genomics*, 63:374–383, 2000.

[76] J.R. Lupski. Genomic disorders: structural features of the genome can lead to DNA rearrangements and human disease traits. *Trends in Genetics*, 14:417–422, 1998.

[77] C. Mayor, M. Brudno, J.R. Schwartz, A. Poliakov, E.M. Rubin, K.A. Frazer, L.S. Pachter, and I. Dubchak. VISTA: visualizing global DNA sequence alignments of arbitrary length. *Bioinformatics*, 16:1046–1047, 2000.

[78] H.E. McDermid and B.E. Morrow. Genomic disorders on 22q11. *Am. J. Hum. Genet.*, 70:1077–1088, 2002.

[79] M.H. Meisler. Evolutionary conserved noncoding DNA in the human genome: how much and what for? *Genome Research*, 11(10):1617–1618, 2001.

[80] F. Mignone, C. Gissi, S. Liuni, and G. Pesole. Untranslated regions of mRNAs. *Genome Biology*, 3(3):reviews0004.1–0004.10, 2002.

[81] B. Momjian. *PostgreSQL*. Addison Wesley, 2000.

[82] F. Mueller, P. Blader, and U. Straehle. Search for enhancers: teleost models in comparative genomic and transgenic analysis of cis regulatory elements. *BioEssays*, 24:564–572, 2002.

[83] R.J. Mural, M.D. Adams, E.W. Myers, and cols. A comparison of whole-genome shotgun-derived mouse chromosome 16 and the human genome. *Science*, 296:1661–1671, 2002.

[84] T. Nagase, K. Ishikawa, N. Miyajima, A. Tanaka, H. Kotani, N. Nomura, and O. Ohara. Prediction of the coding sequences of unidentified human genes. IX. The complete sequences of 100 new cDNA clones from brain which can code for large proteins in vitro. *DNA Res,*, 5(1):31–39, 1998.

[85] T. Nagase, K. Ishikawa, M. Suyama, R. Kikuno, N. Miyajima, A. Tanaka, H. Kotani, N. Nomura, and O. Ohara. Prediction of the coding sequences of unidentified human genes. XII. The complete sequences of 100 new cDNA clones from brain which code for large proteins in vitro. *DNA Res,*, 5(5):277–286, 1998.

[86] Casavant N.C., Scott L., Cantrell M.A., Wiggins L.E., Baker R.J., and Wichman H.A. The end of the LINE?: lack of recent L1 activity in a group of South American rodents. *Genetics*, 154(4):1809–1817, 2000.

[87] J.S. Oeltjen, T.M. Malley, D.M. Muzny, W. Miller, R.A. Gibbs, and J.W. Belmont. Large-scale comparative sequence analysis of the human and murine brutons tyrosine kinase loci reveals conserved regulatory domains. *Genome Research*, 7:315–329, 1997.

[88] P. Onyango, W. Miller, J. Lehoczky, C.T. Leung, B. Birren, S. Wheelan, K. Dewar, and A.P. Feinberg. Sequence and comparative analysis of the mouse 1-megabase region orthologous to the human 11p15 imprinted domain. *Genome Research*, 10:1697–1710, 2000.

[89] D. Pinkel, T. Straume, and W. Gray. Cytognetic analysis using quantitative, high-sensitivity, fluorescence hybridization. *Proc. Natl. Acad. Sci. USA*, 83:2934–2938, 1986.

[90] Y. Qiu, L. Cavelier, S. Chiu, X. Yang, E. Rubin, and J. Cheng. Human and mouse ABCA1 comparative sequencing and transgenesis studies revealing novel regulatory sequences. *Genomics*, 73:66–76, 2001.

[91] S. Rathke-Hartlieb, V.C. Schmidt, H. Jockusch, T. Schmitt-John, and J.W. Bartsch. Spationtemporal progression of neurodegeneration and glia activation in the 'wobbler' neuropathy of the mouse. *Neuroreport*, 10(16):3411–3416, 1999.

[92] K. Reichwald, J. Thiesen, T. Wiehe, J. Weitzel, W.H. Straetling, P. Kioschis, A. Poustka, A. Rosenthal, and M. Platzer. Comparative sequence analysis of the MECP2-locus in human and mouse reveals new transcribed regions. *Mammalian Genome*, 11:182–190, 2000.

[93] K. Resch. *Interspezies-Genom-Vergleich im Bereich des humanen Chr2p13: neue Kandidatengene für die neurologische Mutation 'wobbler' der Maus.* PhD thesis, Universität Bielefeld, 2000.

[94] K. Resch, H. Jockusch, and T. Schmitt-John. Assignment of homologous genes, Peli1/PELI1 and Peli2/PELI2, for the Pelle adaptor protein Pellino to mouse chromosomes 11 and 14 and human chromosomes 2p13.3 and 14q21, respectively, by physical and radiation hybrid mapping. *Cytogenet Cell Genet*, 92(1-2):172–174, 2001.

[95] K. Resch, D. Korthaus, N. Wedemeyer, A. Lengeling, M. Ronsiek, C. Thiel, K. Baer, H. Jockusch, and T. Schmitt-John. Homology between human Chromosome 2p13.3 and the 'wobbler' critical region on mouse chromosome 11: comparative high-resolution mapping of STS and EST loci on YAC/BAC contigs. *Mammalian Genome*, 9:893–898, 1998.

[96] O. Rinner and B. Morgenstern. AGenDA: gene prediction by comparative sequence analysis. *In Silico Biology*, 2:0018, 2002.

[97] W. Robberecht. Genetics of amyotrophic lateral sclerosis. *Journal of Neurology*, 247(Suppl.6):VI2–VI6, 2000.

[98] C. Schleiermacher. *Algorithmic support for PCR and genome-wide repeat analysis.* PhD thesis, Bielefeld University, 2001.

[99] T. Schmitt-John, M. Platzer, and H. Jockusch. Vergleichende Genomanalyse als Werkzeug fuer die Positionsklonierung. *BIOspektrum*, 5:390–397, 1999.

[100] S. Schwartz, W.J. Kent, Z. Smit, A. Zhang, R. Baertsch, R.C. Hardison, D. Haussler, and W. Miller. Human-mouse alignments with BLASTZ. *Genome Research*, 13:103–107, 2003.

[101] S. Schwartz, Z. Zhang, K. Frazer, A. Smit, C. Riemer, J. Bouck, R. Gibbs, R. Hardison, and W. Miller. PipMaker - a web server for aligning two genomic DNA sequences. *Genome Research*, 10(4):577–586, 2000.

[102] A. Sczyrba, J. Krüger, H. Mersch, S. Kurtz, and R. Giegerich. RNA-related tools on the Bielefeld Bioinformatics Server. *Nucleic Acids Research*, 2003. (submitted).

[103] Altschul S.F., Gish W., Miller W., Myers E.W., and Lipman D.J. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[104] L.G. Shaffer and J.R. Lupski. Molecular mechanisms for constitutional chromosomal rearrangements in humans. *Annu. Rev. Genet.*, 34:297–329, 2000.

[105] T.H. Shaik, H. Kurahashi, and B.S. Emanuel. Evolutionary conserved low copy repeats (LCRs) in 22q11 mediate deletions, duplications, translocations, and genomic instability: an update and literature review. *Genetics in Medicine*, 3(1):6–13, 2001.

[106] T.H. Shaikh, H. Kurahashi, S.C. Saitta, A.M. O'Hare, P. Hu, B.A. Roe, D.A. Driscoll, D.M. McDonald-McGinn, E.H. Zackai, M.L. Budarf, and B.S. Emanuel. Chromosome 22-specific low copy repeats and the 22q11.2 deletion syndrome: genomic organization and deletion endpoint analysis. *Hum. Mol. Genet.*, 9(4):489–501, 2000.

[107] A.F. Smit. Interspersed repeats and other mementos of transposable elements in mammalian genomes. *Curr. Opin. Genet. Dev.*, 9:657–663, 1999.

[108] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *JMB*, 147, 195-197 1981.

[109] G. Stoesser, W. Baker, A. Broek, M. Garcia-Pastor, C. Kany, T. Kulikova, R. Leinonen, Q. Lin, V. Lombard, R. Lopez, R. Mancuso, F. Nardone, P. Stoehr, M.A. Tuli, K. Tzouvara, and R. Vaughan. The EMBL nucleotide sequence database: major new developments. *Nucleic Acids Research*, 31(1), 2003.

[110] K. Sumiyama, C. Kim, and F.H. Ruddle. An efficient *cis*-element discovery method using multiple sequence comparisons based on evolutionary relationships. *Genomics*, 71:260–262, 2001.

[111] S. Tan and T.J. Richmond. Eukaryotic transcription factors. *Current Opinion in Structural Biology*, 8:41–48, 1998.

[112] A. Toyoda, H. Noguchi, T.D. Taylor, T. Ito, M.T. Pletcher, Y. Sakaki, R.H. Reeves, and M. Hattori. Comparative genomic sequence analysis of the human chromosome 21 Down Syndrome Critical Region. *Genome Research*, 12:1323–1332, 2002.

[113] A. Ureta-Vidal, L. Ettwiller, and E. Birney. Comparative genomics: genome-wide analysis in metazoan eukaryotes. *Nature Reviews Genetics*, 4:251–262, 2003.

[114] M. Valentin, M. Balvers, W. Pusch, G.F. Weinbauer, J. Knudsen, and R. Ivell. Structure and expression of the mouse gene encoding the endozepine-like peptide from haploid male germ cells. *European Journal of Biochemistry*, 267:5438–5449, 2000.

[115] W.W. Wasserman, M. Palumbo, W. Thompson, J.W. Fickett, and C.E. Lawrence. Human-mouse genome comparison to locate regulatory sites. *Nature Genetics*, 26:225–228, 2000.

[116] R. Waterson, E.S. Lander, and E. Birney. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420:520–573, 2002.

[117] J.D. Watson and F.H. Crick. A structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, 1953.

[118] L. Wei, Y. Liu, I. Dubchak, J. Shon, and J. Park. Comparative genomics approaches to study organism similarities and differences. *Journal of Biomedical Informatics*, 35:142–150, 2002.

[119] Z. Zhang, S. Schwartz, L. Wagner, and W. Miller. A greedy algorithm for aligning DNA sequences. *J Comput Biol*, 7:203–14, 2000.

A Links to Human and Mouse Genome Projects

The Bielefeld University Bioinformatics Server offers bioinformatic tools developed by the Research Group in Practical Computer Science and some collaborators. They are collected in the Sequence Analysis Department of the Bibiserv. For more details see A. Sczyrba [102].

The Bibiserv Library section provides some interesting links related to mouse and human genome sequencing projects. It serves as a complement for such comparative studies, with particular interest in the mouse chromosome 11 and human chromosome 2. The whole web-site is structured as follows:

1. *General Information*: both Mouse and Human Genome Projects are described in general articles about their histories, meanings and goals. Some laboratories involved in the sequencing projects are listed, includig specific information about mouse chromosome 11 and human chromosome 2, in the second subtopic.

2. *Homology Between Species*: many questions are made with reference to the use of mice as a model organism in order to study and interpret some human diseases. This topic links to comments about mouse and human homologies.

3. The large amount of data generated since the beginning of the genome sequencing projects has led us to take advantage of the computational biology to interpret and understand those data. Links to some general bioinformatics services and specific homology and sequence analysis programs can be found in the topic *Bioinformatic Tools*.

4. Other interesting related sites are linked in the topic *More Genome Links*, including references to some industries, that develop and supply molecular biological products.

5. The last topic *Related Publications* links to some journals for personal article searches, including a direct link to PubMed *mouse/human/homology* query, with possibility of further subject restrictions.

B Useful Web Sites

1. Bielefeld Bioinformatics Server:

   - BIBISERV: http://bibiserv.techfak.uni-bielefeld.de
   - BIBISERV - Literatute: http://bibiserv.techfak.uni-bielefeld.de/library/genomes

1. Tools:

   - `REPuter`: http://www.genomes.org
   - `RepeatMasker`: http://repeatmasker.genome.washington.edu/RM/RepeatMasker.html
   - *GenFisher*: http://bibiserv.techfak.uni-bielefeld.de/genefisher
   - `GENSCAN`: http://genes.mit.edu/GENSCAN.html
   - `PipMaker`: http://bio.cse.psu.edu
   - `VISTA`: http://www-gsd.lbl.gov/vista

1. General Links to Genome Projects:

   - NCBI: http://www.ncbi.nlm.nih.gov
   - Genome Sequencing Center Jena: http://genome.imb-jena.de
   - The Jackson Laboratory: http://www.jax.org
   - The Whithead Institute, MIT: http://www-genome.wi.mit.edu
   - Mouse Genome Center: http://www.mgc.har.mrc.ac.uk
   - Baylor College of Medicine: http://www.mouse-genome.bcm.tmc.edu
   - The Mouse Genome Informatics: http://www.informatics.jax.org
   - Online Mendelian Inheritance in Man: http://www.ncbi.nlm.nih.gov/omim
   - The Human Genome Organisation: http://www.gene.ucl.ac.uk/hugo
   - The Sanger Center: http://www.sanger.ac.uk

C Database Relations

input_seq	
Attribute	Datatype
id	INTEGER DEFAULT nextval('seq_id_seq') PRIMARY KEY
project	VARCHAR(20) DEFAULT '-'
filename	VARCHAR(50) DEFAULT '-'
name	TEXT UNIQUE DEFAULT '-'
acess_no	VARCHAR(20) DEFAULT '-'
species	VARCHAR(50) DEFAULT '-'
no_contigs	INT DEFAULT -1
length	INT DEFAULT -1
sequence	TEXT DEFAULT '-'
created	TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Table C.1: *Genomic Sequences* features.

contigs	
Attribute	Datatype
id	INT DEFAULT nextval('seq_id_seq') PRIMARY KEY
input_seq_fk	INT DEFAULT -1
header	VARCHAR(200) DEFAULT '-'
ctg_nr	INT DEFAULT -1
rel_length	INT DEFAULT -1
abs_start	INT DEFAULT -1
abs_stop	INT DEFAULT -1
created	TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Table C.2: Contigs features.

repeat_masker	
Attribute	Datatype
id	INTEGER DEFAULT -1
input_seq_fk	INT DEFAULT -1
sw_score INT4	DEFAULT -1
diversity_perc	FLOAT DEFAULT 0.0
deletion_perc	FLOAT DEFAULT 0.0
insertion_perc	FLOAT DEFAULT 0.0
sequence_header	TEXT DEFAULT ”
query_begin	INT4 DEFAULT -1
query_end INT4	DEFAULT -1
repeat_abs_pos	INT4 DEFAULT -1
repeat_length	INT4 DEFAULT -1
query_left	INT4 DEFAULT -1
query_strand	CHAR(1) DEFAULT ’Z’
repeat_type	VARCHAR(25) DEFAULT ’-’
repeat_class	VARCHAR(25) DEFAULT ’-’
repeat_begin	VARCHAR(25) DEFAULT ’-’
repeat_end	VARCHAR(25) DEFAULT ’-’
repeat_left	VARCHAR(10) DEFAULT ’-’
repeat_id INT4	DEFAULT 0
created	TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Table C.3: Repeated elements features.

genscan	
Attribute	Datatype
id	INTEGER DEFAULT -1
input_seq_fk	INT DEFAULT -1
gn_ex	FLOAT DEFAULT 0.0
type	VARCHAR(25) DEFAULT '-'
strand	CHAR(1) DEFAULT 'Z'
ex_begin	INT4 DEFAULT -1
ex_end	INT4 DEFAULT -1
length	INT4 DEFAULT -1
fr	INT4 DEFAULT -1
ph	INT4 DEFAULT -1
i_ac	INT4 DEFAULT -1
do_t	INT4 DEFAULT -1
cdg_rg	INT4 DEFAULT -1
p	FLOAT DEFAULT 0.0
tscr	FLOAT DEFAULT 0.0
created	TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Table C.4: `GENSCAN` prediction.

input_cdnas	
Attribute	Datatype
id	INTEGER DEFAULT nextval('seq_id_cdnas') PRIMARY KEY
input_seq_fk	INT DEFAULT -1
cdnadir	VARCHAR(100) DEFAULT '-'
filename	VARCHAR(20) DEFAULT '-'
name	TEXT DEFAULT '-'
acess_no	VARCHAR(20) DEFAULT '-'
length	INT DEFAULT -1
sequence	TEXT DEFAULT '-'
created	TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Table C.5: *cDNA* sequences features.

cdna_match_file	
Attribute	Datatype
id	INTEGER DEFAULT -1
matchvalues_fk	INTEGER DEFAULT -1
input_seq_fk	INT DEFAULT -1
DB_sequence	VARCHAR(50) DEFAULT '-'
CDNA_sequence	VARCHAR(50) DEFAULT '-'
match_length_DB	INT4 DEFAULT -1
contig_DB	INT4 DEFAULT -1
rel_pos_DB	INT4 DEFAULT -1
abs_pos_DB	INT4 DEFAULT -1
type	CHAR(1) DEFAULT 'Z'
match_length_Q	INT4 DEFAULT -1
contig_Q	INT4 DEFAULT -1
rel_pos_Q	INT4 DEFAULT -1
abs_pos_Q	INT4 DEFAULT -1
distance_value	INT4 DEFAULT -0
e_value	FLOAT DEFAULT 0.0
score	FLOAT DEFAULT 0.0
percent_identity	FLOAT DEFAULT 0.0
created	TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Table C.6: *Exon* table: unspliced cDNAs.

system	
Attribute	Datatype
table_name	VARCHAR(80)DEFAULT 'Z'
id	INT DEFAULT -1

Table C.7: Last utilized *id* numbers in each table.

seq_match_file	
Attribute	Datatype
id	INTEGER DEFAULT -1
matchvalues_fk	INTEGER DEFAULT -1
match_length_DB	INT4 DEFAULT -1
contig_DB	INT4 DEFAULT -1
rel_pos_DB	INT4 DEFAULT -1
abs_pos_DB	INT4 DEFAULT -1
type	CHAR(1) DEFAULT 'Z'
match_length_Q	INT4 DEFAULT -1
contig_Q	INT4 DEFAULT -1
rel_pos_Q	INT4 DEFAULT -1
abs_pos_Q	INT4 DEFAULT -1
distance_value	INT4 DEFAULT -0
e_value	FLOAT DEFAULT 0.0
score	FLOAT DEFAULT 0.0
percent_identity	FLOAT DEFAULT 0.0
created	TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Table C.8: Comparison of two genomic sequences.

cdna_match_values or seq_match_values	
Attribute	Datatype
id	INTEGER DEFAULT -1
match_file	VARCHAR(50) UNIQUE DEFAULT '-'
DB_sequence	VARCHAR(50) DEFAULT '-'
QUERY_sequence	VARCHAR(50) DEFAULT '-'
type	CHAR(20) DEFAULT 'Z'
min_length	INTEGER DEFAULT -1
approach	CHAR(10)DEFAULT 'Z'
approach_value	INT4 DEFAULT -1
seedlength	INT4 DEFAULT -1
leastscore	INT4 DEFAULT -1
created	TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Table C.9: History of `vmatch` parameters for matching tasks.

cns_DB of cns_Q	
Attribute	Datatype
id	INTEGER DEFAULT -1
id	INTEGER DEFAULT -1
seq_match_file_fk	INTEGER DEFAULT -1
length	INT4 DEFAULT -1
start	INT4 DEFAULT -1
created	TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Table C.10: *Potential CNSs* table (for DB and Q sequences).

loc_DB or loc_Q	
Attribute	Datatype
id	INTEGER DEFAULT -1
cns_Q_fk	INTEGER DEFAULT -1
gene_type	VARCHAR(1) DEFAULT '-'
gene_name	INT4 DEFAULT -2
loc_type	VARCHAR(2) DEFAULT '-'
created	TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Table C.11: Localization of pCNSs (for DB and Q sequences).